Attribute-based Keyed (Fully) Homomorphic Encryption

Keita Emura* Shingo Sato† Atsushi Takayasu‡

April 25, 2024

Abstract

Keyed homomorphic public key encryption (KHPKE) is a variant of homomorphic public key encryption, where only users who have a homomorphic evaluation key can perform a homomorphic evaluation. Then, KHPKE satisfies the CCA2 security against users who do not have a homomorphic evaluation key, while it satisfies the CCA1 security against users who have the key. Thus far, several KHPKE schemes have been proposed under the standard Diffie-Hellman-type assumptions and *keyed fully homomorphic encryption* (KFHE) schemes have also been proposed from lattices although there are no KFHE schemes secure solely under the LWE assumption in the standard model. As a natural extension, there is an identity-based variant of KHPKE; however, the security is based on a q-type assumption and there are no attribute-based variants. Moreover, there are no identity-based variants of KFHE schemes due to the complex design of the known KFHE schemes. In this paper, we provide two constructions of attribute-based variants. First, we propose an attribute-based KFHE (ABKFHE) scheme from lattices. We start by designing the first KFHE scheme secure solely under the LWE assumption in the standard model. Since the design is conceptually much simpler than known KFHE schemes, we replace their building blocks with attribute-based ones and obtain the proposed ABKFHE schemes. Next, we propose an efficient attribute-based KHPKE (ABKHE) scheme from a pair encoding scheme (PES). Due to the benefit of PES, we obtain various ABKHE schemes that contain the first identity-based KHPKE scheme secure under the standard k-linear assumption and the first pairing-based ABKHE schemes supporting more expressive predicates.

*Kanazawa University, Japan
†Yokohama National University, Japan
‡The University of Tokyo and National Institute of Advanced Industrial Science and Technology, Japan

Contents

1 Introduction

1.1 Background

Given two ciphertexts $\mathsf{ct}^{(1)}$ and $\mathsf{ct}^{(2)}$ of (multiplicative) homomorphic encryption (HE), where they are encryptions of $\mu^{(1)}$ and $\mu^{(2)}$, respectively, arbitrary users can compute an evaluated ciphertext ct that is an encryption of $\mu^{(1)} \cdot \mu^{(2)}$. Given an arbitrary circuit C and ciphertexts $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(L)}$ of fully homomorphic encryption (FHE), where they are encryptions of $\mu^{(1)}, \ldots, \mu^{(L)}$, respectively, arbitrary users can compute an evaluated ciphertext ct_{C} that is an encryption of $\mathsf{C}(\mu^{(1)}, \ldots, \mu^{(L)})$. After Gentry [Gen09] proposed the first FHE scheme, several improved FHE schemes have been proposed such as [Bra12, BGV12, BV11a, BV11b, BV14, GSW13, vGHV10]. The publicly computable homomorphism provides several applications such as delegated computation and multi-party computation. In contrast, the nature inherently prevents (F)HE schemes from achieving the CCA2 security. Thus, several CCA1-secure (F)HE schemes have been proposed such as the Cramer-Shoup-lite [CS98] and FHE schemes [CRRV17, DGM15, LMSV12, ZPS12]. However, Loftus et al. showed that CCA1-secure FHE schemes may be vulnerable if there are ciphertext validity checking oracles [LMSV12] as Bleichenbacher's attack on RSA [Ble98].

To reconcile homomorphic operations and the chosen ciphertext security, Emura et al. introduced a notion of *keyed homomorphic public key encryption* (KHPKE) [EHO+13]. As opposed to (F)HE, only users who have a homomorphic evaluation key can compute evaluated ciphertexts of KHPKE. The standard security requirement of KHPKE called the KH-CCA security ensures that a KHPKE scheme satisfies the CCA2/CCA1 security against an adversary without/with a homomorphic evaluation key, respectively. Thus, the KH-CCA security is strictly stronger than the CCA1 security. Moreover, KH-CCA-secure KHPKE schemes are secure even in the presence of ciphertext validity checking oracles [Emu21]. Libert et al. [LPJY14] proposed the first KH-CCA-secure multiplicative KHPKE scheme, then Jutla and Roy [JR15] and Emura et al. [EHN+18] proposed improved schemes. Among them, Emura et al.'s scheme is the most efficient since it does not require pairing unlike [JR15, LPJY14] and satisfies the KH-CCA security under the DDH assumption.

Lai et al. extended the notion of KHPKE and proposed the first keyed FHE (KFHE) scheme [LDM+16] under the LWE assumption and iO [BGI+01]; however, it does not satisfy the KH-CCA security but only the weaker security which is not CCA1 but only the CPA security against an adversary with a homomorphic evaluation key. Then, Sato et al. proposed the first KH-CCA-secure KFHE scheme under the LWE assumption [SET22]. In particular, Sato et al. followed the complex design methodology of Jutla and Roy's KHPKE scheme [JR15] based on a strong dual-system simulation-sound NIZK system for Diffie-Hellman languages. To construct a strong dual-system simulation-sound NIZK system for FHE ciphertexts, Sato et al. have to rely on either zk-SNARKs for arithmetic circuits based on knowledge assumptions [BBC+18, BCC+17, BCCT13, GGPR13, MBKM19, ZSZ+22] or zk-SNARKs for NP in the (quantum) random oracle model [CMS19]. Thus, there are no known KFHE schemes whose KH-CCA security is based solely on the LWE assumption in the standard model. Maeda and Nuida [MN22] proposed a keyed two-level homomorphic encryption scheme that supports the additive homomorphism with a single multiplication under the SXDH assumption.

As another direction of the topic, Emura et al. constructed a pairing-based identity-based keyed homomorphic encryption (IBKHE) scheme [EHN+18]. Although the scheme satisfies the adaptive KH-CCA security, it is based on a q-type assumption. Thus far, there are no known pairing-based IBKHE schemes under the standard assumptions although there are various pairing-based homomorphic identity-based encryption (IBE) schemes under such assumptions [BB04, CLL+14, CW14, Lew12, Wat05, Wat09]. Similarly, there are no known attribute-based keyed homomorphic encryption (ABKHE) schemes supporting more expressive predicates although the pair encoding framework [Att14, Wee14] enables us to construct various pairing-based expressive attribute-based encryption (ABE) schemes [AC16, AC17, Amb21, ABS17, Att16, CGW15, CG17, Tak21]. The ABE schemes are adaptively secure under the q-ratio assumption and the standard k-linear assumption for expressive and simple predicates, respectively. Moreover, there are no known identity-based keyed fully homomorphic encryption (IBKFHE) schemes and attribute-based keyed fully homomorphic encryption (ABKFHE) schemes, while there are various known lattice-based identity-based and attribute-based FHE schemes such as [BCTW16, CM15, GSW13, HK17, ML19, PD20]. These situations stem from the fact that known design methodologies of KHPKE and KFHE are too complex to extend to identity/attribute-based settings. In other words, known constructions of KH-CCA-secure K(F)HE schemes rely on specific techniques that are not common in the context of public key encryption. For example, Emura et al. [EHN+18] introduced additional security notions for universal$_2$ hash proof system [CS02] and proved the KH-CCA security, where the additional security notions have not been used in other papers. As we explained above, Jutla and Roy [JR15] and Sato et al. [SET22] used strong dual-system simulation-sound NIZK systems that have been used only in these papers.

1.2 Our Contribution

In this paper, we first propose a generic construction of ABKFHE whose building blocks can be instantiated under the standard LWE assumption. For this purpose, we start by designing the first KH-CCA-secure KFHE scheme solely based on the LWE assumption in the standard model by modifying Canetti et al.'s CCA1-secure FHE scheme [CRRV17]. Specifically, Canetti et al. constructed a CCA1-secure FHE scheme from multi-key FHE (MFHE) [AJJM20, CM15, LTV12, MW16, PS16] and IBE, where MFHE schemes [AJJM20, MW16, PS16] are secure in the standard model and there are various IBE schemes secure in the standard model such as [ABB10a, Yam17]. In addition to MFHE and IBE, we use only simple primitives and construct KFHE. Indeed, we additionally use one-time signatures (OTS) and message authentication codes (MAC). The design methodology is very simple since we just combine the Canetti-Halevi-Katz transformation [CHK04] and the encrypt-then-MAC paradigm [BN08] which are the standard techniques to prove the CCA2 security of public/symmetric key encryption. As a result, the simplicity enables us to extend the proposed KFHE scheme and obtain a KH-CCA-secure ABKFHE scheme supporting cross-attribute evaluations by replacing IBE and MAC with delegatable ABE (DABE).

Unfortunately, the proposed ABKFHE scheme is not very efficient since the size of an evaluated ciphertext depends on the number of input ciphertexts although the feature is not the disadvantage of the proposed ABKFHE scheme since the known CCA1-secure FHE scheme secure solely under the LWE assumption in the standard model [CRRV17] and attribute-based FHE schemes supporting cross-attribute evaluation [BCTW16, ML19, PD20] have similar features. Thus, we overcome the issue by restricting the functionality and propose an efficient ABKHE scheme that supports multiplicative homomorphism without cross-attribute evaluations. Specifically, we construct the proposed ABKHE scheme from a pair encoding scheme (PES) [Att14, Wee14]. Due to the benefit of the pair encoding framework, we obtain adaptively KH-CCA-secure ABKHE schemes for various expressive predicates under the q-ratio assumption and those for simple predicates under the standard k-linear assumption using known PES such as [AC16, AC17, Att14, Att16, Att19, AY15, CGW15, Tak21, Wee14]. The result includes the first pairing-based IBKHE scheme under the standard k-linear assumption. Our design methodology is similar to Emura et al.'s KHPKE scheme [EHN+18]. Although Emura et al.'s proof based on the hash proof system [CS02] is complicated, we can simplify the proof by focusing on the

Table 1: Comparison among keyed homomorphic encryption schemes

Scheme	Homomorphism	Access Control	Complexity Assumption
LPJY14 [LPJY14]	Multiplicative	None	DLIN
JR15 [JR15]	Multiplicative	None	SXDH
LDM+16 [LDM+16]	Fully	None	LWE + iO
EHN+18 [EHN+18]	Multiplicative	None	DDH
	Additive	None	DCR
	Multiplicative	Identity-based	q-ABDHE
SET22 [SET22]	Fully	None	LWE + Knowledge
			LWE + (Q)ROM
MN22 [MN22]	Two-Level	None	SXDH
This Work	Fully	Attribute-based	LWE
	Multiplicative	Identity-based	k-Lin
	Multiplicative	Attribute-based	k-Lin or q-ratio

DLIN stands for the decisional linear assumption. SXDH stands for the symmetric external Diffie-Hellman assumption. LWE stands for the learning with errors assumption. iO stands for the indistinguishability obfuscation. DDH stands for the decisional Diffie-Hellman assumption. DCR stands for the decisional composite residuosity assumption. q-ABDHE stands for the truncated decisional augmented bilinear Diffie-Hellman exponent assumption. Knowledge indicates the lattice-based knowledge assumption. (Q)ROM stands for the (quantum) random oracle model. k-Lin stands for the k-linear assumption. q-ratio stands for the q-ratio assumption.

matrix DDH assumption [EHK+17]. Then, as Emura et al. extended the Cramer-Shoup cryptosystem [CS98] to their KHPKE scheme, we extend PES-based ABE schemes over dual system groups [AC16, AC17, CGW15] to our proposed ABKHE schemes.

1.3 Technical Overview

In this section, we explain overviews of our proposed IBK(F)HE schemes denoted by $\Pi_{\mathsf{IBK(F)HE}}$.

Notation. For non-negative integers a and b such that $a < b$, let $[a] := \{1, 2, \ldots, a\}$ and $[a, b] := \{a, a+1, \ldots, b\}$. For a finite set S, let $s \leftarrow_R S$ denote a uniform sampling from S and $|S|$ denote the size of S. "Probabilistic polynomial time" is abbreviated as "PPT". For two security games Game_i and Game_j, $\mathsf{Game}_i \approx_c \mathsf{Game}_j$, $\mathsf{Game}_i \approx \mathsf{Game}_j$, and $\mathsf{Game}_i \equiv \mathsf{Game}_j$ indicate that Game_i and Game_j are computationally indistinguishable, statistically indistinguishable, and identically distributed, respectively.

1.3.1 Model

We briefly explain models of KHPKE, KFHE, and IBK(F)HE. See Sections 2.1 and 4 for a detailed definition. Since KHPKE and KFHE follow the same model, we explain KFHE. A KFHE scheme has three types of keys, i.e., a public key $\mathsf{KFHE.pk}$, a decryption key $\mathsf{KFHE.dk}$, and a homomorphic evaluation key $\mathsf{KFHE.hk}$. Although an encryptor encrypts a message only with $\mathsf{KFHE.pk}$, $\mathsf{KFHE.dk}$ and $\mathsf{KFHE.hk}$ are required to decrypt a ciphertext $\mathsf{KFHE.ct}$ and evaluate ciphertexts $\mathsf{KFHE.ct}^{(1)}, \ldots, \mathsf{KFHE.ct}^{(L)}$, respectively. In the KH-CCA security game, an adversary can make a homomorphic evaluation key reveal query and evaluation queries in addition to the traditional decryption queries, where the adversary can receive hk and evaluated ciphertexts by the additional queries. There are two restrictions for decryption queries to prevent trivial attacks. First, the adversary is not allowed to make decryption queries after it receives both $\mathsf{KFHE.hk}$ and the challenge ciphertext. Briefly speaking, KFHE satisfy the CCA2 security against users who do not have $\mathsf{KFHE.hk}$, while it satisfies the CCA1 security against users who have $\mathsf{KFHE.hk}$. Second, the challenger keeps a list $\mathcal{L}$ that contains the challenge ciphertext. When the adversary makes evaluation queries on ciphertexts in $\mathcal{L}$, the challenger puts evaluated ciphertexts on $\mathcal{L}$. Then, the adversary cannot make decryption queries on ciphertexts in $\mathcal{L}$.

IBK(F)HE is almost the same with some exceptions. A decryption key $\mathsf{dk}_{\mathsf{id}}$ and a homomorphic evaluation key $\mathsf{hk}_{\mathsf{id}}$ depend on an identity id. Unlike KFHE, $\mathsf{dk}_{\mathsf{id}'}$ can decrypt a ciphertext $\mathsf{ct}_{\mathsf{id}}$ and $\mathsf{hk}_{\mathsf{id}'}$ can evaluate ciphertexts $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(L)}$ only if $\mathsf{id} = \mathsf{id}'$ holds. In the security game, the adversary can make a decryption key reveal query and receive $\mathsf{dk}_{\mathsf{id}}$ as long as $\mathsf{id} \neq \mathsf{id}^\star$, where $\mathsf{id}^\star$ is the challenge identity. Even when the adversary receives $\mathsf{hk}_{\mathsf{id}}$ such that $\mathsf{id} \neq \mathsf{id}^\star$ and the challenge ciphertext, it can still make decryption queries until it receives $\mathsf{hk}_{\mathsf{id}^\star}$.

1.3.2 Overview of IBKFHE

We explain an overview of Π_{IBKFHE} based on MFHE scheme Π_{MFHE}, hierarchical IBE (HIBE) scheme Π_{HIBE}, a collision-resistant hash function H, and a one-time signature (OTS) scheme Π_{OTS}.

CCA1-*secure* FHE *Scheme.* We first review Canetti et al.'s CCA1-secure FHE scheme Π_{FHE} [CRRV17] based on Brakerski et al.'s generic construction of IBFHE [BCTW16] from MFHE and IBE. The scheme Π_{FHE} has $\mathsf{FHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk})$ and $\mathsf{FHE.sk} = \mathsf{IBE.msk}$. To encrypt a message μ, an encryptor runs the key generation algorithm of MFHE; $(\mathsf{MFHE.pk}, \mathsf{MFHE.sk}) \leftarrow \mathsf{MFHE.KGen}(\mathsf{MFHE.pp})$, samples a random identity $\mathsf{rid} \leftarrow_R \mathcal{ID}$, and computes a pre-evaluated ciphertext;

$$\mathsf{FHE.ct} = (\mathsf{rid}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{rid}}, \mathsf{MFHE.ct}),$$

where $\mathsf{IBE.ct}_{\mathsf{rid}}$ and $\mathsf{MFHE.ct}$ are encryptions of $\mathsf{MFHE.sk}$ and μ, respectively. To decrypt a pre-evaluated FHE ciphertext $\mathsf{FHE.ct}$, a decryptor computes an IBE secret key $\mathsf{IBE.sk}_{\mathsf{rid}}$ by using $\mathsf{FHE.sk} = \mathsf{IBE.msk}$, recovers an MFHE secret key $\mathsf{MFHE.sk}$ by decrypting $\mathsf{IBE.ct}_{\mathsf{rid}}$ using $\mathsf{IBE.sk}_{\mathsf{rid}}$, and recovers a message μ by decrypting $\mathsf{MFHE.ct}$ using $\mathsf{MFHE.sk}$. To evaluate pre-evaluated ciphertexts $(\mathsf{FHE.ct}^{(\ell)} = (\mathsf{rid}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{rid}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}))_{\ell \in [L]}$ for a circuit C, where $\mathsf{IBE.ct}^{(\ell)}_{\mathsf{rid}^{(\ell)}}$ and $\mathsf{MFHE.ct}^{(\ell)}$ are encryptions of $\mathsf{MFHE.sk}^{(\ell)}$ and $\mu^{(\ell)}$, respectively, an evaluator computes $\mathsf{MFHE.ct}_{\mathsf{C}}$ which is an MFHE evaluated ciphertext of $(\mathsf{MFHE.ct}^{(\ell)})_{\ell \in [L]}$ for C and outputs

$$\mathsf{FHE.ct}_{\mathsf{C}} = \left((\mathsf{rid}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{rid}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}\right).$$

To decrypt an evaluated FHE ciphertext $\mathsf{FHE.ct}_{\mathsf{C}}$, a decryptor computes IBE secret keys $\mathsf{IBE.sk}^{(\ell)}_{\mathsf{rid}^{(\ell)}}$ by using $\mathsf{FHE.sk} = \mathsf{IBE.msk}$ and recovers MFHE secret keys $\mathsf{MFHE.sk}^{(\ell)}$ by decrypting $\mathsf{IBE.ct}^{(\ell)}_{\mathsf{rid}^{(\ell)}}$ using $\mathsf{IBE.sk}^{(\ell)}_{\mathsf{rid}^{(\ell)}}$ for $\ell \in [L]$, and recovers a message $\mathsf{C}((\mu^{(\ell)})_{\ell \in [L]})$ by decrypting $\mathsf{MFHE.ct}_{\mathsf{C}}$ using $(\mathsf{MFHE.sk}^{(\ell)})_{\ell \in [L]}$.

Let $\mathsf{FHE.ct}^\star = (\mathsf{rid}^\star, \mathsf{MFHE.pk}^\star, \mathsf{IBE.ct}^\star_{\mathsf{rid}^\star}, \mathsf{MFHE.ct}^\star)$ be the challenge ciphertext. The $\mathsf{CCA1}$ security of the FHE scheme Π_{FHE} follows from the CPA security of Π_{MFHE} and Π_{IBE}. In particular, we first use the CPA security of IBE to ensure that $\mathsf{IBE.ct}^\star_{\mathsf{rid}^\star}$ is indistinguishable from encryption of a random string, then the CPA security of MFHE ensures that $\mathsf{MFHE.ct}^\star$ is indistinguishable from an encryption of a random string. We briefly explain the first reduction. In Phase 1, $\mathcal{A}$ does not know $\mathsf{rid}^\star$ sampled by $\mathcal{C}$ uniformly from an exponentially large space $\mathcal{ID}$. Thus, all ciphertexts $\mathsf{FHE.ct} = (\mathsf{rid}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{rid}}, \mathsf{MFHE.ct})$ on which the $\mathsf{CCA1}$ adversary $\mathcal{A}$ makes decryption queries satisfy $\mathsf{rid} \neq \mathsf{rid}^\star$. Therefore, the reduction algorithm of IBE can answer all decryption queries.

KH-CCA-*secure* KFHE. By modifying Π_{FHE}, we construct the first KFHE scheme Π_{KFHE} whose KH-CCA security is based solely on the LWE assumption in the standard model. At first, we apply the CHK transform [CHK04] to pre-evaluated ciphertexts so that Π_{KFHE} satisfies the $\mathsf{CCA2}$ security against an adversary without $\mathsf{KFHE.hk}$. Then, we have

$$\mathsf{KFHE.ct} = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma),$$

where a random identity rid is replaced by a verification key vk of Π_{OTS} that satisfies the strong EUF-CMA security, and σ is a signature for a message $(\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct})$. To evaluate pre-evaluated ciphertexts $(\mathsf{KFHE.ct}^{(\ell)} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)}))_{\ell \in [L]}$, we discard signatures[1] $(\sigma^{(\ell)})_{\ell \in [L]}$, apply the evaluation algorithm of Π_{FHE}, and obtain $\mathsf{KFHE.ct}_{\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$ which is the same as $\mathsf{FHE.ct}_{\mathsf{C}}$ except $\mathsf{rid}^{(\ell)}$ are replaced with $\mathsf{vk}^{(\ell)}$.

Since we do not introduce a homomorphic evaluation key hk, the current scheme is still insecure. What we have achieved so far is that the CHK transform ensures that the pre-evaluated ciphertexts $\mathsf{KFHE.ct}$ satisfy the $\mathsf{CCA2}$ security as long as it cannot be evaluated, while the $\mathsf{CCA1}$ security of Π_{FHE} ensures that the evaluated ciphertexts satisfy the $\mathsf{CCA1}$ security. Thus, we design an evaluation algorithm and a homomorphic evaluation key hk so that pre-evaluated ciphertexts cannot be evaluated without hk and evaluated ciphertexts satisfy the $\mathsf{CCA2}$ security against an adversary without hk. In other words, we only have to focus on an adversary without hk. To this end, although KFHE itself is a public key primitive, the treatment of hk is similar to a symmetric key primitive. Therefore, we use a simple encrypt-then-MAC paradigm [BN08] for constructing a $\mathsf{CCA2}$-secure symmetric key encryption scheme to design Π_{KFHE}. We set hk as a secret key of MAC and an evaluated ciphertext becomes

$$\mathsf{KFHE.ct}_{\mathsf{C}} = \left((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \tau\right),$$

where τ is a MAC tag of a message $((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$. A decryption key dk consists of $\mathsf{IBE.msk}$ and a secret key of MAC. A decryptor first checks the validity of τ and recovers a message $\mathsf{C}((\mu^{(\ell)})_{\ell \in [L]})$ in the same way as $\mathsf{FHE.ct}_{\mathsf{C}}$. Since the strong EUF-CMA security of MAC ensures that an adversary without hk cannot evaluate ciphertexts by itself, Π_{KFHE} satisfies the $\mathsf{CCA2}$ security against the adversary. Thus, Π_{KFHE} achieves the KH-CCA security.

KH-CCA-*secure* IBKFHE. Due to the simplicity of the above KFHE scheme Π_{KFHE}, we construct a KH-CCA-secure IBKFHE scheme Π_{IBKFHE} by replacing several building blocks of Π_{KFHE} with identity-based ones. In particular, we replace IBE of Π_{KFHE} by HIBE to construct $\mathsf{CCA2}$-secure IBE. Similarly, we also replace MAC with an identity-based signature (IBS) scheme, where we

[1]Since there are no $\mathsf{MFHE.ct}^{(1)}, \ldots, \mathsf{MFHE.ct}^{(L)}$ in an evaluated ciphertext $\mathsf{KFHE.ct}_{\mathsf{C}}$, the signatures $\left(\sigma^{(\ell)}\right)_{\ell \in [L]}$ are useless in the sense that we cannot verify them.

use a secret key of HIBE as a signature by following the Naor transform. However, the Naor transform is insufficient since the resulting IBS scheme does not satisfy the *strong* $\mathsf{EUF\text{-}CMA}$ security. Thus, we apply Huang et al.'s generic transformation [HWZ07] so that the identity-based signature scheme satisfies the strong $\mathsf{EUF\text{-}CMA}$ security by combining with the strongly $\mathsf{EUF\text{-}CMA}$-secure one-time signature scheme Π_{OTS}. Then, we use a two-level HIBE scheme Π_{HIBE} to play the roles of $\mathsf{CCA2}$-secure IBE and strongly $\mathsf{EUF\text{-}CMA}$-secure IBS. For an identity id, we set a decryption key $\mathsf{IBKFHE.dk}_{\mathsf{id}} = \mathsf{HIBE.sk}_{0\|\mathsf{id}}$, a homomorphic evaluation key $\mathsf{IBKFHE.hk}_{\mathsf{id}} = \mathsf{HIBE.sk}_{1\|\mathsf{id}}$, a pre-evaluated ciphertext

$$\mathsf{IBKFHE.ct}_{\mathsf{id}} = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{HIBE.ct}_{0\|\mathsf{id},\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma),$$

where $\mathsf{HIBE.ct}_{0\|\mathsf{id},\mathsf{vk}}$ and $\mathsf{MFHE.ct}$ are encryptions of $\mathsf{MFHE.sk}$ and μ, respectively, and an evaluated ciphertext

$$\mathsf{IBKFHE.ct}_{\mathsf{id},\mathsf{C}} = \begin{pmatrix} (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{HIBE.ct}^{(\ell)}_{0\|\mathsf{id},\mathsf{vk}^{(\ell)}})_{\ell\in[L]} \\ \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{HIBE.sk}_{1\|\mathsf{id},\mathsf{vk}}, \sigma \end{pmatrix},$$

where σ is a signature of $((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{HIBE.ct}^{(\ell)}_{0\|\mathsf{id},\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{HIBE.sk}_{1\|\mathsf{id},\mathsf{vk}})$ for vk and $(\mathsf{HIBE.sk}_{1\|\mathsf{id},\mathsf{vk}}, \sigma)$ plays a role of strongly $\mathsf{EUF\text{-}CMA}$-secure IBS for the message vk. The $\mathsf{KH\text{-}CCA}$ security of Π_{IBKFHE} follows from the similar discussion as the case of Π_{KFHE}.

1.3.3 Overview of IBKHE

We first review a variant of a CPA-secure ElGamal encryption scheme. Then, we review an adaptively CPA-secure IBE scheme over dual system groups Π_{DSG} [CGW15, CW14] and Emura et al.'s $\mathsf{KH\text{-}CCA}$-secure KHPKE scheme Π_{KHPKE} [EHN+18], then explain an overview of our proposed adaptively $\mathsf{KH\text{-}CCA}$-secure IBKHE scheme Π_{IBKHE}. See Sections 7.1 and 8.1.1 to check notations for cyclic groups and bilinear groups, respectively.

CPA-*secure* PKE. Let $(\mathbf{A}, \mathbf{a}^\perp) \in \mathbb{Z}_p^{(k+1)\times k} \times \mathbb{Z}_p^{k+1}$ denote an instance of the matrix distribution such that $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$. A variant of the ElGamal PKE scheme Π_{PKE} is described as follows:

$$\mathsf{PKE.pk} = ([\mathbf{A}], [\mathbf{A}^\top \mathbf{u}]), \qquad \mathsf{PKE.sk} = \mathbf{u},$$
$$\mathsf{PKE.ct} = \Big(\mathsf{PKE.ct}_0 = [\mathbf{A}\mathbf{s}], \quad \mathsf{PKE.ct}_\mu = \mu \cdot [\mathbf{s}^\top \mathbf{A}^\top \mathbf{u}]\Big),$$

where $\mathbf{u} \leftarrow_R \mathbb{Z}_p^{k+1}$ and $\mathbf{s} \leftarrow_R \mathbb{Z}_p^k$. We can correctly decrypt $\mathsf{PKE.ct} = (\mathsf{PKE.ct}_0, \mathsf{PKE.ct}_\mu)$ and recover a plaintext μ by using $\mathsf{PKE.sk}$ since we can compute $[\mathbf{s}^\top \mathbf{A}^\top \mathbf{u}]$ from $\mathsf{PKE.ct}_0$ and $\mathsf{PKE.sk}$.

To prove the CPA security, we change the challenge ciphertext to be

$$\mathsf{PKE.ct}^\star = \Big(\mathsf{PKE.ct}_0^\star = [\mathbf{c}], \quad \mathsf{PKE.ct}_\mu^\star = \mu^\star \cdot [\mathbf{c}^\top \mathbf{u}]\Big),$$

where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{k+1}$. $\mathcal{A}$ cannot detect the change under the matrix DDH assumption. Then, even an unbounded adversary $\mathcal{A}$ cannot learn $\mu^\star$ from $\mathsf{PKE.ct}^\star$. Specifically, although the unbounded $\mathcal{A}$ can learn $\widehat{\mathbf{u}}$ such that $\mathbf{u} = \widehat{\mathbf{u}} + \tilde{\alpha}\mathbf{a}^\perp$ from $[\mathbf{A}]$ and $[\mathbf{A}^\top \mathbf{u}]$, $\tilde{\alpha}$ is distributed uniformly at random over $\mathbb{Z}_p$ from $\mathcal{A}$'s view. Observe that

$$\mathsf{PKE.ct}_\mu^\star = \mu^\star \cdot [\mathbf{c}^\top \mathbf{u}] = \mu^\star \cdot [\mathbf{c}^\top (\widehat{\mathbf{u}} + \tilde{\alpha}\mathbf{a}^\perp)] = \mu^\star \cdot [\mathbf{c}^\top \widehat{\mathbf{u}}] \cdot [\mathbf{c}^\top \mathbf{a}^\perp]^{\tilde{\alpha}}. \tag{1}$$

Since $\mathbf{c}$ is distributed uniformly at random over $\mathbb{Z}_p^{k+1}$, it does not live in the span of $\mathbf{A}$, i,e., $\mathbf{c}^\top \mathbf{a}^\perp \neq \mathbf{0}$, with overwhelming probability. Thus, $[\mathbf{c}^\top \mathbf{a}^\perp]$ is a generator of $\mathbb{G}$. Therefore, $[\mathbf{c}^\top \mathbf{a}^\perp]^{\tilde{\alpha}}$ is distributed uniformly at random over $\mathbb{G}$ from $\mathcal{A}$'s view and masks $\mu^\star$.

CPA-secure IBE Scheme Π_{IBE}. We review an IBE scheme Π_{DSG} over the dual system group [CGW15, CW14] equipped with an asymmetric bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ as follows:

$$\begin{aligned}
\mathsf{IBE.mpk} &= \left(\mathsf{IBE.pp} = \begin{pmatrix} [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, [\mathbf{W}_2^\top \mathbf{A}]_1 \\ [\mathbf{B}]_2, [\mathbf{W}_1 \mathbf{B}]_2, [\mathbf{W}_2 \mathbf{B}]_2 \end{pmatrix}, [\mathbf{A}^\top \mathbf{u}]_T\right), \qquad \mathsf{IBE.msk} = \mathbf{u}, \\
\mathsf{IBE.sk_{id}} &= \left([\mathbf{Br}]_2, [\mathbf{u}]_2 \cdot [(\mathbf{W}_1 + \mathsf{id} \cdot \mathbf{W}_2)\mathbf{Br}]_2\right), \\
\mathsf{IBE.ct_{id}} &= \left(\mathsf{IBE.ct}_0 = [\mathbf{As}]_1, \mathsf{IBE.ct}_1 = [(\mathbf{W}_1^\top + \mathsf{id} \cdot \mathbf{W}_2^\top)\mathbf{As}]_1, \mathsf{IBE.ct}_T = \mu \cdot [\mathbf{s}^\top \mathbf{A}^\top \mathbf{u}]_T\right),
\end{aligned}$$

where $\mathbf{B} \in \mathbb{Z}_p^{(k+1)\times k}$ is a matrix sampled from the matrix distribution and $\mathbf{W}_1, \mathbf{W}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$. $\mathsf{IBE.mpk}$ and $\mathsf{IBE.ct_{id}}$ are similar to $\mathsf{PKE.pk}$ and $\mathsf{PKE.ct}$, respectively, except that the matrices $\mathbf{W}_1, \mathbf{W}_2$ are used to encode id. As the case of Π_{PKE}, Π_{DSG} is correct since we can recover $[\mathbf{s}^\top \mathbf{A}^\top \mathbf{u}]_T$ from $(\mathsf{IBE.ct}_0, \mathsf{IBE.ct}_1)$ and $\mathsf{IBE.sk_{id}}$ by computing

$$\frac{e(\mathsf{IBE.ct}_0, [\mathbf{u}]_2 \cdot [(\mathbf{W}_1 + \mathsf{id} \cdot \mathbf{W}_2)\mathbf{Br}]_2)}{e(\mathsf{IBE.ct}_1, [\mathbf{Br}]_2)} = [\mathbf{s}^\top \mathbf{A}^\top \mathbf{u}]_T.$$

To prove the adaptive CPA security of Π_{IBE}, we follow the proof of Π_{PKE} and change the challenge ciphertext to be

$$\mathsf{IBE.ct}^\star_{\mathsf{id}^\star} = \left(\mathsf{IBE.ct}_0 = [\mathbf{c}]_1, \mathsf{IBE.ct}_1 = [(\mathbf{W}_1^\top + \mathsf{id}^\star \cdot \mathbf{W}_2^\top)\mathbf{c}]_1, \mathsf{IBE.ct}_T = \mu^\star \cdot [\mathbf{c}^\top \mathbf{u}]_T\right), \tag{2}$$

where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{k+1}$. $\mathcal{A}$ cannot detect the change under the matrix DDH assumption over $\mathbb{G}_1$. However, unlike the case of Π_{PKE}, the unbounded $\mathcal{A}$ can still learn $\mu^\star$ since it can receive $\mathsf{IBE.sk_{id}}$ for $\mathsf{id} \neq \mathsf{id}^\star$. In particular, the unbounded $\mathcal{A}$ can learn $\mathsf{IBE.msk} = \mathbf{u}$ from $\mathsf{IBE.mpk}$ and $\mathsf{IBE.sk_{id}}$.

The dual system encryption methodology [Wat09] enables us to circumvent the issue by using the following *semi-functional* secret key

$$\mathsf{IBE.sk_{id}} = \left([\mathbf{Br}]_2, [\mathbf{u} + \alpha_{\mathsf{id}} \mathbf{a}^\perp]_2 \cdot [(\mathbf{W}_1 + \mathsf{id} \cdot \mathbf{W}_2)\mathbf{Br}]_2\right),$$

where $\alpha_{\mathsf{id}} \leftarrow_R \mathbb{Z}_p$. Briefly speaking, the semi-functional $\mathsf{IBE.sk_{id}}$ is the same as the normal one except that $\mathsf{IBE.msk} = \mathbf{u}$ is replaced with $\mathbf{u} + \alpha_{\mathsf{id}} \mathbf{a}^\perp$. After we change the challenge ciphertext to be (2), we change $\mathsf{IBE.sk_{id}}$ queried by $\mathcal{A}$ to be semi-functional one by one. When all $\mathsf{IBE.sk_{id}}$ which $\mathcal{A}$ receives become semi-functional, it cannot learn $\mathsf{IBE.msk} = \mathbf{u}$ but can learn only $\mathbf{u} + \alpha_{\mathsf{id}} \mathbf{a}^\perp$. As the proof of Π_{PKE}, $\mathcal{A}$ can learn $\widehat{\mathbf{u}}$ such that $\mathbf{u} = \widehat{\mathbf{u}} + \tilde{\alpha}\mathbf{a}^\perp$ from $[\mathbf{A}]_1$ and $[\mathbf{A}^\top \mathbf{u}]_T$. Since $\mathbf{u} + \alpha_{\mathsf{id}} \mathbf{a}^\perp$ which $\mathcal{A}$ learns from semi-functional $\mathsf{IBE.sk_{id}}$ does not help to learn $\tilde{\alpha}$, $\tilde{\alpha}$ is distributed uniformly at random over $\mathbb{Z}_p$ from $\mathcal{A}$'s view. Thus, $[\mathbf{c}^\top \mathbf{a}^\perp]^{\tilde{\alpha}}$ is distributed uniformly at random over $\mathbb{G}$ from $\mathcal{A}$'s view and masks $\mu^\star$ as the proof of Π_{PKE}.

As we discussed, we can prove the CPA security of Π_{IBE} if we can change all $\mathsf{IBE.sk_{id}}$ queried by $\mathcal{A}$ to be semi-functional. To complete the change, there is an inherent property of the dual system technique. In particular, $\mathcal{A}$ itself cannot create $\mathsf{IBE.ct_{id}}$ which follows the same distribution as (2). More specifically, $\mathcal{A}$ cannot create $\mathsf{IBE.ct_{id}} = \left(\mathsf{IBE.ct}_0 = [\mathbf{c}]_1, \mathsf{IBE.ct}_1 = [(\mathbf{W}_1^\top + \mathsf{id} \cdot \mathbf{W}_2^\top)\mathbf{c}]_1, \mathsf{IBE.ct}_T = \mu \cdot [\mathbf{c}^\top \mathbf{u}]_T\right)$ if the discrete logarithm of $\mathsf{IBE.ct}_0$, i.e., $\mathbf{c} \in \mathbb{Z}_p^{k+1}$, does not live in the span of $\mathbf{A}$, i.e., $\mathbf{c}^\top \mathbf{a}^\perp \neq \mathbf{0}$. If $\mathcal{A}$ can create such $\mathsf{IBE.ct_{id}}$, it can detect whether given $\mathsf{IBE.sk_{id}}$ is normal or semi-functional by decrypting the above

$\mathsf{IBE.ct_{id}}$, where a decryption result of $\mathsf{IBE.ct_{id}}$ by a semi-functional $\mathsf{IBE.sk_{id}}$ is not μ but $\mu\cdot[\mathbf{c}^\top\mathbf{a}^\perp]^{-\alpha_{\mathsf{id}}}$ by following the similar calculation as (1).

$\mathsf{KH\text{-}CCA}$-*secure* KHPKE. We review Emura et al.'s KHPKE scheme Π_{KHPKE} [EHN+18] by instantiating the hash proof system under the matrix DDH assumption [EHK+17] as follows:

$$\begin{aligned}
&\mathsf{KHPKE.pk} = ([\mathbf{A}], ([\mathbf{A}^\top\mathbf{u}_\iota]_{\iota\in[0,3]}), H),\\
&\mathsf{KHPKE.dk} = (\mathbf{u}_\iota)_{\iota\in[0,3]}, \qquad \mathsf{KHPKE.hk} = (\mathbf{u}_\iota)_{\iota\in[2]},\\
&\mathsf{KHPKE.ct} = \begin{pmatrix} \mathsf{KHPKE.ct}_0 = [\mathbf{A}\mathbf{s}], \quad \mathsf{KHPKE.ct}_\mu = \mu\cdot[\mathbf{s}^\top\mathbf{A}^\top\mathbf{u}_0] \\ \mathsf{KHPKE.}\pi = [\mathbf{s}^\top\mathbf{A}^\top(\mathbf{u}_1 + h\cdot\mathbf{u}_2)], \quad \mathsf{KHPKE.}\pi' = [\mathbf{s}^\top\mathbf{A}^\top\mathbf{u}_3] \end{pmatrix},
\end{aligned}$$

where $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \leftarrow_R \mathbb{Z}_p^{k+1}$, H is a collision-resistant hash function, and $h = H(\mathsf{KHPKE.ct}_0, \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi')$. Briefly speaking, $\mathsf{KHPKE.pk}$ is the same as $\mathsf{PKE.pk}$ with four secret keys $(\mathbf{u}_\iota)_{\iota\in[0,3]}$. Moreover, Π_{KHPKE} is a combination of the CCA1-secure Cramer-Shoup-lite and the CCA2-secure Cramer-Shoup cryptosystem [CS98]; Π_{KHPKE} becomes the same as the former and the latter by removing the elements depending on $(\mathbf{u}_1, \mathbf{u}_2)$ and $\mathbf{u}_3$, respectively. As the case of Π_{PKE}, Π_{KHPKE} is correct since the structure of Π_{PKE} enables us to recover $[\mathbf{s}^\top\mathbf{A}^\top\mathbf{u}_\iota]$ from $\mathsf{KHPKE.ct}_0$ and $\mathbf{u}_\iota$. Given a ciphertext $\mathsf{KHPKE.ct} = (\mathsf{KHPKE.ct}_0, \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi, \mathsf{KHPKE.}\pi')$, a decryptor first checks the validities of $\mathsf{KHPKE.}\pi$ and $\mathsf{KHPKE.}\pi'$ by using $([\mathbf{s}^\top\mathbf{A}^\top\mathbf{u}_\iota])_{\iota\in[2]}$ and $[\mathbf{s}^\top\mathbf{A}^\top\mathbf{u}_3]$, respectively. If they are valid, the decryptor recovers μ from $\mathsf{KHPKE.ct}_\mu$ and $[\mathbf{s}^\top\mathbf{A}^\top\mathbf{u}_0]$. To evaluate $\mathsf{KHPKE.ct}^{(1)} = (\mathsf{KHPKE.ct}_0^{(1)} = [\mathbf{A}\mathbf{s}^{(1)}], \mathsf{KHPKE.ct}_\mu^{(1)}, \mathsf{KHPKE.}\pi^{(1)}, \mathsf{KHPKE.}\pi'^{(1)})$ and $\mathsf{KHPKE.ct}^{(2)} = (\mathsf{KHPKE.ct}_0^{(2)} = [\mathbf{A}\mathbf{s}^{(2)}], \mathsf{KHPKE.ct}_\mu^{(2)}, \mathsf{KHPKE.}\pi^{(2)}, \mathsf{KHPKE.}\pi'(2))$, an evaluator first checks the validities of $\mathsf{KHPKE.}\pi^{(1)}$ and $\mathsf{KHPKE.}\pi^{(2)}$ by using $([(\mathbf{s}^{(1)})^\top\mathbf{A}^\top\mathbf{u}_\iota])_{\iota\in[2]}$ and $([(\mathbf{s}^{(2)})^\top\mathbf{A}^\top\mathbf{u}_\iota])_{\iota\in[2]}$, respectively. If they are valid, the evaluator computes $\mathsf{KHPKE.ct}_0 = [\mathbf{A}\mathbf{s}], \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi'$ by multiplying $\mathsf{KHPKE.ct}_0^{(1)}, \mathsf{KHPKE.ct}_\mu^{(1)}, \mathsf{KHPKE.}\pi'^{(1)}$ with $\mathsf{KHPKE.ct}_0^{(2)}, \mathsf{KHPKE.ct}_\mu^{(2)}, \mathsf{KHPKE.}\pi'^{(2)}$, respectively, and computes $\mathsf{KHPKE.}\pi$ from $h = H(\mathsf{KHPKE.ct}_0, \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi')$ and $([\mathbf{s}^\top\mathbf{A}^\top\mathbf{u}_\iota])_{\iota\in[2]}$.

Let $\mathsf{KHPKE.ct}^\star$ denote a challenge ciphertext and $\mathsf{KHPKE.ct}^{(1)} = \mathsf{KHPKE.ct}^\star, \mathsf{KHPKE.ct}^{(2)}, \dots, \mathsf{KHPKE.ct}^{(D)}$ denote ciphertexts in the list $\mathcal{L}$. To prove the KH-CCA security, we change distributions of the ciphertexts in $\mathcal{L}$ one by one so that they are independent of $\mu^\star$. Here, we explain how to change the distribution of $\mathsf{KHPKE.ct}^\star$. For this purpose, we follow the proof of Π_{PKE} and change the challenge ciphertext to be

$$\mathsf{KHPKE.ct}^\star = ([\mathbf{c}], \mu^\star\cdot[\mathbf{c}^\top\mathbf{u}_0], [\mathbf{c}^\top(\mathbf{u}_1 + h^\star\cdot\mathbf{u}_2)], [\mathbf{c}^\top\mathbf{u}_3]), \tag{3}$$

where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{k+1}$. $\mathcal{A}$ cannot detect the change under the matrix DDH assumption. We note that we do not use the above $\mathsf{KHPKE.ct}^\star$ but a normal encryption of $\mu^\star$ to compute $\mathsf{KHPKE.ct}^{(2)}, \dots, \mathsf{KHPKE.ct}^{(D)}$ in the list $\mathcal{L}$. Then, the distribution of $\mathsf{KHPKE.ct}^\star$ does not depend on $\mu^\star$ since even an unbounded $\mathcal{A}$ cannot learn $\mu^\star$ from $\mathsf{KHPKE.ct}^\star$. As the proof of Π_{PKE}, $\mathcal{A}$ can learn $\widehat{\mathbf{u}}_\iota$ such that $\mathbf{u}_\iota = \widehat{\mathbf{u}}_\iota + \tilde{\alpha}_\iota\mathbf{a}^\perp$ from $[\mathbf{A}]$ and $[\mathbf{A}^\top\mathbf{u}_\iota]$ for $\iota\in[0,3]$, respectively; however, $\tilde{\alpha}_0$ is distributed uniformly at random over $\mathbb{Z}_p$ from $\mathcal{A}$'s view. Thus, $[\mathbf{c}^\top\mathbf{a}^\perp]^{\tilde{\alpha}_0}$ is distributed uniformly at random over $\mathbb{G}$ from $\mathcal{A}$'s view and masks $\mu^\star$ as the proof of Π_{PKE}.

To ensure that the unbounded $\mathcal{A}$ cannot learn $\tilde{\alpha}_0$, we have to care about $\mathcal{A}$'s decryption queries and evaluation queries which are not allowed in the case of Π_{PKE}. We call $\mathcal{A}$'s decryption query on $\mathsf{KHPKE.ct} = (\mathsf{KHPKE.ct}_0 = [\mathbf{c}], \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi, \mathsf{KHPKE.}\pi')$ a *critical decryption query* if $\mathsf{KHPKE.}\pi$ and $\mathsf{KHPKE.}\pi'$ are valid, $\mathsf{KHPKE.ct}$ follows the same distribution as (3), and $\mathbf{c}$ does not live in the span of $\mathbf{A}$, i.e., $\mathbf{c}^\top\mathbf{a}^\perp \neq \mathbf{0}$. If $\mathcal{A}$ can make a critical decryption query, the answer is $\mu\cdot[\mathbf{c}^\top\mathbf{a}^\perp]^{\tilde{\alpha}_0}$ by following the similar calculation as (1) and $\mathcal{A}$ can learn $\tilde{\alpha}_0$. In contrast, answers to decryption queries do not reveal the information of α_0 if $\mathbf{c}$ lives in the span of $\mathbf{A}$. The structures of the CCA1-secure Cramer-Shoup-lite and the CCA2-secure Cramer-Shoup cryptosystem [CS98] ensure that $\mathcal{A}$ cannot make critical decryption queries since it cannot create valid $\mathsf{KHPKE}.\pi$ or $\mathsf{KHPKE}.\pi'$. If the unbounded $\mathcal{A}$ can create valid $\mathsf{KHPKE}.\pi$ and $\mathsf{KHPKE}.\pi'$, and make critical decryption queries, it has to know $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ and $\tilde{\alpha}_3$, respectively. We note that $\mathcal{A}$ can receive $\mathsf{KHPKE.hk} = (\mathbf{u}_1, \mathbf{u}_2)$ in the KH-CCA security game and is allowed to make decryption queries until it receives both KHPKE.hk and $\mathsf{KHPKE.ct}^\star$. Thus, all we have to ensure is that $\mathcal{A}$ does not know $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ or $\tilde{\alpha}_3$ until it receives both KHPKE.hk and $\mathsf{KHPKE.ct}^\star$. At first, $\mathcal{A}$ cannot learn $\tilde{\alpha}_3$ until it receives $\mathsf{KHPKE.ct}^\star$ thanks to the structure of the CCA1-secure Cramer-Shoup-lite [CS98]. When $\mathcal{A}$ makes a decryption query or an evaluation query on $\mathsf{KHPKE.ct} = (\mathsf{KHPKE.ct}_0, \dots)$ such that the discrete logarithm of $\mathsf{KHPKE.ct}_0$ does not live in the span of $\mathbf{A}$ and the answer is $\perp$, $\mathcal{A}$ can eliminate a candidate of $\tilde{\alpha}_3$; however, it can eliminate only polynomially many numbers of candidates throughout the security game. Thus, $\mathcal{A}$ cannot guess $\tilde{\alpha}_3$ with non-negligible probability. Next, $\mathcal{A}$ cannot learn $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ until it receives KHPKE.hk thanks to the structure of the CCA2-secure Cramer-Shoup cryptosytem [CS98]. Observe that $\mathsf{KHPKE.ct}^\star$ reveals the value of $\tilde{\alpha}_1 + h^\star \tilde{\alpha}_2$ to the unbounded $\mathcal{A}$. Thus, $\mathcal{A}$ can learn $(\tilde{\alpha}_\iota)_{\iota \in [2]}$ if it learns the value of $\tilde{\alpha}_1 + h\tilde{\alpha}_2$ for some $h \neq h^\star$. When $\mathcal{A}$ makes a decryption query on $\mathsf{KHPKE.ct} = (\mathsf{KHPKE.ct}_0, \cdots)$ such that the discrete logarithm of $\mathsf{KHPKE.ct}_0$ does not live in the span of $\mathbf{A}$ and the answer is $\perp$, $\mathcal{A}$ can eliminate a candidate of $(\tilde{\alpha}_1, \tilde{\alpha}_2)$; however, it can eliminate only polynomially many numbers of candidates throughout the security game. Thus, $\mathcal{A}$ cannot guess $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ with non-negligible probability.

KH-CCA-secure IBKHE Scheme Π_{IBKHE}. Hereafter, we explain an overview of our proposed IBKHE scheme Π_{IBKHE}. Let $\mathsf{IBE.sk_{id}}[\mathbf{u}_\iota]$ denote id's secret key of Π_{IBE} for a master secret key $\mathbf{u}_\iota$. We combine Π_{IBE} and Π_{KHPKE}, and construct Π_{IBKHE} as follows:

$$\begin{aligned}
\mathsf{mpk} &= \left(\mathsf{IBE.pp}, ([\mathbf{A}^\top \mathbf{u}_\iota]_T)_{\iota \in [0,2]}, H\right), \qquad \mathsf{msk} = (\mathbf{u}_\iota)_{\iota \in [0,2]}, \\
\mathsf{dk_{id}} &= (\mathsf{IBE.sk_{id}}[\mathbf{u}_\iota])_{\iota \in [0,2]}, \qquad \mathsf{hk_{id}} = (\mathsf{IBE.sk_{id}}[\mathbf{u}_\iota])_{\iota \in [2]}, \\
\mathsf{ct_{id}} &= \left(\mathsf{IBE.ct_{id}} = (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ct}_\mu), \pi = [\mathbf{s}^\top \mathbf{A}^\top (\mathbf{u}_1 + h \cdot \mathbf{u}_2)]_T\right),
\end{aligned}$$

where $h = H(\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ct}_\mu)$. Briefly speaking, mpk is the same as IBE.mpk with three master secret keys $(\mathbf{u}_\iota)_{\iota \in [0,2]}$, while KHPKE.pk is the same as PKE.pk with four secret keys $(\mathbf{u}_\iota)_{\iota \in [0,2]}$. As the case of Π_{KHPKE}, Π_{IBKHE} is correct since the structure of Π_{IBE} enables us to recover $[\mathbf{s}^\top \mathbf{A}^\top \mathbf{u}_\iota]_T$ from $(\mathsf{ct}_0, \mathsf{ct}_1)$ and $\mathsf{IBE.sk_{id}}[\mathbf{u}_\iota]$.

To prove the adaptive KH-CCA security, we change distributions of the ciphertexts in $\mathcal{L}$ one by one so that they are independent of $\mu^\star$ as the case of Π_{KHPKE}. Here, we explain how to change the distribution of the challenge ciphertext $\mathsf{ct}^\star_{\mathsf{id}^\star}$. As the proofs of Π_{IBE} and Π_{KHPKE}, we change the challenge ciphertext to be

$$\mathsf{ct}^\star_{\mathsf{id}^\star} = ([\mathbf{c}]_1, [(\mathbf{W}_1^\top + \mathsf{id}^\star \cdot \mathbf{W}_2^\top)\mathbf{c}]_1, \mu^\star \cdot [\mathbf{c}^\top \mathbf{u}_0]_T, [\mathbf{c}^\top (\mathbf{u}_1 + h^\star \cdot \mathbf{u}_2)]_T), \tag{4}$$

where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{k+1}$. The unbounded $\mathcal{A}$ can learn $\widehat{\mathbf{u}}_\iota$ such that $\mathbf{u}_\iota = \widehat{\mathbf{u}}_\iota + \tilde{\alpha}_\iota \mathbf{a}^\perp$ from $[\mathbf{A}]_1$ and $[\mathbf{A}^\top \mathbf{u}_\iota]_T$ for $\iota \in [0,2]$, respectively. If $\mathcal{A}$ cannot learn $\tilde{\alpha}_0$, we can prove the security. Although $\mathcal{A}$ can receive $\mathsf{dk_{id}} = (\mathsf{IBE.sk_{id}}[\mathbf{u}_\iota])_{\iota \in [0,2]}$ and still learn $\tilde{\alpha}_0$ from $\mathsf{IBE.sk_{id}}[\mathbf{u}_0]$, the dual system technique enables us to circumvent the issue by changing all normal $\mathsf{IBE.sk_{id}}[\mathbf{u}_0]$ which $\mathcal{A}$ receives to be semi-functional $\mathsf{IBE.sk_{id}}[\mathbf{u}_0 + \alpha_{0,\mathsf{id}} \mathbf{a}^\perp]$ as the case of Π_{DSG}. As the case of Π_{KHPKE}, the unbounded $\mathcal{A}$ may be able to learn $\tilde{\alpha}_0$ via decryption queries.

We call $\mathcal{A}$'s decryption query on $\mathsf{ct_{id}} = (\mathsf{ct}_0 = [\mathbf{c}]_1, \mathsf{ct}_1, \mathsf{ct}_\mu, \pi)$ a *critical decryption query* if π is valid, ct follows the same distribution as (4), and $\mathbf{c}$ does not live in the span of $\mathbf{A}$, i.e., $\mathbf{c}^\top \mathbf{a}^\perp \neq \mathbf{0}$. As the case of Π_{KHPKE}, all we have to ensure is that $\mathcal{A}$ cannot make critical decryption queries until it receives both $\mathsf{hk}_{\mathsf{id}^\star}$ and $\mathsf{ct}^\star_{\mathsf{id}^\star}$. Observe that the unbounded $\mathcal{A}$ can make critical decryption queries since it can receive $(\mathsf{IBE.sk}_{\mathsf{id}}[\mathbf{u}_\iota])_{\iota\in[2]}$ unlike the case of Π_{KHPKE}. On the surface, the dual system technique seems to be sufficient to circumvent the issue by changing all normal $(\mathsf{IBE.sk}_{\mathsf{id}}[\mathbf{u}_\iota])_{\iota\in[2]}$ which $\mathcal{A}$ receives to be semi-functional $(\mathsf{IBE.sk}_{\mathsf{id}}[\mathbf{u}_\iota])_{\iota\in[2]}$; however, we cannot take the approach directly since $\mathcal{A}$ can receive $\mathsf{hk}_{\mathsf{id}^\star} = (\mathsf{IBE.sk}_{\mathsf{id}^\star}[\mathbf{u}_\iota])_{\iota\in[2]}$ which we cannot change to be semi-functional. Moreover, even when $\mathsf{id} \neq \mathsf{id}^\star$ holds, we cannot also change $\mathsf{hk}_{\mathsf{id}} = (\mathsf{IBE.sk}_{\mathsf{id}}[\mathbf{u}_\iota])_{\iota\in[2]}$ which $\mathcal{A}$ receives in Phase 1 to be semi-functional since we cannot detect whether $\mathsf{id} \neq \mathsf{id}^\star$ holds.

To circumvent the issue, we divide $\mathcal{A}$'s attack strategies into two types. We call a strategy Type-1 if $\mathcal{A}$ receives $\mathsf{hk}_{\mathsf{id}^\star}$ in Phase 1 and Type-2 otherwise. To prove the security against $\mathcal{A}$ of Type-2, we change all normal $(\mathsf{IBE.sk}_{\mathsf{id}}[\mathbf{u}_\iota])_{\iota\in[2]}$ which $\mathcal{A}$ receives to be semi-functional $(\mathsf{IBE.sk}_{\mathsf{id}}[\mathbf{u}_\iota+\alpha_{\iota,\mathsf{id}}\mathbf{a}^\perp])_{\iota\in[2]}$ until $\mathcal{A}$'s query to receive $\mathsf{hk}_{\mathsf{id}^\star}$. Since the definition of the Type-2 strategy ensures that $\mathcal{A}$ queries to receive $\mathsf{hk}_{\mathsf{id}^\star}$ only in Phase 2, we can detect whether $\mathsf{id} \neq \mathsf{id}^\star$ holds and complete the change. Since $\mathcal{A}$ cannot learn $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ until it receives both $\mathsf{hk}_{\mathsf{id}^\star}$ and $\mathsf{ct}^\star_{\mathsf{id}^\star}$, it cannot create valid π and make critical decryption queries. To prove the security against $\mathcal{A}$ of Type-1, we cannot change $(\mathsf{IBE.sk}_{\mathsf{id}}[\mathbf{u}_\iota])_{\iota\in[2]}$ which $\mathcal{A}$ receives to be semi-functional since we cannot detect whether $\mathsf{id} \neq \mathsf{id}^\star$ holds upon $\mathcal{A}$'s queries to receive $\mathsf{hk}_{\mathsf{id}}$. Although we ensured that $\mathcal{A}$ cannot create $\mathsf{KHPKE}.\pi'$ and make critical decryption queries in the case of Π_{KHPKE}, there does not seem to be the corresponding element in $\mathsf{ct}_{\mathsf{id}}$ on the surface. However, the inherent property of the dual system technique ensures that $\mathcal{A}$ cannot make critical decryption queries. In particular, since $\mathcal{A}$ against Π_{DSG} cannot create $\mathsf{IBE.ct}_{\mathsf{id}}$ to make critical decryption queries, $\mathcal{A}$ of Type-1 cannot also create $\mathsf{ct}_{\mathsf{id}} = (\mathsf{IBE.ct}_{\mathsf{id}}, \pi)$ and make critical decryption queries. Thus, we can prove the adaptive KH-CCA security of Π_{IBKHE} against $\mathcal{A}$ of both types as the case of Π_{KHPKE}.

1.4 Organization

We aim to provide a generic construction of ABKFHE in Sections 3–6 and a pairing-based construction of ABKHE in Sections 7 and 8. In Section 2, we review cryptographic primitives which we will use in this paper. In Section 3, we propose a generic construction of KFHE. In Section 4, we extend the definition of IBKHE [EHN+18] and define ABK(F)HE. In Section 5, we define delegatable ABE and provide a concrete construction under the LWE assumption. In Section 6, we propose a generic construction of ABKFHE whose building blocks can be instantiated under the LWE assumption. In Section 7, we revisit Emura et al.'s KHPKE scheme and give a simpler proof under the matrix DDH assumption. In Section 8, we propose an efficient pairing-based ABKHE from pair encoding schemes.

2 Cryptographic Primitives

2.1 Keyed Fully Homomorphic Encryption

A *keyed fully homomorphic encryption* (KFHE) scheme consists of four polynomial-time algorithms $\Pi_{\mathsf{KFHE}} = (\mathsf{KFHE.KGen}, \mathsf{KFHE.Enc}, \mathsf{KFHE.Eval}, \mathsf{KFHE.Dec})$ defined as follows.

$\mathsf{KFHE.KGen}(1^\lambda) \to (\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.hk})$. On input the security parameter 1^λ, it outputs a public key KFHE.pk a decryption key KFHE.dk, and a homomorphic evaluation key KFHE.hk, where KFHE.pk implicitly contains a message space $\mathcal{M}$.

$\mathsf{KFHE.Enc}(\mathsf{KFHE.pk}, \mu) \to \mathsf{KFHE.ct}$. On input a KFHE.pk and a message $\mu \in \mathcal{M}$, it outputs a pre-evaluated ciphertext KFHE.ct.

$\mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C}) \to \mathsf{KFHE.ct}_{\mathsf{C}}/\perp$. On input a $\mathsf{KFHE.pk}$, $\mathsf{KFHE.hk}$, a tuple of L ciphertexts $(\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}$, and a circuit $\mathsf{C} : \mathcal{M}^L \to \mathcal{M}$, it outputs an evaluated ciphertext $\mathsf{KFHE.ct}_{\mathsf{C}}$ or a rejection symbol $\perp$.

$\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.ct}/\mathsf{KFHE.ct}_{\mathsf{C}}) \to \mu/\perp$. On input a $\mathsf{KFHE.pk}$, $\mathsf{KFHE.dk}$ and $\mathsf{KFHE.ct}/\mathsf{KFHE.ct}_{\mathsf{C}}$, it outputs a decryption result $\mu \in \mathcal{M}$ or a rejection symbol $\perp$.

Remark 1. *A* keyed homomorphic public key encryption (KHPKE) *scheme* $\Pi_{\mathsf{KHPKE}} = (\mathsf{KHPKE.KGen}, \mathsf{KHPKE.Enc}, \mathsf{KHPKE.Eval}, \mathsf{KHPKE.Dec})$ *is defined in the same way except that* $\mathsf{KHPKE.Eval}$ *does not take a circuit* C *as input since a* KHPKE *scheme supports only either multiplicative or additive homomorphism.*

Definition 1 (Correctness)**.** $\Pi_{\mathsf{KFHE}} = (\mathsf{KFHE.KGen}, \mathsf{KFHE.Enc}, \mathsf{KFHE.Eval}, \mathsf{KFHE.Dec})$ *satisfies correctness if the following conditions hold with overwhelming probability:*

- *For every* $(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.hk}) \leftarrow \mathsf{KFHE.KGen}(1^\lambda)$ *and* $\mu \in \mathcal{M}$*, it holds that* $\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.Enc}(\mathsf{KFHE.pk}, \mu)) = \mu$*.*
- *For every* $(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.hk}) \leftarrow \mathsf{KFHE.KGen}(1^\lambda)$*, circuit* $\mathsf{C} : \mathcal{M}^L \to \mathcal{M}$*, and* $(\mu^{(1)}, \ldots, \mu^{(L)}) \in \mathcal{M}^L$*, it holds that* $\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.ct}_{\mathsf{C}}) = \mathsf{C}(\mu^{(1)}, \ldots, \mu^{(L)})$*, where* $\mathsf{KFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$ *and* $\mathsf{KFHE.ct}^{(\ell)} \leftarrow \mathsf{KFHE.Enc}(\mathsf{KFHE.pk}, \mu^{(\ell)})$ *for every* $\ell \in [L]$*.*

Definition 2 (Compactness)**.** $\Pi_{\mathsf{KFHE}} = (\mathsf{KFHE.KGen}, \mathsf{KFHE.Enc}, \mathsf{KFHE.Eval}, \mathsf{KFHE.Dec})$ *satisfies compactness if there exists a polynomial* poly *such that* $|\mathsf{KFHE.ct}_{\mathsf{C}}|$*, where* $\mathsf{KFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$*, is independent of the size and depth of* C *and at most* $L \cdot \mathsf{poly}(\lambda)$ *for every security parameter* λ*.*

Although we follow the syntax, correctness, and compactness of KFHE by following previous works [EHN+18, SET22], we introduce a slightly stronger notion of the KH-CCA security. Specifically, to introduce as strong requirement as possible, we consider the case that a pre-evaluated ciphertext $\mathsf{KFHE.ct}$ and an evaluated ciphertext $\mathsf{KFHE.ct}_{\mathsf{C}}$ follow distinct distributions which are easily detectable. Our proposed KFHE scheme in Section 3 and ABKFHE scheme in Section 6 satisfy the condition.

Definition 3 (KH-CCA security)**.** *The* KH-CCA *security of* $\Pi_{\mathsf{KFHE}} = (\mathsf{KFHE.KGen}, \mathsf{KFHE.Enc}, \mathsf{KFHE.Eval}, \mathsf{KFHE.Dec})$ *is defined by the security game between a challenger* $\mathcal{C}$ *and an adversary* $\mathcal{A}$ *as follows.*

Init. $\mathcal{C}$ *runs* $(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.hk}) \leftarrow \mathsf{KFHE.KGen}(1^\lambda)$ *and sends* $\mathsf{KFHE.pk}$ *to* $\mathcal{A}$*.*

Phase 1. $\mathcal{A}$ *is allowed to make the following three types of queries to* $\mathcal{C}$*.*

Homomorphic Evaluation Key Reveal Query. *Upon* $\mathcal{A}$*'s query,* $\mathcal{C}$ *sends* $\mathsf{KFHE.hk}$ *to* $\mathcal{A}$*.*

Evaluation Query. *Upon* $\mathcal{A}$*'s query on* $((\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$*,* $\mathcal{C}$ *sends the result of* $\mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$ *to* $\mathcal{A}$*.*

Decryption Query. *Upon* $\mathcal{A}$*'s query on* $\mathsf{KFHE.ct}/\mathsf{KFHE.ct}_{\mathsf{C}}$*,* $\mathcal{C}$ *sends the result of* $\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.ct}/\mathsf{KFHE.ct}_{\mathsf{C}})$ *to* $\mathcal{A}$*.*

Challenge Query. $\mathcal{A}$ *is allowed to make the query only once. Upon* $\mathcal{A}$*'s query on* $(\mu_0^\star, \mu_1^\star)$ *such that* $|\mu_0^\star| = |\mu_1^\star|$*,* $\mathcal{C}$ *samples* $\mathsf{coin} \leftarrow_R \{0,1\}$*, runs* $\mathsf{KFHE.ct}^\star \leftarrow \mathsf{KFHE.Enc}(\mathsf{KFHE.pk}, \mu_{\mathsf{coin}}^\star)$*, creates a list of ciphertexts* $\mathcal{L} = \{\mathsf{KFHE.ct}^\star\}$*, and sends* $\mathsf{KFHE.ct}^\star$ *to* $\mathcal{A}$*.*

Phase 2. *$\mathcal{A}$ is allowed to make the same three types of queries to $\mathcal{C}$ as in Phase 1 with the following exceptions.*

Evaluation Query. *If $\{\mathsf{KFHE.ct}^{(\ell)}\}_{\ell\in[L]} \cap \mathcal{L} \neq \emptyset$ holds and the evaluation result is not $\perp$ but $\mathsf{KFHE.ct_C}$, $\mathcal{C}$ updates a list $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathsf{KFHE.ct_C}\}$.*

Decryption Query. *Upon $\mathcal{A}$'s query on $\mathsf{KFHE.ct}$, $\mathcal{C}$ outputs $\perp$ if $\mathsf{KFHE.ct} = \mathsf{KFHE.ct}^\star$ holds. Upon $\mathcal{A}$'s query on $\mathsf{KFHE.ct_C}$, $\mathcal{C}$ outputs $\perp$ if $\mathsf{KFHE.ct_C} \in \mathcal{L}$ holds. $\mathcal{C}$ also outputs $\perp$ if $\mathcal{A}$ has already made a homomorphic evaluation key reveal query.*

Guess. *$\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0,1\}$ as a guess of coin and terminates the game.*

If the advantage of $\mathcal{A}$ for breaking the $\mathsf{KH\text{-}CCA}$ security of Π_{KFHE} defined by $\mathsf{Adv}^{\mathsf{KH\text{-}CCA}}_{\Pi_{\mathsf{KFHE}},\mathcal{A}}(\lambda) := \left|\Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2}\right|$ is negligible in λ, Π_{KFHE} is said to satisfy the $\mathsf{KH\text{-}CCA}$ security.

Remark 2. *If a pre-evaluated ciphertext $\mathsf{KFHE.ct}$ and an evaluated ciphertext $\mathsf{KFHE.ct_C}$ follow the same distribution, we change the restriction of decryption queries in Phase 2:*

Decryption Query. *Upon $\mathcal{A}$'s query on $\mathsf{KFHE.ct}$, $\mathcal{C}$ outputs $\perp$ if $\mathsf{KFHE.ct} \in \mathcal{L}$ holds. Otherwise, $\mathcal{C}$ proceeds the same way as in Phase 1.*

Specifically, in Definition 3, the adversary is allowed to make a decryption query on a pre-evaluated ciphertext $\mathsf{KFHE.ct} \neq \mathsf{KFHE.ct}^\star$ in Phase 2 even after $\mathcal{A}$'s homomorphic evaluation key reveal query. When a pre-evaluated ciphertext $\mathsf{KFHE.ct}$ and an evaluated ciphertext $\mathsf{KFHE.ct_C}$ follow the same distribution, we have to prohibit such queries since the queried $\mathsf{KFHE.ct}$ may be an evaluation result of $\mathsf{KFHE.ct}^\star$ by $\mathsf{KFHE.hk}$.

Remark 3. *We call $\mathcal{A}$'s evaluation query on $(\mathsf{KHPKE.ct}^{(\ell)})_{\ell\in[L]}$ a* dependent evaluation query *if the answer is stored in $\mathcal{L}$. In other words, $\mathcal{A}$'s dependent evaluation query on $(\mathsf{KHPKE.ct}^{(\ell)})_{\ell\in[L]}$ satisfies $\{\mathsf{KHPKE.ct}^{(\ell)}\}_{\ell\in[L]} \cap \mathcal{L} \neq \emptyset$. Otherwise, we call $\mathcal{A}$'s evaluation query on $(\mathsf{KHPKE.ct}^{(\ell)})_{\ell\in[L]}$ an* independent evaluation query*.*

2.2 Multi-Key Fully Homomorphic Encryption

A *multi-key fully homomorphic encryption* (MFHE) scheme consists of five polynomial-time algorithms $\Pi_{\mathsf{MFHE}} = (\mathsf{MFHE.Setup}, \mathsf{MFHE.KGen}, \mathsf{MFHE.Enc}, \mathsf{MFHE.Dec}, \mathsf{MFHE.Eval})$ defined as follows.

$\mathsf{MFHE.Setup}(1^\lambda) \rightarrow \mathsf{MFHE.pp}$. On input the security parameter 1^λ, it outputs a public parameter $\mathsf{MFHE.pp}$. Although we do not explicitly describe, the following algorithms take $\mathsf{MFHE.pp}$ as input.

$\mathsf{MFHE.KGen} \rightarrow (\mathsf{MFHE.pk}, \mathsf{MFHE.sk})$. It outputs a public/secret key pair $(\mathsf{MFHE.pk}, \mathsf{MFHE.sk})$.

$\mathsf{MFHE.Enc}(\mathsf{MFHE.pk}, \mu) \rightarrow \mathsf{MFHE.ct}$. On input $\mathsf{MFHE.pk}$ and a message μ, it outputs a pre-evaluated ciphertext $\mathsf{MFHE.ct}$.

$\mathsf{MFHE.Dec}(\mathsf{MFHE.sk}, \mathsf{MFHE.ct}) \rightarrow \mu/\perp$. On input a secret key $\mathsf{MFHE.sk}$ and a pre-evaluated ciphertext $\mathsf{MFHE.ct}$, it outputs a decryption result μ or a failure symbol $\perp$.

$\mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C}) \rightarrow \mathsf{MFHE.ct_C}$. On input L public key/ciphertext pairs $(\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell\in[L]}$ and a circuit C, it outputs an evaluated ciphertext $\mathsf{MFHE.ct_C}$.

$\mathsf{MFHE.Dec}((\mathsf{MFHE.sk}^{(\ell)})_{\ell\in[L]}, \mathsf{MFHE.ct}_\mathsf{C}) \to \mu/\bot$. On input L secret keys $(\mathsf{MFHE.sk}^{(\ell)})_{\ell\in[L]}$ and an evaluated ciphertext $\mathsf{MFHE.ct}_\mathsf{C}$, it outputs a decryption result μ or a failure symbol $\bot$.

Definition 4 (Correctness)**.** $\Pi_{\mathsf{MFHE}} = (\mathsf{MFHE.Setup}, \mathsf{MFHE.KGen}, \mathsf{MFHE.Enc}, \mathsf{MFHE.Dec}, \mathsf{MFHE.Eval})$ *satisfies correctness if the following conditions hold with overwhelming probability:*

- *For every* $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$*,* $(\mathsf{MFHE.pk}, \mathsf{MFHE.sk}) \leftarrow \mathsf{MFHE.KGen}$*, and* $\mu \in \mathcal{M}$*, it holds that* $\mathsf{MFHE.Dec}(\mathsf{MFHE.sk}, \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}, \mu)) = \mu$*.*
- *For every* $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$*,* $(\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.sk}^{(\ell)}) \leftarrow \mathsf{MFHE.KGen}$ *for* $\ell \in [L]$*, a circuit* $\mathsf{C} : \mathcal{M}^L \to \mathcal{M}$*, and* $(\mu^{(1)}, \dots, \mu^{(L)}) \in \mathcal{M}^L$*, it holds that* $\mathsf{MFHE.Dec}((\mathsf{MFHE.sk}^{(\ell)})_{\ell\in[L]}, \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})) = \mathsf{C}(\mu^{(1)}, \dots, \mu^{(L)})$*, where* $\mathsf{KFHE.ct}^{(\ell)} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^{(\ell)}, \mu^{(\ell)})$ *for* $\ell \in [L]$*.*

Definition 5 (Compactness)**.** $\Pi_{\mathsf{MFHE}} = (\mathsf{MFHE.Setup}, \mathsf{MFHE.KGen}, \mathsf{MFHE.Enc}, \mathsf{MFHE.Dec}, \mathsf{MFHE.Eval})$ *satisfies compactness if there exists a polynomial* poly *such that* $|\mathsf{MFHE.ct}_\mathsf{C}|$*, where* $\mathsf{KFHE.ct}_\mathsf{C} \leftarrow \mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$*, is independent of the size and depth of* C *and at most* $L \cdot \mathsf{poly}(\lambda)$ *for every security parameter* λ*.*

Definition 6 (IND-CPA Security)**.** *The* IND-CPA *security of* $\Pi_{\mathsf{MFHE}} = (\mathsf{MFHE.Setup}, \mathsf{MFHE.KGen}, \mathsf{MFHE.Enc}, \mathsf{MFHE.Dec}, \mathsf{MFHE.Eval})$ *is defined by the security game between a challenger* $\mathcal{C}$ *and an adversary* $\mathcal{A}$ *as follows.*

Init. $\mathcal{C}$ *runs* $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$ *and* $(\mathsf{MFHE.pk}, \mathsf{MFHE.sk}) \leftarrow \mathsf{MFHE.KGen}$*, and sends* $(\mathsf{MFHE.pp}, \mathsf{MFHE.pk})$ *to* $\mathcal{A}$*.*

Challenge Query. $\mathcal{A}$ *is allowed to make the query only once. Upon* $\mathcal{A}$*'s query on* $(\mu_0^\star, \mu_1^\star)$ *such that* $|\mu_0^\star| = |\mu_1^\star|$*,* $\mathcal{C}$ *samples* $\mathsf{coin} \leftarrow_R \{0,1\}$*, runs* $\mathsf{MFHE.ct}^\star \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}, \mu^\star_{\mathsf{coin}})$*, and sends the challenge cipehrtext* $\mathsf{MFHE.ct}^\star$ *to* $\mathcal{A}$*.*

Guess. $\mathcal{A}$ *outputs* $\widehat{\mathsf{coin}} \in \{0,1\}$ *as a guess of* coin *and terminates the game.*

If the advantage of $\mathcal{A}$ *for breaking the* IND-CPA *security of* Π_{MFHE} *defined by* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{MFHE}},\mathcal{A}}(\lambda) := \left|\Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2}\right|$ *is negligible in* λ*,* Π_{MFHE} *is said to satisfy the* IND-CPA *security.*

2.3 Identity-based Encryption

An *identity-based encryption* (IBE) scheme with an identity space $\mathcal{ID}$ consists of four polynomial-time algorithms $\Pi_{\mathsf{IBE}} = (\mathsf{IBE.Setup}, \mathsf{IBE.KGen}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ defined as follows.

$\mathsf{IBE.Setup}(1^\lambda) \to (\mathsf{IBE.mpk}, \mathsf{IBE.msk})$. On input the security parameter 1^λ, it outputs a master public/secret key pair $(\mathsf{IBE.mpk}, \mathsf{IBE.msk})$, where $\mathsf{IBE.mpk}$ implicitly contains a message space $\mathcal{M}$. Although we do not explicitly describe, the following algorithms take $\mathsf{IBE.mpk}$ as input.

$\mathsf{IBE.Enc}(\mathsf{id}, \mu) \to \mathsf{IBE.ct}_\mathsf{id}$. On input an identity $\mathsf{id} \in \mathcal{ID}$ and a message $\mu \in \mathcal{M}$, it outputs a ciphertext $\mathsf{IBE.ct}_\mathsf{id}$ for id.

$\mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{id}) \to \mathsf{IBE.sk}_\mathsf{id}$. On input a master secret key $\mathsf{IBE.msk}$, it outputs a secret key $\mathsf{IBE.sk}_\mathsf{id}$ for id.

$\mathsf{IBE.Dec}(\mathsf{IBE.sk}_\mathsf{id}, \mathsf{IBE.ct}_\mathsf{id}) \to \mu/\bot$. On input $\mathsf{IBE.sk}_\mathsf{id}$ and $\mathsf{IBE.ct}_\mathsf{id}$, it outputs a decryption result μ or a failure symbol $\bot$.

Definition 7 (Correctness). *$\Pi_{\mathsf{IBE}} = (\mathsf{IBE.Setup}, \mathsf{IBE.KGen}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ is said to satisfy the correctness if for every $\mu \in \mathcal{M}$, $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$, and $\mathsf{id} \in \mathcal{ID}$, it holds that $\mu \leftarrow \mathsf{IBE.Dec}(\mathsf{IBE.sk}_{\mathsf{id}}, \mathsf{IBE.ct}_{\mathsf{id}})$ with overwhelming probability, where $\mathsf{IBE.ct}_{\mathsf{id}} \leftarrow \mathsf{IBE.Enc}(\mathsf{id}, \mu)$ and $\mathsf{IBE.sk}_{\mathsf{id}} \leftarrow \mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{id})$.*

Definition 8 (Adaptive IND-CPA Security). *The adaptive IND-CPA security of $\Pi_{\mathsf{IBE}} = (\mathsf{IBE.Setup}, \mathsf{IBE.KGen}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ is defined by the security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows.*

Init. *$\mathcal{C}$ runs $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and sends $\mathsf{IBE.mpk}$ to $\mathcal{A}$.*

Phase 1. *$\mathcal{A}$ is allowed to make the following secret key reveal queries to $\mathcal{C}$.*

> **Secret Key Reveal Query.** *Upon $\mathcal{A}$'s query on $\mathsf{id} \in \mathcal{ID}$, $\mathcal{C}$ runs $\mathsf{IBE.sk}_{\mathsf{id}} \leftarrow \mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{id})$ and sends $\mathsf{IBE.sk}_{\mathsf{id}}$ to $\mathcal{A}$.*

Challenge Query. *$\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $(\mathsf{id}^\star, \mu_0^\star, \mu_1^\star)$ such that $|\mu_0^\star| = |\mu_1^\star|$, $\mathcal{C}$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$, runs $\mathsf{IBE.ct}^\star_{\mathsf{id}^\star} \leftarrow \mathsf{IBE.Enc}(\mathsf{id}^\star, \mu^\star_{\mathsf{coin}})$, and sends the challenge cipehrtext $\mathsf{IBE.ct}^\star_{\mathsf{id}^\star}$ to $\mathcal{A}$.*

Phase 2. *$\mathcal{A}$ is allowed to make secret key reveal queries as in Phase 1 except that $\mathcal{C}$ outputs $\perp$ if $\mathsf{id} = \mathsf{id}^\star$ holds.*

Guess. *$\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0,1\}$ as a guess of coin and terminates the game.*

If the advantage of $\mathcal{A}$ for breaking the adaptive IND-CPA security of Π_{IBE} defined by $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{IBE}},\mathcal{A}}(\lambda) := \left|\Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 0\right] - \Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 1\right]\right|$ is negligible in λ, Π_{IBE} is said to satisfy the adaptive IND-CPA security.

Definition 9 (Adaptive OW-CPA Security). *The adaptive OW-CPA security of $\Pi_{\mathsf{IBE}} = (\mathsf{IBE.Setup}, \mathsf{IBE.KGen}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ is defined by the security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows.*

Init. *$\mathcal{C}$ runs $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and sends $\mathsf{IBE.mpk}$ to $\mathcal{A}$.*

Phase 1. *$\mathcal{A}$ is allowed to make the following secret key reveal queries to $\mathcal{C}$.*

> **Secret Key Reveal Query.** *Upon $\mathcal{A}$'s query on $\mathsf{id} \in \mathcal{ID}$, $\mathcal{C}$ runs $\mathsf{IBE.sk}_{\mathsf{id}} \leftarrow \mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{id})$ and sends $\mathsf{IBE.sk}_{\mathsf{id}}$ to $\mathcal{A}$.*

Challenge Query. *$\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $\mathsf{id}^\star$, $\mathcal{C}$ samples $\mu^\star \leftarrow_R \mathcal{M}$, runs $\mathsf{IBE.ct}^\star_{\mathsf{id}^\star} \leftarrow \mathsf{IBE.Enc}(\mathsf{id}^\star, \mu^\star)$, and sends the challenge cipehrtext $\mathsf{IBE.ct}^\star_{\mathsf{id}^\star}$ to $\mathcal{A}$.*

Phase 2. *$\mathcal{A}$ is allowed to make secret key reveal queries as in Phase 1 except that $\mathcal{C}$ outputs $\perp$ if $\mathsf{id} = \mathsf{id}^\star$ holds.*

Guess. *$\mathcal{A}$ outputs $\widehat{\mu} \in \mathcal{M}$ as a guess of $\mu^\star$ and terminates the game.*

If the advantage of $\mathcal{A}$ for breaking the adaptive OW-CPA security of Π_{IBE} defined by $\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\Pi_{\mathsf{IBE}},\mathcal{A}}(\lambda) := \left|\Pr\left[\widehat{\mu} = \mu\right] - \frac{1}{|\mathcal{M}|}\right|$ is negligible in λ, Π_{IBE} is said to satisfy the adaptive OW-CPA security.

2.4 Attribute-based Encryption

An *attribute-based encryption* (ABE) scheme for a predicate $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ consists of four polynomial-time algorithms $\Pi_{\mathsf{ABE}} = (\mathsf{ABE.Setup}, \mathsf{ABE.KGen}, \mathsf{ABE.Enc}, \mathsf{ABE.Dec})$ defined as follows.

$\mathsf{ABE.Setup}(1^\lambda) \to (\mathsf{ABE.mpk}, \mathsf{ABE.msk})$. On input the security parameter 1^λ, it outputs a master public/secret key pair $(\mathsf{ABE.mpk}, \mathsf{ABE.msk})$, where $\mathsf{ABE.mpk}$ implicitly contains a message space $\mathcal{M}$. Although we do not explicitly describe, the following algorithms take $\mathsf{ABE.mpk}$ as input.

$\mathsf{ABE.Enc}(x, \mu) \to \mathsf{ABE.ct}_x$. On input a ciphertext attribute $x \in \mathcal{X}$ and a message μ, it outputs a ciphertext $\mathsf{ABE.ct}_x$ for x.

$\mathsf{ABE.KGen}(\mathsf{ABE.msk}, y) \to \mathsf{ABE.sk}_y$. On input a master secret key $\mathsf{DABE.msk}$ and a key attribute $y \in \mathcal{Y}$, it outputs a secret key $\mathsf{ABE.sk}_y$ for y.

$\mathsf{ABE.Dec}(\mathsf{ABE.sk}_y, \mathsf{ABE.ct}_x) \to \mu/\bot$. On input $\mathsf{ABE.sk}_y$ and $\mathsf{ABE.ct}_x$, it outputs a decryption result μ or a failure symbol $\bot$.

Definition 10 (Correctness)**.** $\Pi_{\mathsf{DABE}} = (\mathsf{ABE.Setup}, \mathsf{ABE.KGen}, \mathsf{ABE.Enc}, \mathsf{ABE.Dec})$ *is said to satisfy the correctness if for every* $\mu \in \mathcal{M}$*,* $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$*, and* $(x, y) \in \mathcal{X} \times \mathcal{Y}$ *such that* $f(x, y) = 1$*, it holds that* $\mu \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.sk}_y, \mathsf{ABE.ct}_x)$ *with overwhelming probability, where* $\mathsf{ABE.ct}_x \leftarrow \mathsf{ABE.Enc}(x, \mu)$ *and* $\mathsf{ABE.sk}_y \leftarrow \mathsf{ABE.KGen}(\mathsf{ABE.msk}, y)$*.*

Definition 11 (Selective $\mathsf{IND\text{-}CPA}$ Security)**.** *The selective* $\mathsf{IND\text{-}CPA}$ *security of* $\Pi_{\mathsf{ABE}} = (\mathsf{ABE.Setup}, \mathsf{ABE.KGen}, \mathsf{ABE.Enc}, \mathsf{ABE.Dec})$ *is defined by the security game between a challenger* $\mathcal{C}$ *and an adversary* $\mathcal{A}$ *as follows.*

Init. $\mathcal{A}$ *declares a challenge ciphertext attribute* $x^\star$ *to* $\mathcal{C}$*. Then,* $\mathcal{C}$ *runs* $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$ *and sends* $\mathsf{ABE.mpk}$ *to* $\mathcal{A}$*.*

Phase 1. $\mathcal{A}$ *is allowed to make the following secret key reveal queries to* $\mathcal{C}$*.*

Secret Key Reveal Query. *Upon* $\mathcal{A}$*'s query on* $y \in \mathcal{Y}$*,* $\mathcal{C}$ *outputs* $\bot$ *if* $f(x^\star, y) = 1$ *holds. Otherwise,* $\mathcal{C}$ *runs* $\mathsf{ABE.sk}_y \leftarrow \mathsf{ABE.KGen}(\mathsf{ABE.msk}, y)$ *and sends* $\mathsf{ABE.sk}_y$ *to* $\mathcal{A}$*.*

Challenge Query. $\mathcal{A}$ *is allowed to make the query only once. Upon* $\mathcal{A}$*'s query on* $(\mu_0^\star, \mu_1^\star)$ *such that* $|\mu_0^\star| = |\mu_1^\star|$*,* $\mathcal{C}$ *samples* $\mathsf{coin} \leftarrow_R \{0,1\}$*, runs* $\mathsf{ABE.ct}_{x^\star}^\star \leftarrow \mathsf{ABE.Enc}(x^\star, \mu_{\mathsf{coin}}^\star)$*, and sends the challenge cipehrtext* $\mathsf{ABE.ct}_{x^\star}^\star$ *to* $\mathcal{A}$*.*

Phase 2. $\mathcal{A}$ *is allowed to make secret key reveal queries as in Phase 1.*

Guess. $\mathcal{A}$ *outputs* $\widehat{\mathsf{coin}} \in \{0,1\}$ *as a guess of* coin *and terminates the game.*

If the advantage of $\mathcal{A}$ *for breaking the selective* $\mathsf{IND\text{-}CPA}$ *security of* Π_{ABE} *defined by* $\mathsf{Adv}_{\Pi_{\mathsf{ABE}},\mathcal{A}}^{\mathsf{IND\text{-}CPA}}(\lambda) := \left|\Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 0\right] - \Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 1\right]\right|$ *is negligible in* λ*,* Π_{ABE} *is said to satisfy the selective* $\mathsf{IND\text{-}CPA}$ *security.*

2.5 One-time Signatures

A *one-time signature* (OTS) scheme consists of three polynomial-time algorithms $\Pi_{\mathsf{OTS}} = (\mathsf{OTS.KGen}, \mathsf{OTS.Sign}, \mathsf{OTS.Ver})$ defined as follows.

$\mathsf{OTS.KGen}(1^\lambda) \to (\mathsf{sigk}, \mathsf{vk})$. On input the security parameter 1^λ, it outputs a signing/verification key pair $(\mathsf{sigk}, \mathsf{vk})$.

$\mathsf{OTS.Sign}(\mathsf{sigk}, \mu) \to \sigma$. On input sigk and a message μ, it outputs a signature σ.

$\mathsf{OTS.Ver}(\mathsf{vk}, \mu, \sigma) \to 0/1$. On input vk, μ, and σ, it outputs 0 which indicates "reject" or 1 which indicates "accept".

Definition 12 (Correctness)**.** $\Pi_{\mathsf{OTS}} = (\mathsf{OTS.KGen}, \mathsf{OTS.Sign}, \mathsf{OTS.Ver})$ *is said to satisfy the correctness if for every* $\mu \in \mathcal{M}$ *and* $(\mathsf{sigk}, \mathsf{vk}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$*, it holds that* $\mathsf{OTS.Ver}(\mathsf{vk}, \mu, \mathsf{OTS.Sign}(\mathsf{sigk}, \mu)) = 1$ *with overwhelming probability.*

Definition 13 (Strong Q-$\mathsf{EUF\text{-}CMA}$ Security)**.** *The strong* Q-$\mathsf{EUF\text{-}CMA}$ *security of* $\Pi_{\mathsf{OTS}} = (\mathsf{OTS.KGen}, \mathsf{OTS.Sign}, \mathsf{OTS.Ver})$ *is defined by the security game between a challenger* $\mathcal{C}$ *and an adversary* $\mathcal{A}$ *as follows.*

Init. *$\mathcal{C}$ runs* $(\mathsf{sigk}^{\langle q \rangle}, \mathsf{vk}^{\langle q \rangle}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$ *for* $q \in [Q]$ *and sends* $\{\mathsf{vk}^{\langle q \rangle}\}_{q \in [Q]}$ *to* $\mathcal{A}$*.*

Sign Query. *$\mathcal{A}$ is allowed to make the query only once for each* $q \in [Q]$*. Upon $\mathcal{A}$'s query on* $(q, \mu^{\langle q \rangle})$*,* $\mathcal{C}$ *runs* $\sigma^{\langle q \rangle} \leftarrow \mathsf{OTS.Sign}(\mathsf{sigk}^{\langle q \rangle}, \mu^{\langle q \rangle})$ *and sends* $\sigma^{\langle q \rangle}$ *to* $\mathcal{A}$*.*

Forge. *$\mathcal{A}$ outputs* $(\mu^\star, \sigma^\star)$ *which is not a pair of a queried message and a returned signature of sign queries and terminates the game.*

If the advantage of $\mathcal{A}$ *for breaking the* $\mathsf{EUF\text{-}CMA}$ *security of* Π_{OTS} *defined by* $\mathsf{Adv}^{Q\text{-}\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{OTS}}, \mathcal{A}}(\lambda) := \Pr\left[\sum_{q \in [Q]} \mathsf{OTS.Ver}(\mathsf{vk}^{\langle q \rangle}, \mu^\star, \sigma^\star) \geq 1\right]$ *is negligible in* λ*,* Π_{OTS} *is said to satisfy the* $\mathsf{EUF\text{-}CMA}$ *security.*

Remark 4. *If* $Q = 1$*, we simply call the strong* $\mathsf{EUF\text{-}CMA}$ *security. Moreover,* $\mathcal{A}$ *does not make a sign query on* $(1, \mu)$ *but on* μ*.*

2.6 Message Authentication Codes

A *message authentication code* (MAC) scheme consists of three polynomial-time algorithms $\Pi_{\mathsf{MAC}} = (\mathsf{MAC.KGen}, \mathsf{MAC.TAG}, \mathsf{MAC.Ver})$ defined as follows.

$\mathsf{MAC.KGen}(1^\lambda) \to \mathsf{mk}$. On input the security parameter 1^λ, it outputs a MAC secret key mk.

$\mathsf{MAC.TAG}(\mathsf{mk}, \mu) \to \tau$. On input mk and a message μ, it outputs a tag τ.

$\mathsf{MAC.Ver}(\mathsf{mk}, \mu, \tau) \to 0/1$. On input mk, μ, and τ, it outputs 0 which indicates "reject" or 1 which indicates "accept".

Definition 14 (Correctness)**.** $\Pi_{\mathsf{MAC}} = (\mathsf{MAC.KGen}, \mathsf{MAC.TAG}, \mathsf{MAC.Ver})$ *is said to satisfy the correctness if for every* $\mu \in \mathcal{M}$ *and* $\mathsf{mk} \leftarrow \mathsf{MAC.KGen}(1^\lambda)$*, it holds that* $\mathsf{MAC.Ver}(\mathsf{mk}, \mu, \mathsf{MAC.TAG}(\mathsf{mk}, \mu)) = 1$ *with overwhelming probability.*

Definition 15 (Strong $\mathsf{EUF\text{-}CMA}$ Security)**.** *The strong* $\mathsf{EUF\text{-}CMA}$ *security of* $\Pi_{\mathsf{MAC}} = (\mathsf{MAC.KGen}, \mathsf{MAC.TAG}, \mathsf{MAC.Ver})$ *is defined by the security game between a challenger* $\mathcal{C}$ *and an adversary* $\mathcal{A}$ *as follows.*

Init. $\mathcal{C}$ *runs* $\mathsf{mk} \leftarrow \mathsf{MAC.KGen}(1^\lambda)$.

Tag Query. *Upon* $\mathcal{A}$*'s query on* μ, $\mathcal{C}$ *runs* $\tau \leftarrow \mathsf{MAC.TAG}(\mathsf{mk}, \mu)$ *and sends* τ *to* $\mathcal{A}$.

Verify Query. *Upon* $\mathcal{A}$*'s query on* (μ, τ), $\mathcal{C}$ *sends a result of* $\mathsf{MAC.Ver}(\mathsf{mk}, \mu, \tau)$ *to* $\mathcal{A}$.

Forge. $\mathcal{A}$ *outputs* $(\mu^\star, \tau^\star)$ *which is not a pair of a queried message and a returned MAC tag of tag queries and terminates the game.*

If the advantage of $\mathcal{A}$ *for breaking the strong* $\mathsf{EUF\text{-}CMA}$ *security of* Π_{MAC} *defined by* $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MAC}},\mathcal{A}}(\lambda) \coloneqq \Pr[\mathsf{MAC.Ver}(\mathsf{mk}, \mu^\star, \tau^\star) = 1]$ *is negligible in* λ, Π_{MAC} *is said to satisfy the strong* $\mathsf{EUF\text{-}CMA}$ *security.*

2.7 Hash Function

Definition 16 (Collision Resistance)**.** *A family of hash functions* $\mathcal{H} = \{H_i : \{0,1\}^* \to \mathcal{R}\}_i$ *satisfies the collision resistance if any PPT adversary* $\mathcal{A}$ *which is given* $H \leftarrow_R \mathcal{H}$ *cannot find* x, x' *such that* $x \neq x' \wedge H(x) = H(x')$ *with non-negligible probability.*

3 Generic Construction of KFHE

In this section, we propose a generic construction of keyed KFHE. We describe the generic construction in Section 3.1 and prove its $\mathsf{KH\text{-}CCA}$ security in Section 3.2.

3.1 Construction

We follow the idea explained in Section 1.3.2 and propose a generic construction of KFHE from MFHE, IBE, OTS, and MAC.

$\mathsf{KFHE.KGen}(1^\lambda) \to (\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.hk})$. Run $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$, $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$, and $\mathsf{mk} \leftarrow \mathsf{MAC.KGen}(1^\lambda)$. Choose a one-time signature scheme Π_{OTS}. Output $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$, $\mathsf{KFHE.dk} = (\mathsf{IBE.msk}, \mathsf{mk})$, and $\mathsf{KFHE.hk} = \mathsf{mk}$.

$\mathsf{KFHE.Enc}(\mathsf{KFHE.pk}, \mu) \to \mathsf{KFHE.ct}$. Parse $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$. Run

- $(\mathsf{MFHE.pk}, \mathsf{MFHE.sk}) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$,
- $\mathsf{MFHE.ct} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}, \mu)$,
- $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,
- $\mathsf{IBE.ct_{vk}} \leftarrow \mathsf{IBE.Enc}(\mathsf{vk}, \mathsf{MFHE.sk})$,
- $\sigma \leftarrow \mathsf{Sign}\left(\mathsf{sigk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct_{vk}}, \mathsf{MFHE.ct})\right)$.

Output

$$\mathsf{KFHE.ct} = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct_{vk}}, \mathsf{MFHE.ct}, \sigma).$$

We say that a pre-evaluated ciphertext $\mathsf{KFHE.ct}$ *is valid if* σ *is a valid signature for* $(\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct_{vk}}, \mathsf{MFHE.ct})$.

$\mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C}) \to \mathsf{KFHE.ct}_{\mathsf{C}}/\perp$. Output $\perp$ if there are invalid ciphertexts $\mathsf{KFHE.ct}^{(\ell)}$ for some $\ell \in [L]$. Otherwise, parse $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$, $\mathsf{KFHE.hk} = \mathsf{mk}$, and $\mathsf{KFHE.ct}^{(\ell)} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)})$ for $\ell \in [L]$. Run

- $\mathsf{MFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C})$,
- $\tau \leftarrow \mathsf{MAC.TAG}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}))$.

Output

$$\mathsf{KFHE.ct}_{\mathsf{C}} = \left((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \tau\right).$$

We say that an evaluated ciphertext $\mathsf{KFHE.ct}_{\mathsf{C}}$ is valid if τ is a valid MAC tag for $((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$.

$\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.ct}/\mathsf{KFHE.ct}_{\mathsf{C}}) \to \mu/\perp$. Parse $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$ and $\mathsf{KFHE.dk} = (\mathsf{IBE.msk}, \mathsf{mk})$. Proceed as follows.

Case of Pre-evaluated Ciphertexts. Output $\perp$ if $\mathsf{KFHE.ct}$ is invalid. Otherwise, parse $\mathsf{KFHE.ct} = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma)$. Run

* $\mathsf{IBE.sk}_{\mathsf{vk}} \leftarrow \mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{vk})$,
* $\mathsf{MFHE.sk} \leftarrow \mathsf{IBE.Dec}(\mathsf{IBE.sk}_{\mathsf{vk}}, \mathsf{IBE.ct}_{\mathsf{vk}})$,

and output $\mu \leftarrow \mathsf{MFHE.Dec}(\mathsf{MFHE.sk}, \mathsf{MFHE.ct})$.

Case of Evaluated Ciphertexts. Output $\perp$ if $\mathsf{KFHE.ct}_{\mathsf{C}}$ is invalid. Otherwise, parse $\mathsf{KFHE.ct}_{\mathsf{C}} = \left((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \tau\right)$. For $\ell \in [L]$, run

* $\mathsf{IBE.sk}_{\mathsf{vk}^{(\ell)}} \leftarrow \mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{vk}^{(\ell)})$,
* $\mathsf{MFHE.sk}^{(\ell)} \leftarrow \mathsf{IBE.Dec}(\mathsf{IBE.sk}_{\mathsf{vk}^{(\ell)}}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})$,

and output $\mu \leftarrow \mathsf{MFHE.Dec}((\mathsf{MFHE.sk}^{(\ell)})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$.

Theorem 1. *If the underlying* MFHE *scheme* Π_{MFHE}*,* IBE *scheme* Π_{IBE}*, one-time signature scheme* Π_{OTS}*, and* MAC *scheme* Π_{MAC} *satisfies the correctness, the proposed* KFHE *scheme* Π_{KFHE} *satisfies the correctness.*

Proof of Theorem 1. For every $\mu \in \mathcal{M}$,

- $(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.hk}) \leftarrow \mathsf{KFHE.KGen}(1^\lambda)$;
  - $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$,
  - $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$,
  - $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$, $\mathsf{KFHE.dk} = (\mathsf{IBE.msk}, \mathsf{mk})$, and $\mathsf{KFHE.hk} = \mathsf{mk}$,
- $\mathsf{KFHE.ct} \leftarrow \mathsf{KFHE.Enc}(\mathsf{KFHE.pk}, \mu)$;
  - $(\mathsf{MFHE.pk}, \mathsf{MFHE.sk}) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$,
  - $\mathsf{MFHE.ct} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}, \mu)$,
  - $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,

- $\mathsf{IBE.ct}_{\mathsf{vk}} \leftarrow \mathsf{IBE.Enc}(\mathsf{vk}, \mathsf{MFHE.sk})$,
- $\sigma \leftarrow \mathsf{Sign}\,(\mathsf{sigk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}))$,

the correctness of Π_{OTS} ensures that $\mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1$ holds, the correctness of Π_{IBE} ensures that $\mathsf{IBE.Dec}(\mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{vk}), \mathsf{IBE.ct}_{\mathsf{vk}}) = \mathsf{MFHE.sk}$ holds, and the correctness of Π_{MFHE} ensures that $\mathsf{MFHE.Dec}(\mathsf{MFHE.sk}, \mathsf{MFHE.ct}) = \mu$ holds. Thus, $\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.ct}) = \mu$ holds.

For every circuit $\mathsf{C} : \mathcal{M}^L \to \mathcal{M}$, $(\mu^{(1)}, \ldots, \mu^{(L)}) \in \mathcal{M}^L$,

- $(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.hk}) \leftarrow \mathsf{KFHE.KGen}(1^\lambda)$;
  - $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$,
  - $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$,
  - $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$, $\mathsf{KFHE.dk} = (\mathsf{IBE.msk}, \mathsf{mk})$, and $\mathsf{KFHE.hk} = \mathsf{mk}$,
- $\mathsf{KFHE.ct}^{(\ell)} \leftarrow \mathsf{KFHE.Enc}(\mathsf{KFHE.pk}, \mu^{(\ell)})$ for $\ell \in [L]$;
  - $(\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.sk}^{(\ell)}) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$,
  - $\mathsf{MFHE.ct}^{(\ell)} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^{(\ell)}, \mu^{(\ell)})$,
  - $(\mathsf{vk}^{(\ell)}, \mathsf{sigk}^{(\ell)}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,
  - $\mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}} \leftarrow \mathsf{IBE.Enc}(\mathsf{vk}^{(\ell)}, \mathsf{MFHE.sk}^{(\ell)})$,
  - $\sigma^{(\ell)} \leftarrow \mathsf{Sign}\left(\mathsf{sigk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}}, \mathsf{MFHE.ct}^{(\ell)})\right)$,
  - $\mathsf{KFHE.ct}^{(\ell)} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)})$,
- $\mathsf{KFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C})$;
  - $\mathsf{MFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C})$,
  - $\tau \leftarrow \mathsf{MAC.TAG}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}))$,
  - $\mathsf{KFHE.ct}_{\mathsf{C}} = \left((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]} \mathsf{MFHE.ct}_{\mathsf{C}}, \tau\right)$,

the correctness of Π_{MAC} ensures that $\mathsf{MAC.Ver}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}),$ $\tau) = 1$ holds, the correctness of Π_{IBE} ensures that $\mathsf{IBE.Dec}(\mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{vk}^{(\ell)}), \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}) =$ $\mathsf{MFHE.sk}^{(\ell)}$ holds, and the correctness of Π_{MFHE} ensures that $\mathsf{MFHE.Dec}((\mathsf{MFHE.sk}^{(\ell)})_{\ell \in [L]},$ $\mathsf{MFHE.ct}_{\mathsf{C}}) = \mathsf{C}((\mu^{(\ell)})_{\ell \in [L]})$. Thus, $\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.ct}_{\mathsf{C}}) = \mathsf{C}((\mu^{(\ell)})_{\ell \in [L]})$ holds. □

Theorem 2. *The proposed* KFHE *scheme* Π_{KFHE} *satisfies compactness if the underlying* MFHE *scheme satisfies compactness.*

Proof of Theorem 2. For every λ,

- $(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.hk}) \leftarrow \mathsf{KFHE.KGen}(1^\lambda)$;
  - $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$,
  - $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$,

- $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$, $\mathsf{KFHE.dk} = (\mathsf{IBE.msk}, \mathsf{mk})$, and $\mathsf{KFHE.hk} = \mathsf{mk}$,

- $\mathsf{KFHE.ct}^{(\ell)} \leftarrow \mathsf{KFHE.Enc}(\mathsf{KFHE.pk}, \mu^{(\ell)})$ for $\ell \in [L]$;
  - $(\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.sk}^{(\ell)}) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$,
  - $\mathsf{MFHE.ct}^{(\ell)} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^{(\ell)}, \mu^{(\ell)})$,
  - $(\mathsf{vk}^{(\ell)}, \mathsf{sigk}^{(\ell)}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,
  - $\mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}} \leftarrow \mathsf{IBE.Enc}(\mathsf{vk}^{(\ell)}, \mathsf{MFHE.sk}^{(\ell)})$,
  - $\sigma^{(\ell)} \leftarrow \mathsf{Sign}\left(\mathsf{sigk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}}, \mathsf{MFHE.ct}^{(\ell)})\right)$,
  - $\mathsf{KFHE.ct}^{(\ell)} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)})$,

- $\mathsf{KFHE.ct_C} \leftarrow \mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$;
  - $\mathsf{MFHE.ct_C} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$,
  - $\tau \leftarrow \mathsf{MAC.TAG}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct_C}))$,
  - $\mathsf{KFHE.ct_C} = \left((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}\mathsf{MFHE.ct_C}, \tau\right)$,

the compactness of Π_{MFHE} ensures that $|\mathsf{MFHE.ct_C}|$ is independent of the size and depth of C and at most $L \cdot \mathsf{poly}(\lambda)$, and $|(\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}|$ and $|\tau|$ are independent of the size and depth of C and at most $L \cdot \mathsf{poly}(\lambda)$. Thus, $|\mathsf{KFHE.ct_C}|$ is independent of the size and depth of C and at most $L \cdot \mathsf{poly}(\lambda)$. □

3.2 Security

Theorem 3 (KH-CCA Security of Π_{KFHE})**.** *If the underlying* MFHE *scheme* Π_{MFHE} *satisfies the* IND-CPA *security,* IBE *scheme* Π_{IBE} *satisfies the selective* IND-CPA *security, one-time signature scheme* Π_{OTS} *and* MAC *scheme* Π_{MAC} *satisfy the strong* EUF-CMA *security, the proposed* KFHE *scheme* Π_{KFHE} *satisfies the* KH-CCA *security.*

Although we already explained the intuition of a proof in Section 1.3.2, we provide a more detailed overview. We prove Theorem 3 by using a sequence of games $\mathsf{Game}_0, \cdots, \mathsf{Game}_3$. Let $\mathsf{KFHE.ct}^\star = (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star, \sigma^\star)$ denote a challenge ciphertext. We can prove Theorem 3 when $\mathsf{MFHE.ct}^\star$ which is an encryption of $\mu^\star_{\mathsf{coin}}$ becomes indistinguishable from an encryption of a random string based on the IND-CPA security of Π_{MFHE} in Game_3. To prove the task, we change $\mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}$ which is an encryption of $\mathsf{MFHE.sk}^\star$ to be an encryption of a random string in Game_3, where the selective IND-CPA security of Π_{IBE} ensures $\mathsf{Game}_2 \approx_c \mathsf{Game}_3$. For this purpose, we have to ensure that the challenger $\mathcal{C}$ does not use an IBE secret key $\mathsf{IBE.sk}_{\mathsf{vk}^\star}$ to answer all the adversary $\mathcal{A}$'s decryption queries. In other words, what all we have to ensure is that $\mathcal{A}$ does not make decryption queries on pre-evaluated ciphertexts $\mathsf{KFHE.ct} = (\mathsf{vk}, \cdots)$ such that $\mathsf{vk} = \mathsf{vk}^\star$ and evaluated ciphertexts $\mathsf{KFHE.ct_C} = ((\mathsf{vk}^{(\ell)}, \cdots)_{\ell\in[L]}, \cdots)$ such that $\mathsf{vk}^\star \in (\mathsf{vk}^{(\ell)})_{\ell\in[L]}$. We can prove the claim for pre-evaluated ciphertexts (resp. evaluated ciphertexts) in Game_1 (resp. Game_2) by following the CHK transformation [CHK04] (resp. the encrypt-then-MAC paradigm [BN08]). In particular, the strong EUF-CMA security of Π_{OTS} (resp. Π_{MAC}) ensures $\mathsf{Game}_0 \approx_c \mathsf{Game}_1$ (resp. $\mathsf{Game}_1 \approx_c \mathsf{Game}_2$).

Proof of Theorem 3. We prove the theorem by using a sequence of games $\mathsf{Game}_0, \cdots, \mathsf{Game}_4$, where E_i denotes an event that $\mathcal{A}$ wins in Game_i.

Game_0. This is the $\mathsf{KH\text{-}CCA}$ security game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$. Hereafter, let

$$\mathsf{KFHE.ct}^\star = (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star, \sigma^\star).$$

denote a challenge ciphertext, where $\mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}$ and $\mathsf{MFHE.ct}^\star$ are encryptions of $\mathsf{MFHE.sk}^\star$ and $\mu^\star_{\mathsf{coin}}$, respectively. Due to the definition of the $\mathsf{KH\text{-}CCA}$ security game, $\mathcal{C}$ stores the challenge ciphertext $\mathsf{KFHE.ct}^\star$ and its evaluation results in the list $\mathcal{L}$.

Game_1. This is the same as Game_0 except that upon $\mathcal{A}$'s evaluation queries and decryption queries on pre-evaluated ciphertexts. Upon $\mathcal{A}$'s evaluation queries on $((\mathsf{KFHE.ct}^{(\ell)} = (\mathsf{vk}^{(\ell)}, \cdots, \sigma^{(\ell)}))_{\ell\in[L]}, \mathsf{C})$ such that $\mathsf{vk}^\star \in (\mathsf{vk}^{(\ell)})_{\ell\in[L]} \wedge \mathsf{KFHE.ct}^\star \notin (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}$, $\mathcal{C}$ always outputs $\perp$. Upon $\mathcal{A}$'s decryption queries on $\mathsf{KFHE.ct} = (\mathsf{vk}, \cdots, \sigma)$ such that $\mathsf{vk} = \mathsf{vk}^\star$, $\mathcal{C}$ always outputs $\perp$.

The output is not $\perp$ only if $\sigma^{(\ell)}$ and σ are valid signatures accepted by $\mathsf{vk}^\star$. The strong $\mathsf{EUF\text{-}CMA}$ security of Π_{OTS} ensures that $\mathcal{A}$ cannot forge a signature $\sigma^{(\ell)}$ or σ. Thus, $\mathsf{Game}_1 \approx_c \mathsf{Game}_2$ holds.

Lemma 1 ($\mathsf{Game}_0 \approx_c \mathsf{Game}_1$)**.** *If Π_{OTS} satisfies the strong* $\mathsf{EUF\text{-}CMA}$ *security,* Game_0 *and* Game_1 *are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 1. Let F_1 denote an event that $\mathcal{A}$ makes an evaluation query on $((\mathsf{KFHE.ct}^{(\ell)} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)}))_{\ell\in[L]}, \mathsf{C})$ such that

$$\begin{aligned}&\mathsf{vk}^\star \in (\mathsf{vk}^{(\ell)})_{\ell\in[L]} \wedge \mathsf{KFHE.ct}^\star \notin (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]} \wedge \\ &\sum_{\ell\in[L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) = L\end{aligned}$$

or a decryption query on a pre-evaluated ciphertext $\mathsf{KFHE.ct} = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma)$ such that

$$\mathsf{vk} = \mathsf{vk}^\star \wedge \mathsf{KFHE.ct} \neq \mathsf{KFHE.ct}^\star \wedge \mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1.$$

If $\sum_{\ell\in[L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) < L$ holds upon $\mathcal{A}$'s evaluation query, there is an invalid pre-evaluated ciphertext in $(\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}$ and the design of Π_{KFHE} ensures that an answer to the query is $\perp$. If $\mathsf{KFHE.ct} = \mathsf{KFHE.ct}^\star$ holds upon $\mathcal{A}$'s decryption query, the definition of the $\mathsf{KH\text{-}CCA}$ security ensures that an answer to the query is $\perp$. If $\mathsf{OTS.Ver}(\mathsf{vk}^\star, (\mathsf{vk}^\star, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 0$ holds upon $\mathcal{A}$'s decryption query, the pre-evaluated ciphertext $\mathsf{KFHE.ct}$ is invalid and the design of Π_{KFHE} ensures that an answer to the query is $\perp$. Thus, $\mathsf{Game}_0 = \mathsf{Game}_1$ holds if F_1 does not occur. Therefore, it holds that $\Pr[E_0] \leq \Pr[E_1] + \Pr[F_1]$.

We construct a reduction algorithm $\mathcal{B}_1$ which interacts with $\mathcal{A}$ against Π_{KFHE} and breaks the strong $\mathsf{EUF\text{-}CMA}$ security of Π_{OTS}. After $\mathcal{B}_1$ receives $\mathsf{vk}^\star$ from $\mathcal{C}$ in the strong $\mathsf{EUF\text{-}CMA}$ security game of Π_{OTS}, it runs $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$, $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$, and $\mathsf{mk} \leftarrow \mathsf{MAC.KGen}(1^\lambda)$, and sends $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. Since $\mathcal{B}_1$ knows $\mathsf{KFHE.dk} = (\mathsf{IBE.msk}, \mathsf{mk})$ and $\mathsf{KFHE.hk} = \mathsf{mk}$, it can properly answer all $\mathcal{A}$'s homomorphic evaluation key reveal query, evaluation queries, and decryption queries on evaluated ciphertexts until F_1 occurs.

Upon $\mathcal{A}$'s challenge query on $(\mu^\star_0, \mu^\star_1)$, $\mathcal{B}_1$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$, runs $(\mathsf{MFHE.pk}^\star, \mathsf{MFHE.sk}^\star) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$, $\mathsf{MFHE.ct}^\star \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^\star, \mu^\star_{\mathsf{coin}})$, and

$\mathsf{IBE.ct}^\star_{\mathsf{vk}^\star} \leftarrow \mathsf{IBE.Enc}(\mathsf{vk}^\star, \mathsf{MFHE.sk}^\star)$, makes a sign query on $(\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star)$ to $\mathcal{C}$ and receives $\sigma^\star$, and sends $\mathsf{KFHE.ct}^\star = (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star, \sigma^\star)$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s evaluation query on $((\mathsf{KFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C})$, $\mathcal{B}_1$ can check whether F_1 occurs. If $\sum_{\ell \in [L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) < L$ holds, $\mathcal{B}_1$ sends $\perp$ to $\mathcal{A}$ due to the design of Π_{KFHE}. If $(\mathsf{vk}^\star \notin (\mathsf{vk}^{(\ell)})_{\ell \in [L]} \vee \mathsf{KFHE.ct}^\star \in (\mathsf{KFHE.ct}^{(\ell)})_{\ell \in [L]}) \wedge \sum_{\ell \in [L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) = L$ holds, $\mathcal{B}_1$ sends the result of $\mathsf{KFHE.Eval}(\mathsf{KFHE.pk}, \mathsf{KFHE.hk}, (\mathsf{KFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C})$ to $\mathcal{A}$. Upon $\mathcal{A}$'s decryption query on a pre-evaluated ciphertext $\mathsf{KFHE.ct}$, $\mathcal{B}_1$ can check whether F_1 occurs. If $\mathsf{KFHE.ct} = \mathsf{KFHE.ct}^\star \vee \mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 0$ holds, $\mathcal{B}_1$ sends $\perp$ to $\mathcal{A}$ due to the definition of the $\mathsf{KH\text{-}CCA}$ security and the design of Π_{KFHE}. If $\mathsf{vk} \neq \mathsf{vk}^\star \wedge \mathsf{KFHE.ct} \neq \mathsf{KFHE.ct}^\star \wedge \mathsf{OTS.Ver}(\mathsf{vk}^\star, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1$ holds, $\mathcal{B}_1$ sends the result of $\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk}, \mathsf{KFHE.ct})$ to $\mathcal{A}$. Otherwise, if F_1 occurs, $\mathcal{B}_1$ knows $\mathsf{KFHE.ct} = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma)$ such that $\mathsf{vk} = \mathsf{vk}^\star \wedge \mathsf{KFHE.ct} \neq \mathsf{KFHE.ct}^\star \wedge \mathsf{OTS.Ver}(\mathsf{vk}^\star, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1$. Then, $\mathcal{B}_1$ sends $((\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma)$ to $\mathcal{C}$ as a pair of a message and a forged signature. Since the condition $\mathsf{KFHE.ct} \neq \mathsf{KFHE.ct}^\star$ ensures that $((\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma)$ is not a pair of a queried message and a returned signature, while the condition $\mathsf{OTS.Ver}(\mathsf{vk}^\star, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1$ ensures that σ is a valid signature of a message $(\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct})$, $\mathcal{B}_1$ breaks the strong $\mathsf{EUF\text{-}CMA}$ security of Π_{OTS} with probability 1 if F_1 occurs. Therefore, it holds that

$$\Pr[E_0] \leq \Pr[E_1] + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{OTS}}, \mathcal{B}_1}(\lambda).$$

$\square$

Game_2. This is the same as Game_1 except that upon $\mathcal{A}$'s decryption queries on evaluated ciphertexts $\mathsf{KFHE.ct}_{\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \cdots)_{\ell \in [L]}, \cdots, \tau)$ such that $\mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell \in [L]}$, $\mathcal{C}$ always outputs $\perp$.

The output is not $\perp$ only if τ is a valid forged tag. The strong $\mathsf{EUF\text{-}CMA}$ security of Π_{MAC} ensures that $\mathcal{A}$ cannot forge a tag τ. Thus, $\mathsf{Game}_2 \approx_c \mathsf{Game}_3$ holds.

Lemma 2 ($\mathsf{Game}_1 \approx_c \mathsf{Game}_2$)**.** *If Π_{MAC} satisfies the strong $\mathsf{EUF\text{-}CMA}$ security, Game_1 and Game_2 are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 2. Let F_2 denote an event that $\mathcal{A}$ makes a decryption query on an evaluated ciphertext $\mathsf{KFHE.ct}_{\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \tau)$ such that

$$\begin{aligned}&\mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell \in [L]} \wedge \mathsf{KFHE.ct}_{\mathsf{C}} \notin \mathcal{L} \wedge \\ &\mathsf{MAC.Ver}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau) = 1.\end{aligned}$$

If $\mathsf{KFHE.ct}_{\mathsf{C}} \in \mathcal{L}$ holds, the definition of the $\mathsf{KH\text{-}CCA}$ security ensures that an answer to the query is $\perp$. If $\mathsf{MAC.Ver}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau) = 0$ holds, the evaluated ciphertext is invalid and the design of Π_{KFHE} ensures that the answer to the query is $\perp$. Thus, $\mathsf{Game}_1 = \mathsf{Game}_2$ holds if F_2 does not occur. Therefore, it holds that $\Pr[E_1] \leq \Pr[E_2] + \Pr[F_2]$.

We construct a reduction algorithm $\mathcal{B}_2$ which interacts with $\mathcal{A}$ against Π_{KFHE} and breaks the strong $\mathsf{EUF\text{-}CMA}$ security of Π_{MAC} with $\mathcal{C}$. Since $\mathcal{A}$ can make decryption queries only until it makes a homomorphic evaluation key reveal query, $\mathcal{A}$ does not make a homomorphic evaluation key reveal query during the reduction. After $\mathcal{B}_2$ begins the strong $\mathsf{EUF\text{-}CMA}$ security game of Π_{MAC}, it runs $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$ and $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$, chooses a one-time signature scheme Π_{OTS}, and sends $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. $\mathcal{B}_2$ answers the challenge query in the same way as in Game_1.

Upon $\mathcal{A}$'s evaluation query on $((\mathsf{KFHE.ct}^{(\ell)} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)}))_{\ell\in[L]}, \mathsf{C})$, $\mathcal{B}_2$ sends $\perp$ to $\mathcal{A}$ if $\mathsf{vk}^\star \in (\mathsf{vk}^{(\ell)})_{\ell\in[L]} \wedge \mathsf{KFHE.ct}^\star \notin (\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}$ holds as we modified in Game_1. $\mathcal{B}_2$ also sends $\perp$ to $\mathcal{A}$ if $\sum_{\ell\in[L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) < L$ holds since there is an invalid pre-evaluated ciphertext in $(\mathsf{KFHE.ct}^{(\ell)})_{\ell\in[L]}$. Otherwise, $\mathcal{B}_2$ runs $\mathsf{MFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$, makes a tag query on $((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$ and receives τ, and sends $\mathsf{KFHE.ct}_{\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \tau)$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s decryption query on a pre-evaluated ciphertext $\mathsf{KFHE.ct} = (\mathsf{vk}, \cdots)$, $\mathcal{B}_2$ sends $\perp$ to $\mathcal{A}$ if $\mathsf{vk} = \mathsf{vk}^\star$ holds as we modified in Game_1. Otherwise, $\mathcal{B}_2$ sends the result of $\mathsf{KFHE.Dec}(\mathsf{KFHE.pk}, \mathsf{KFHE.dk} = (\mathsf{IBE.msk}, \perp), \mathsf{KFHE.ct})$ to $\mathcal{A}$, where the answer is properly distributed since mk is not required. Upon $\mathcal{A}$'s decryption query on an evaluated ciphertext $\mathsf{KFHE.ct}_{\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \tau)$, $\mathcal{B}_2$ can check whether F_2 occurs by making a verification query on $(((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau)$ to $\mathcal{C}$ and receiving the result of $\mathsf{MAC.Ver}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau)$. If $\mathsf{KFHE.ct}_{\mathsf{C}} \in \mathcal{L} \vee \mathsf{MAC.Ver}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau) = 0$ holds, $\mathcal{B}_2$ sends $\perp$ to $\mathcal{A}$ due to the definition of the $\mathsf{KH\text{-}CCA}$ security and the design of Π_{KFHE}. If $\mathsf{vk}^\star \notin \{\mathsf{vk}^{(\ell)}\}_{\ell\in[L]} \wedge \mathsf{KFHE.ct}_{\mathsf{C}} \notin \mathcal{L} \wedge \mathsf{MAC.Ver}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau) = 1$ holds, $\mathcal{B}_2$ sends the result of $\mathsf{MFHE.Dec}((\mathsf{IBE.Dec}(\mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{vk}^{(\ell)}), \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}))_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$ to $\mathcal{A}$. Otherwise, if F_2 occurs, $\mathcal{B}_2$ knows $\mathsf{KFHE.ct}_{\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \tau)$ such that $\mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell\in[L]} \wedge \mathsf{KFHE.ct}_{\mathsf{C}} \notin \mathcal{L} \wedge \mathsf{MAC.Ver}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau) = 1$. Then, $\mathcal{B}_2$ sends $(((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau)$ to $\mathcal{C}$ as a pair of a message and a forged tag. Since the condition $\mathsf{KFHE.ct}_{\mathsf{C}} \notin \mathcal{L}$ ensures that $(((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau)$ is not a pair of a queried message and a returned tag, while the condition $\mathsf{MAC.Ver}(\mathsf{mk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}), \tau) = 1$ ensures that τ is a valid tag of a message $((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$, $\mathcal{B}_2$ breaks the strong $\mathsf{EUF\text{-}CMA}$ security of Π_{MAC} with probability 1 if F_2 occurs. Therefore, it holds that

$$\Pr[E_1] \leq \Pr[E_2] + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MAC}},\mathcal{B}_2}(\lambda).$$

$\square$

Game_3. This is the same as Game_2 except that $\mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}$ is an encryption of a random string sampled independently from $\mathsf{MFHE.sk}^\star$.

The selective $\mathsf{IND\text{-}CPA}$ security of the IBE scheme Π_{IBE} ensures that $\mathsf{Game}_2 \approx_c \mathsf{Game}_3$ holds. In short, the reduction algorithm runs $(\mathsf{vk}^\star, \mathsf{sigk}^\star) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$ at the beginning of the security game, and declares $\mathsf{vk}^\star$ as the challenge identity of the IBE security game. In the challenge phase, the reduction algorithm runs $(\mathsf{MFHE.pk}^\star, \mathsf{MFHE.sk}^\star) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$, samples a random string $\mu^\star$ whose length is the same as $\mathsf{MFHE.sk}^\star$ but the distribution is independent of $\mathsf{MFHE.sk}^\star$. Then, the reduction algorithm declares $(\mathsf{MFHE.sk}^\star, \mu^\star)$ as the challenge messages in the IBE security game and receives $\mathsf{IBE.ct}^{\star}_{\mathsf{vk}^{\star}}$ from the IBE challenger. The reduction algorithm can create the other elements of the challenge ciphertext by itself. Due to the changes in Game_1 and Game_2, the reduction algorithm can answer all $\mathcal{A}$'s decryption queries by receiving IBE secret keys of vk such that $\mathsf{vk} \neq \mathsf{vk}^{\star}$. Thus, it holds that $\mathsf{Game}_3 \approx_c \mathsf{Game}_4$.

Lemma 3 ($\mathsf{Game}_2 \approx_c \mathsf{Game}_3$)**.** *If Π_{IBE} satisfies the selective* $\mathsf{IND\text{-}CPA}$ *security,* Game_2 *and* Game_3 *are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 3. We construct a reduction algorithm $\mathcal{B}_3$ which interacts with $\mathcal{A}$ against Π_{KFHE} and breaks the selective $\mathsf{IND\text{-}CPA}$ security of Π_{IBE}. At the beginning of the game, $\mathcal{B}_3$ runs $(\mathsf{vk}^{\star}, \mathsf{sigk}^{\star}) \leftarrow \mathsf{OTS.KGen}(1^{\lambda})$ and declares $\mathsf{vk}^{\star}$ to $\mathcal{C}$ as the challenge identity of the selective $\mathsf{IND\text{-}CPA}$ security game of Π_{IBE}. After $\mathcal{B}_3$ receives $\mathsf{IBE.mpk}$ from $\mathcal{C}$, it runs $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^{\lambda})$ and $\mathsf{mk} \leftarrow \mathsf{MAC.KGen}(1^{\lambda})$, chooses a one-time signature scheme Π_{OTS}, and sends $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. Since $\mathcal{B}_3$ knows $\mathsf{KFHE.hk} = \mathsf{mk}$, it can properly answer all $\mathcal{A}$'s homomorphic evaluation key reveal query and evaluation queries.

Upon $\mathcal{A}$'s decryption query on a pre-evaluated ciphertext $\mathsf{KFHE.ct} = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma)$, $\mathcal{B}_3$ sends $\perp$ to $\mathcal{A}$ if $\mathsf{vk} = \mathsf{vk}^{\star}$ holds due to the modification in Game_1. $\mathcal{B}_3$ also sends $\perp$ to $\mathcal{A}$ if $\mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 0$ holds due to the design of Π_{KFHE}. Otherwise, $\mathcal{B}_3$ makes an IBE secret key reveal query on vk to $\mathcal{C}$ and receives $\mathsf{IBE.sk}_{\mathsf{vk}}$, then sends the result of $\mathsf{MFHE.Dec}(\mathsf{IBE.Dec}(\mathsf{IBE.sk}_{\mathsf{vk}}, \mathsf{IBE.ct}_{\mathsf{vk}}), \mathsf{MFHE.ct})$ to $\mathcal{A}$. Upon $\mathcal{A}$'s decryption query on an evaluated ciphertext $\mathsf{KFHE.ct}_{\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \tau)$, $\mathcal{B}_3$ sends $\perp$ to $\mathcal{A}$ if $\mathsf{vk}^{\star} \in \{\mathsf{vk}^{(\ell)}\}_{\ell\in[L]}$ holds due to the modification in Game_2. $\mathcal{B}_3$ also sends $\perp$ to $\mathcal{A}$ if $\sum_{\ell\in[L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) < L$ holds due to the design of Π_{KFHE}. Otherwise, $\mathcal{B}_3$ makes secret key reveal queries on $\mathsf{vk}^{(\ell)}$ to $\mathcal{C}$ and receives $\mathsf{IBE.sk}_{\mathsf{vk}^{(\ell)}}$ for $\ell \in [L]$, then sends the result of $\mathsf{MFHE.Dec}((\mathsf{IBE.Dec}(\mathsf{IBE.sk}_{\mathsf{vk}^{(\ell)}}, \mathsf{IBE.ct}^{(\ell)}_{\mathsf{vk}^{(\ell)}}))_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s challenge query on $(\mu_0^{\star}, \mu_1^{\star})$, $\mathcal{B}_3$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$, runs $(\mathsf{MFHE.pk}^{\star}, \mathsf{MFHE.sk}^{\star}) \leftarrow \mathsf{MFHE.KGen}(1^{\lambda})$ and $\mathsf{MFHE.ct}^{\star} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^{\star}, \mu^{\star}_{\mathsf{coin}})$, makes an IBE challenge query on $(\mathsf{MFHE.sk}^{\star}, \mu^{\star})$ to $\mathcal{C}$, where $\mu^{\star}$ is a random string with the same length as $\mathsf{MFHE.sk}^{\star}$, receives $\mathsf{IBE.ct}^{\star}_{\mathsf{vk}^{\star}}$, further runs $\sigma^{\star} \leftarrow \mathsf{Sign}\left(\mathsf{sigk}^{\star}, (\mathsf{vk}^{\star}, \mathsf{MFHE.pk}^{\star}, \mathsf{IBE.ct}^{\star}_{\mathsf{vk}^{\star}}, \mathsf{MFHE.ct}^{\star})\right)$, and sends $\mathsf{KFHE.ct}^{\star} = (\mathsf{vk}^{\star}, \mathsf{MFHE.pk}^{\star}, \mathsf{IBE.ct}^{\star}_{\mathsf{vk}^{\star}}, \mathsf{MFHE.ct}^{\star}, \sigma^{\star})$ to $\mathcal{A}$. After $\mathcal{B}_3$ receives $\widehat{\mathsf{coin}}$ from $\mathcal{A}$, $\mathcal{B}_3$ sends 0 to $\mathcal{C}$ if $\widehat{\mathsf{coin}} = \mathsf{coin}$ and 1 to $\mathcal{C}$ otherwise.

Although $\mathcal{B}_3$ makes secret key reveal queries to $\mathcal{C}$ for answering $\mathcal{A}$'s decryption queries, the modifications in Game_1 and Game_2 ensure that $\mathcal{B}_3$ does not make a secret key reveal query on $\mathsf{vk}^{\star}$. If $\mathsf{IBE.ct}^{\star}_{\mathsf{vk}^{\star}}$ is an encryption of $\mathsf{MFHE.sk}^{\star}$ (resp. $\mu^{\star}$), $\mathsf{KFHE.ct}^{\star}$ follow the distribution in Game_2 (resp. Game_3). Therefore, it holds that

$$|\Pr[E_2] - \Pr[E_3]| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{IBE}},\mathcal{B}_3}(\lambda).$$

□

Lemma 4 ($\mathsf{KH\text{-}CCA}$ Security in Game_3)**.** *If Π_{MFHE} satisfies the* $\mathsf{IND\text{-}CPA}$ *security, Π_{KFHE} satisfies the* $\mathsf{KH\text{-}CCA}$ *security in* Game_3*.*

Proof of Lemma 4. We construct a reduction algorithm $\mathcal{B}_4$ which interacts with $\mathcal{A}$ against Π_{KFHE} and breaks the $\mathsf{IND\text{-}CPA}$ security of Π_{MFHE}. After $\mathcal{B}_4$ receives $(\mathsf{MFHE.pp}, \mathsf{MFHE.pk}^{\star})$ from $\mathcal{C}$, it runs $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^{\lambda})$ and $\mathsf{mk} \leftarrow \mathsf{MAC.KGen}(1^{\lambda})$, chooses a one-time signature scheme Π_{OTS}, and sends $\mathsf{KFHE.pk} = (\mathsf{MFHE.pp}, \mathsf{IBE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. Since $\mathcal{B}_4$ knows $\mathsf{KFHE.dk} = (\mathsf{IBE.msk}, \mathsf{mk})$ and $\mathsf{KFHE.hk} = \mathsf{mk}$, it can properly answer all $\mathcal{A}$'s homomorphic evaluation key reveal query, evaluation queries, and decryption queries.

Upon $\mathcal{A}$'s challenge query on $(\mu_0^\star, \mu_1^\star)$, $\mathcal{B}_3$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$ and $\mu^\star \leftarrow_R \mathcal{M}$, makes a challenge query on the same $(\mu_0^\star, \mu_1^\star)$ to $\mathcal{C}$ and receives $\mathsf{MFHE.ct}^\star$, runs $(\mathsf{vk}^\star, \mathsf{sigk}^\star) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$, $\mathsf{IBE.ct}^\star_{\mathsf{id}^\star} \leftarrow \mathsf{IBE.Enc}(\mathsf{vk}^\star, \mu^\star)$, and $\sigma^\star \leftarrow \mathsf{Sign}(\mathsf{sigk}^\star, (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star))$, then sends $\mathsf{KFHE.ct}^\star = (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{IBE.ct}^\star_{\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star, \sigma^\star)$ to $\mathcal{A}$. After $\mathcal{B}_4$ receives $\widehat{\mathsf{coin}}$ from $\mathcal{A}$, $\mathcal{B}_4$ sends the same $\widehat{\mathsf{coin}}$ to $\mathcal{C}$.

If $\mathsf{MFHE.ct}^\star$ is an encryption of $\mu_0^\star$ (resp. $\mu_1^\star$), $\mathsf{KFHE.ct}^\star$ is also an encryption of $\mu_0^\star$ (resp. $\mu_1^\star$). Therefore, it holds that

$$\left|\Pr[E_3] - \frac{1}{2}\right| \le \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{MFHE}},\mathcal{B}_4}(\lambda).$$

□

We complete the proof of Theorem 3 since it holds that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{KH\text{-}CCA}}_{\Pi_{\mathsf{KFHE}},\mathcal{A}}(\lambda) &= \left|\Pr[E_0] - \frac{1}{2}\right| \\
&\le \sum_{i\in[3]} \left|\Pr[E_{i-1}] - \Pr[E_i]\right| + \left|\Pr[E_3] - \frac{1}{2}\right| \\
&\le \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{OTS}},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{MAC}},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{IBE}},\mathcal{B}_3}(\lambda) + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{MFHE}},\mathcal{B}_4}(\lambda).
\end{aligned}$$

□

4 Attribute-based Keyed (Fully) Homomorphic Encryption

We define attribute-based keyed fully homomorphic encryption (ABKFHE). An attribute-based keyed fully homomorphic encryption (ABKFHE) scheme for a predicate $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ consists of five polynomial-time algorithms $\Pi_{\mathsf{ABKFHE}} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$. On input the security parameter 1^λ, it outputs a master public/secret key pair $(\mathsf{mpk}, \mathsf{msk})$, where mpk implicitly contains a message space $\mathcal{M}$.

$\mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y) \to (\mathsf{dk}_y, \mathsf{hk}_y)$. On input a mpk, msk, and a key attribute $y \in \mathcal{Y}$, it outputs a decryption key dk_y and a homomorphic evaluation key hk_y for y.

$\mathsf{Enc}(\mathsf{mpk}, x, \mu) \to \mathsf{ct}_x$. On input a mpk, a ciphertext attribute $x \in \mathcal{X}$, and a message $\mu \in \mathcal{M}$, it outputs a pre-evaluated ciphertext ct_x for x.

$\mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C}) \to \mathsf{ct}_{\mathbf{x},\mathsf{C}}/\bot$. On input a mpk, hk_y for y, a circuit $\mathsf{C} : \mathcal{M}^L \to \mathcal{M}$, and a tuple of L ciphertexts $(\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}$, it outputs an evaluated ciphertext $\mathsf{ct}_{\mathbf{x},\mathsf{C}}$ for $\mathbf{x} = (x^{(1)}, \ldots, x^{(L)})$ or a rejection symbol $\bot$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y, \mathsf{ct}_x/\mathsf{ct}_{\mathbf{x},\mathsf{C}}) \to \mu/\bot$. On input a mpk, dk_y and $\mathsf{ct}_x/\mathsf{ct}_{\mathbf{x},\mathsf{C}}$, it outputs a decryption result $\mu \in \mathcal{M}$ or a rejection symbol $\bot$.

It is required that an Π_{ABKFHE} satisfies both correctness and compactness.

Definition 17 (Correctness)**.** *For a vector of ciphertext attributes* $\mathbf{x} = (x^{(1)}, \ldots, x^{(L)}) \in \mathcal{X}^L$ *and a key attribute* $y \in \mathcal{Y}$*, we use the notation* $f(\mathbf{x}, y) = 1$ *if it holds that* $f(x^{(\ell)}, y) = 1$ *for all* $\ell \in [L]$*.* $\Pi_{\mathsf{ABKFHE}} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *satisfies correctness if the following conditions hold with overwhelming probability:*

- *For every* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*,* $(x, y) \in \mathcal{X} \times \mathcal{Y}$ *such that* $f(x, y) = 1$*,* $(\mathsf{dk}_y, \mathsf{hk}_y) \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$*, and* $\mu \in \mathcal{M}$*, it holds that* $\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y, \mathsf{Enc}(\mathsf{mpk}, x, \mu)) = \mu$*.*
- *For every* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*,* $(\mathbf{x} = (x^{(1)}, \ldots, x^{(L)}), y, y') \in \mathcal{X}^L \times \mathcal{Y}^2$ *such that* $f(\mathbf{x}, y) = f(\mathbf{x}, y') = 1$*,* $(\mathsf{dk}_y, \mathsf{hk}_y) \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$*,* $(\mathsf{dk}_{y'}, \mathsf{hk}_{y'}) \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y')$*, circuit* $\mathsf{C} : \mathcal{M}^L \to \mathcal{M}$*, and* $(\mu^{(1)}, \ldots, \mu^{(L)}) \in \mathcal{M}^L$*, it holds that* $\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y, \mathsf{ct}_{\mathbf{x},\mathsf{C}}) = \mathsf{C}(\mu^{(1)}, \ldots, \mu^{(L)})$ *with overwhelming probability, where* $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \leftarrow \mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_{y'}, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}, \mathsf{C})$ *and* $\mathsf{ct}^{(\ell)}_{x^{(\ell)}} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^{(\ell)}, \mu^{(\ell)})$ *for every* $\ell \in [L]$*.*

Definition 18 (Compactness)**.** $\Pi_{\mathsf{ABKFHE}} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *satisfies compactness if there exists a polynomial* poly *such that* $|\mathsf{ct}_{\mathbf{x},\mathsf{C}}|$*, where* $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \leftarrow \mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}, \mathsf{C})$*, is independent of the size and depth of* C *and at most* $L \cdot \mathsf{poly}(\lambda)$ *for every security parameter* λ*.*

Remark 5. *An attribute-based keyed homomorphic encryption (*ABKHE*) scheme* $\Pi_{\mathsf{ABKHE}} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *is defined in the same way except the* Eval *algorithm in two points. At first, since we will construct a fully compact* ABKHE *scheme* Π_{ABKHE} *in the sense that a pre-evaluated ciphertext* ct_x *and an evaluated ciphertext* $\mathsf{ct}_{\mathbf{x},\mathsf{C}}$ *follow the same distribution,* $\mathsf{ct}^{(1)}_{x^{(1)}}, \ldots, \mathsf{ct}^{(L)}_{x^{(L)}}$ *which are inputs of* Eval *satisfy* $x = x^{(1)} = \cdots = x^{(L)}$*. Next, since we will construct an* ABKHE *scheme* Π_{ABKHE} *with multiplicative homomorphism,* Eval *does not take a circuit* C *as input. The correctness ensures that a decryption result of* $\mathsf{ct}_x \leftarrow \mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_y, (\mathsf{ct}^{(\ell)}_x)_{\ell \in [L]})$ *is a product of decryption results of* $\mathsf{ct}^{(\ell)}_x$*.*

We define the KH-CCA security for ABKFHE by following Definition 3.

Definition 19 (KH-CCA security)**.** *The adaptive* KH-CCA *security of* $\Pi_{\mathsf{ABKFHE}} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *is defined by the security game between a challenger* $\mathcal{C}$ *and an adversary* $\mathcal{A}$ *as follows.*

Init. $\mathcal{C}$ *runs* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and sends* mpk *to* $\mathcal{A}$*.*

Phase 1. $\mathcal{A}$ *is allowed to make the following four types of queries to* $\mathcal{C}$*.*

Decryption Key Reveal Query. *Upon* $\mathcal{A}$*'s query on* $y \in \mathcal{Y}$*,* $\mathcal{C}$ *runs* $(\mathsf{dk}_y, \mathsf{hk}_y) \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$ *and sends* dk_y *to* $\mathcal{A}$*.*

Homomorphic Evaluation Key Reveal Query. *Upon* $\mathcal{A}$*'s query on* $y \in \mathcal{Y}$*,* $\mathcal{C}$ *runs* $(\mathsf{dk}_y, \mathsf{hk}_y) \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$ *and sends* hk_y *to* $\mathcal{A}$*.*

Evaluation Query. *Upon* $\mathcal{A}$*'s query on* $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}, \mathsf{C})$*,* $\mathcal{C}$ *runs* $(\mathsf{dk}_y, \mathsf{hk}_y) \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$ *and sends the result of* $\mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}, \mathsf{C})$ *to* $\mathcal{A}$*.*

Decryption Query. *Upon* $\mathcal{A}$*'s query on* $(y, \mathsf{ct}_x/\mathsf{ct}_{\mathbf{x},\mathsf{C}})$*,* $\mathcal{C}$ *runs* $(\mathsf{dk}_y, \mathsf{hk}_y) \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$ *and sends the result of* $\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y, \mathsf{ct}_x/\mathsf{ct}_{\mathbf{x},\mathsf{C}})$ *to* $\mathcal{A}$*.*

Challenge Query. $\mathcal{A}$ *is allowed to make the query only once. Upon* $\mathcal{A}$*'s query on* $(x^\star, \mu^\star_0, \mu^\star_1)$ *such that* $|\mu^\star_0| = |\mu^\star_1|$*,* $\mathcal{C}$ *outputs* $\perp$ *if* $\mathcal{A}$ *has already made a decryption key reveal query on* y *such that* $f(x^\star, y) = 1$*. Otherwise,* $\mathcal{C}$ *samples* $\mathsf{coin} \leftarrow_R \{0, 1\}$*, runs* $\mathsf{ct}^\star_{x^\star} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^\star, \mu^\star_{\mathsf{coin}})$*, creates a list of ciphertexts* $\mathcal{L} = \{\mathsf{ct}^\star_{x^\star}\}$*, and sends* $\mathsf{ct}^\star_{x^\star}$ *to* $\mathcal{A}$*.*

Phase 2. *$\mathcal{A}$ is allowed to make the same four types of queries to $\mathcal{C}$ as in Phase 1 with the following exceptions.*

Decryption Key Reveal Query. *Upon $\mathcal{A}$'s query on $y \in \mathcal{Y}$, $\mathcal{C}$ outputs $\perp$ if $f(x^\star, y) = 1$ holds.*

Evaluation Query. *If $\{\mathsf{ct}^{(\ell)}_{x^{(\ell)}}\}_{\ell\in[L]} \cap \mathcal{L} \neq \emptyset$ holds and the evaluation result is not $\perp$ but $\mathsf{ct}_{\mathbf{x},\mathsf{C}}$, $\mathcal{C}$ updates a list $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathsf{ct}_{\mathbf{x},\mathsf{C}}\}$.*

Decryption Query. *Upon $\mathcal{A}$'s query on (y, ct_x), $\mathcal{C}$ outputs $\perp$ if $\mathsf{ct}_x = \mathsf{ct}^\star_{x^\star}$ holds.*

Upon $\mathcal{A}$'s query on $(y, \mathsf{ct}_{\mathbf{x},\mathsf{C}})$, $\mathcal{C}$ outputs $\perp$ if $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \mathcal{L}$ holds. $\mathcal{C}$ also outputs $\perp$ if $f(x^\star, y) = 1$ holds and $\mathcal{A}$ has already made a homomorphic evaluation key reveal query on y' such that $f(x^\star, y') = 1$.

Guess. *$\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0, 1\}$ as a guess of coin and terminates the game.*

If the advantage of $\mathcal{A}$ for breaking the $\mathsf{KH\text{-}CCA}$ security of Π_{ABKFHE} defined by $\mathsf{Adv}^{\mathsf{KH\text{-}CCA}}_{\Pi_{\mathsf{ABKFHE}},\mathcal{A}}(\lambda) := \left|\Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2}\right|$ is negligible in λ, Π_{ABKFHE} is said to satisfy the adaptive $\mathsf{KH\text{-}CCA}$ security. The selective $\mathsf{KH\text{-}CCA}$ *security is the same except that $\mathcal{A}$ declares $x^\star$ at the beginning of the security game.*

Remark 6. *Since a pre-evaluated ciphertext and an evaluated ciphertext of ABKHE follow the same distribution as we claimed in Remark 5, we change the restriction of decryption queries in Phase 2 as we claimed in Remark 2:*

Decryption Query. *Upon $\mathcal{A}$'s query on (y, ct_x), $\mathcal{C}$ outputs $\perp$ if $\mathsf{ct}_x \in \mathcal{L}$ holds. $\mathcal{C}$ also outputs $\perp$ if $f(x^\star, y) = 1$ holds and $\mathcal{A}$ has already made a homomorphic evaluation key reveal query on y' such that $f(x^\star, y') = 1$. Otherwise, $\mathcal{C}$ proceeds the same way as in Phase 1.*

Remark 7. *As in Remark 3, we call $\mathcal{A}$'s evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]})$ a* dependent evaluation query *if the answer is stored in $\mathcal{L}$. In other words, $\mathcal{A}$'s dependent evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}$ satisfies $\{\mathsf{ct}^{(\ell)}_{x^{(\ell)}}\}_{\ell\in[L]} \cap \mathcal{L} \neq \emptyset$. Otherwise, we call $\mathcal{A}$'s evaluation query on $(\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}$ an* independent evaluation query*.*

5 Delegatable Attribute-based Encryption

In this section, we define delegatable attribute-based encryption (DABE) which is suitable for a building block of ABKFHE. In Section 5.1, we provide the definition of DABE. In Section 5.2, we review basic knowledge of lattice-based cryptography. In Section 5.3, we construct a DABE scheme by combining with Yamada's adaptively secure IBE scheme [Yam17] and Boneh et al.'s selectively secure ABE scheme [BGG+14]. In Section 5.4, we prove the security. Since the construction of the proposed DABE scheme is straightforward, experts of lattice-based cryptography can skip Sections 5.2–5.4.

5.1 Definition

In this paper, let $\Pi_{\mathsf{DABE}} = (\mathsf{DABE.Setup}, \mathsf{DABE.KGen}, \mathsf{DABE.Enc}, \mathsf{DABE.Dec})$ denote a DABE scheme for a predicate $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ with a two-level hierarchical structure, where ciphertext attributes live in $(\mathcal{X} \times \{0, 1\}) \times \mathcal{ID}$, while key attributes live in either $\mathcal{Y} \times \{0, 1\}$ or $(\mathcal{Y} \times \{0, 1\}) \times \mathcal{ID}$. A ciphertext $\mathsf{DABE.ct}_{(x,b),\mathsf{id}}$ for $((x, b), \mathsf{id})$ can be decrypted by a secret key $\mathsf{DABE.sk}_{(y,b'),\mathsf{id}'}$ for $((y, b'), \mathsf{id}')$ iff $f(x, y) = 1 \wedge b = b' \wedge \mathsf{id} = \mathsf{id}'$ holds, while $\mathsf{DABE.sk}_{(y,b'),\mathsf{id}'}$ can be computed from $\mathsf{DABE.sk}_{(y,b')}$ for (y, b').

$\mathsf{DABE.Setup}(1^\lambda) \to (\mathsf{DABE.mpk}, \mathsf{DABE.msk})$. On input the security parameter 1^λ, it outputs a master public/secret key pair $(\mathsf{DABE.mpk}, \mathsf{DABE.msk})$, where $\mathsf{DABE.mpk}$ implicitly contains a message space $\mathcal{M}$. Although we do not explicitly describe, the following algorithms take $\mathsf{DABE.mpk}$ as input.

$\mathsf{DABE.Enc}((x,b), \mathsf{id}, \mu) \to \mathsf{DABE.ct}_{x,b,\mathsf{id}}$. On input a ciphertext attribute $((x,b),\mathsf{id}) \in (\mathcal{X} \times \{0,1\}) \times \mathcal{ID}$ and a message $\mu \in \mathcal{M}$, it outputs a ciphertext $\mathsf{DABE.ct}_{(x,b),\mathsf{id}}$ for $((x,b),\mathsf{id})$.

$\mathsf{DABE.KGen}(\mathsf{DABE.sk}_Y, Y') \to \mathsf{DABE.sk}_{Y'}$. On input a secret key $\mathsf{DABE.sk}_Y$ for a key attribute Y and another key attribute Y', it outputs a secret key $\mathsf{DABE.sk}_{Y'}$ for Y', where $\mathsf{DABE.sk}_Y = \mathsf{DABE.msk}$ holds if $Y' \in \mathcal{Y} \times \{0,1\}$, and $\mathsf{DABE.sk}_Y = \mathsf{DABE.msk} \vee Y \in \mathcal{Y} \times \{0,1\}$ holds if $Y' \in (\mathcal{Y} \times \{0,1\}) \times \mathcal{ID}$.

$\mathsf{DABE.Dec}(\mathsf{DABE.sk}_{(y,b'),\mathsf{id}'}, \mathsf{DABE.ct}_{(x,b),\mathsf{id}}) \to \mu/\bot$. On input $\mathsf{DABE.sk}_{(y,b'),\mathsf{id}'}$ and $\mathsf{DABE.ct}_{(x,b),\mathsf{id}}$, it outputs a decryption result μ or a failure symbol $\bot$.

Definition 20 (Correctness). *$\Pi_{\mathsf{DABE}} = (\mathsf{DABE.Setup}, \mathsf{DABE.KGen}, \mathsf{DABE.Enc}, \mathsf{DABE.Dec})$ is said to satisfy the correctness if for every $\mu \in \mathcal{M}$, $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$, $(x,y) \in \mathcal{X} \times \mathcal{Y}$ such that $f(x,y) = 1$, $b \in \{0,1\}$, and $\mathsf{id} \in \mathcal{ID}$, it holds that $\mu \leftarrow \mathsf{DABE.Dec}(\mathsf{DABE.sk}_{(y,b),\mathsf{id}}, \mathsf{DABE.ct}_{(x,b),\mathsf{id}})$ with overwhelming probability, where $\mathsf{DABE.ct}_{(x,b),\mathsf{id}} \leftarrow \mathsf{DABE.Enc}((x,b),\mathsf{id},\mu)$, $\mathsf{DABE.sk}_{y,b} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y,b))$, and $\mathsf{DABE.sk}_{(y,b),\mathsf{id}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{y,b}, ((y,b),\mathsf{id}))$.*

We define two security notions called *selective* IND-CPA *security* and *second-level adaptive* OW-CPA *security* depending the value of $b \in \{0,1\}$. Let $((x^\star, b^\star), \mathsf{id}^\star)$ denote a challenge ciphertext attribute. A DABE scheme satisfies the selective IND-CPA security if $b^\star = 0$ and the second-level adaptive OW-CPA security if $b^\star = 1$. The selective IND-CPA security follows the traditional definition of IND-CPA security, where the adversary declares the target ciphertext attribute $((x^\star,0),\mathsf{id}^\star)$ at the beginning of the security game. The second-level adaptive OW-CPA security follows the traditional definition of the OW-CPA security, where the adversary declares the first level of the target ciphertext attribute $(x^\star, 1)$ at the beginning of the security game and declares the second level $\mathsf{id}^\star$ in the challenge phase.

Definition 21 (Selective IND-CPA Security). *The selective* IND-CPA *security of $\Pi_{\mathsf{DABE}} = (\mathsf{DABE.Setup}, \mathsf{DABE.KGen}, \mathsf{DABE.Enc}, \mathsf{DABE.Dec})$ is defined by the security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows.*

Init. *$\mathcal{A}$ declares a challenge ciphertext attribute $((x^\star,0),\mathsf{id}^\star)$ to $\mathcal{C}$. Then, $\mathcal{C}$ runs $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$ and sends $\mathsf{DABE.mpk}$ to $\mathcal{A}$.*

Phase 1. *$\mathcal{A}$ is allowed to make the following secret key reveal queries to $\mathcal{C}$.*

- **Secret Key Reveal Query.** *Upon $\mathcal{A}$'s query on $(y,b) \in \mathcal{Y} \times \{0,1\}$, $\mathcal{C}$ outputs $\bot$ if $f(x^\star,y) = 1 \wedge b = 0$ holds. Otherwise, $\mathcal{C}$ runs $\mathsf{DABE.sk}_{(y,b)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y,b))$ and sends $\mathsf{DABE.sk}_{(y,b)}$ to $\mathcal{A}$. Upon $\mathcal{A}$'s query on $((y,b),\mathsf{id}) \in (\mathcal{Y} \times \{0,1\}) \times \mathcal{ID}$, $\mathcal{C}$ outputs $\bot$ if $f(x^\star,y) = 1 \wedge b = 0 \wedge \mathsf{id}^\star = \mathsf{id}$ holds. Otherwise, $\mathcal{C}$ runs $\mathsf{DABE.sk}_{(y,b)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y,b))$ and $\mathsf{DABE.sk}_{(y,b),\mathsf{id}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y,b)}, ((y,b),\mathsf{id}))$, and sends $\mathsf{DABE.sk}_{(y,b),\mathsf{id}}$ to $\mathcal{A}$.*

Challenge Query. *$\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $(\mu_0^\star, \mu_1^\star)$ such that $|\mu_0^\star| = |\mu_1^\star|$, $\mathcal{C}$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$, runs $\mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{id}^\star} \leftarrow \mathsf{DABE.Enc}(((x^\star,0),\mathsf{id}^\star), \mu^\star_{\mathsf{coin}})$, and sends the challenge cipehrtext $\mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{id}^\star}$ to $\mathcal{A}$.*

Phase 2. *$\mathcal{A}$ is allowed to make secret key reveal queries as in Phase 1.*

Guess. *$\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0,1\}$ as a guess of coin and terminates the game.*

If the advantage of $\mathcal{A}$ for breaking the selective IND-CPA *security of Π_{DABE} defined by $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{DABE}},\mathcal{A}}(\lambda) := \left|\Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 0\right] - \Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 1\right]\right|$ is negligible in λ, Π_{DABE} is said to satisfy the selective* IND-CPA *security.*

Definition 22 (Second-level Adaptive OW-CPA Security)**.** *The second-level adaptive* OW-CPA *security of $\Pi_{\mathsf{DABE}} = (\mathsf{DABE.Setup}, \mathsf{DABE.KGen}, \mathsf{DABE.Enc}, \mathsf{DABE.Dec})$ is defined by the security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows.*

Init. *$\mathcal{A}$ declares the first level of a challenge ciphertext attribute $(x^\star, 1)$ to $\mathcal{C}$. Then, $\mathcal{C}$ runs $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$ and sends $\mathsf{DABE.mpk}$ to $\mathcal{A}$.*

Phase 1. *$\mathcal{A}$ is allowed to make the following secret key reveal queries to $\mathcal{C}$.*

- **Secret Key Reveal Query.** *Upon $\mathcal{A}$'s query on $(y, b) \in \mathcal{Y} \times \{0,1\}$, $\mathcal{C}$ outputs $\perp$ if $f(x^\star, y) = 1 \wedge b = 1$ holds. Otherwise, $\mathcal{C}$ runs $\mathsf{DABE.sk}_{(y,b)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y, b))$ and sends $\mathsf{DABE.sk}_{(y,b)}$ to $\mathcal{A}$. Upon $\mathcal{A}$'s query on $((y, b), \mathsf{id}) \in (\mathcal{Y} \times \{0,1\}) \times \mathcal{ID}$, $\mathcal{C}$ runs $\mathsf{DABE.sk}_{(y,b)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y, b))$ and $\mathsf{DABE.sk}_{(y,b),\mathsf{id}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y,b)}, ((y, b), \mathsf{id}))$, and sends $\mathsf{DABE.sk}_{(y,b),\mathsf{id}}$ to $\mathcal{A}$.*

Challenge Query. *$\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $\mathsf{id}^\star$ to declare the second level of a challenge ciphertext attribute, $\mathcal{C}$ outputs $\perp$ if $\mathcal{A}$ made secret key reveal queries on $((y, 1), \mathsf{id})$ in Phase 1 such that $f(x^\star, y) = 1 \wedge \mathsf{id}^\star = \mathsf{id}$. Otherwise, $\mathcal{C}$ samples $\mu^\star \leftarrow_R \mathcal{M}$, runs $\mathsf{DABE.ct}^\star_{(x^\star,1),\mathsf{id}^\star} \leftarrow \mathsf{DABE.Enc}(((x^\star, 1), \mathsf{id}^\star), \mu^\star)$, and sends the challenge cipehrtext $\mathsf{DABE.ct}^\star_{(x^\star,1),\mathsf{id}^\star}$ to $\mathcal{A}$.*

Phase 2. *$\mathcal{A}$ is allowed to make secret key reveal queries as in Phase 1 except that $\mathcal{C}$ outputs $\perp$ upon $\mathcal{A}$'s queries on $((y, 1), \mathsf{id})$ such that $f(x^\star, y) = 1 \wedge \mathsf{id}^\star = \mathsf{id}$.*

Guess. *$\mathcal{A}$ outputs $\widehat{\mu}$ as a guess of $\mu^\star$ and terminates the game.*

If the advantage of $\mathcal{A}$ for breaking the second-level adaptive OW-CPA *security of Π_{DABE} defined by $\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\Pi_{\mathsf{DABE}},\mathcal{A}}(\lambda) := \left|\Pr\left[\widehat{\mu} = \mu^\star\right] - \frac{1}{|\mathcal{M}|}\right|$ is negligible in λ, Π_{DABE} is said to satisfy the second-level adaptive* OW-CPA *security.*

5.2 Preliminaries on Lattices-based Cryptography

5.2.1 Discrete Gaussian Distribution

For a positve integer m, let $D_{\mathbb{Z}^m,\sigma}$ denote a discrete Gaussian distribution over $\mathbb{Z}^m$ with a parameter $\sigma > 0$. We will use the following facts.

Lemma 5 (Lemma 2.5 of [Reg05])**.** *It holds that $\Pr[\|\mathbf{z}\| > \sigma\sqrt{m} : \mathbf{z} \leftarrow D_{\mathbb{Z}^m,\sigma}] \le 2^{-2m}$.*

Lemma 6 (Lemma 1 of [KY16])**.** *Let q, m, m' be positive integers and r be a positive real such that $r > \max\{\omega(\sqrt{\log m}), \omega(\sqrt{\log m'})\}$. For $\mathbf{b} \in \mathbb{Z}_q^m$, $\mathbf{z} \leftarrow D_{\mathbb{Z}^m,r}$, $\mathbf{V} \in \mathbb{Z}^{m\times m'}$, and positive real $s > \|\mathbf{V}\|_2$, there exists a PPT algorithm* ReRand *such that $\mathbf{V}^\top\mathbf{b} + \mathbf{y} \leftarrow \mathsf{ReRand}(\mathbf{B}, \mathbf{b} + \mathbf{z}, r, s)$, where $\mathbf{y}$ is distributed statistically close to $D_{\mathbb{Z}^{m'},2rs}$.*

5.2.2 Learning with Errors

We use the learning with errors (LWE) assumption to prove the security.

Definition 23 (LWE Assumption [Reg05]). *For positive integers $n = n(\lambda)$ and $m = m(n)$, a prime integer $q = q(n) > 2$, a real number $\alpha \in (0, 1)$, an advantage for solving the LWE problem $\mathsf{LWE}_{n,m,q,\alpha}$ by an algorithm $\mathcal{A}$ is defined to be*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{LWE}_{n,m,q,\alpha}}(\lambda) \coloneqq \left| \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + \mathbf{z}) \to 1\right] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{w} + \mathbf{z}) \to 1] \right|,$$

where $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$, $\mathbf{z} \leftarrow D_{\mathbb{Z}^m,\alpha q}$, and $\mathbf{w} \leftarrow_R \mathbb{Z}_q^m$. We say that the $\mathsf{LWE}_{n,m,q,\alpha}$ assumption holds if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{LWE}_{n,m,q,\alpha}}(\lambda)$ is negligible for all PPT $\mathcal{A}$.

5.2.3 Gadget Matrix

For $m > n\lceil \log q\rceil$, a full-rank matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is called a gadget matrix, where there exists a deterministic polynomial time algorithm $\mathbf{G}^{-1}$ which takes $\mathbf{U} \in \mathbb{Z}_q^{n\times m}$ as input and outputs $\mathbf{V} = \mathbf{G}^{-1}(\mathbf{U})$ such that $\mathbf{V} \in \{0, 1\}^{m\times m}$ and it holds that $\mathbf{GV} = \mathbf{U}$.

5.2.4 Trapdoor and Sampling Algorithms

Let $n, m,$ and q be positive integers and $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. For a matrix $\mathbf{V} \in \mathbb{Z}_q^{n\times m'}$, let $\mathbf{A}_\sigma^{-1}(\mathbf{V})$ denote a probability distribution according to the discrete Gaussian $(D_{\mathbb{Z}^m,\sigma})^{m'}$ conditioned on $\mathbf{A} \cdot \mathbf{A}_\sigma^{-1}(\mathbf{V}) = \mathbf{V}$. We use $\mathbf{A}_\sigma^{-1}$ to denote a σ-trapdoor for $\mathbf{A}$, where we can use it to sample $\mathbf{A}_\sigma^{-1}(\mathbf{V})$ for any $\mathbf{V} \in \mathbb{Z}_q^{n\times m'}$ in polynomial time. If there is a subscript such $\mathbf{A}_0$, we use notations $\mathbf{A}_{0,\sigma}^{-1}(\mathbf{V})$ and $\mathbf{A}_{0,\sigma}^{-1}$.

Lemma 7 ([ABB10a, ABB10b, BLP+13, CHKP12, GPV08, MP12]). *The following facts are known for trapdoors and sampling algorithms.*

1. *Given $\mathbf{A}_\sigma^{-1}$, one can obtain $\mathbf{A}_{\sigma'}^{-1}$ for any $\sigma' \geq \sigma$.*
2. *Given $\mathbf{A}_\sigma^{-1}$, one can obtain $[\mathbf{A} \parallel \mathbf{B}]_\sigma^{-1}$ and $[\mathbf{B} \parallel \mathbf{A}]_\sigma^{-1}$ for any $\mathbf{B}$.*
3. *Given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{R} \in \mathbb{Z}^{m\times m}$ with $m \geq n\lceil \log q\rceil$, and a full-rank $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, one can obtain $[\mathbf{A} \parallel \mathbf{AR} + \mathbf{HG}]_\sigma^{-1}$ for $\sigma = m \cdot \|\mathbf{R}\|_\infty \cdot \omega(\sqrt{\log m})$.*
4. *Given $\mathbf{A}_\sigma^{-1}$ for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, one can randomize it and obtain $\mathbf{A}_{\sigma'}^{-1}$ for any $\sigma' = \sigma \cdot \omega(\sqrt{m \log m})$.*
5. *There exists an efficient algorithm $\mathsf{TrapGen}(n, m, q)$ that outputs $(\mathbf{A}, \mathbf{A}_\sigma^{-1})$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for some $m = O(n \log q)$ and is statistically close to uniform, and $\sigma = \omega(\sqrt{n \log q \log m})$.*

5.2.5 Full-rank Difference Map

Agrawal et al. [ABB10a] introduced a notion of full-rank difference map to construct selectively secure IBE scheme under the LWE assumption. For a positive integer n and a prime integer q, there is an efficiently computable map $\mathsf{FRD} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ called the full-rank difference map, where $\mathsf{FRD}(\mathbf{u}) - \mathsf{FRD}(\mathbf{v})$ is full-rank for all distinct $\mathbf{u}$ and $\mathbf{v}$.

5.2.6 Randomness Extraction

We use the following variant of the leftover hash lemma.

Lemma 8 ([ABB10a]). *Let n, m, m', and q be a positive integer such that $m > (n+1)\log_2 q + \omega(\log n)$, $m' = m'(n)$ is polynomial in n, and $q > 2$ is a prime number. For all vector $\mathbf{e} \in \mathbb{Z}_q^m$, it holds that $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top \mathbf{e}) \approx (\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top \mathbf{e})$, where $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{n\times m}$, $\mathbf{B} \leftarrow_R \mathbb{Z}_q^{n\times m'}$, $\mathbf{R} \leftarrow_R \{-1,1\}^{m\times m'}$ mod q.*

5.2.7 Key Homomorphic Computation

$\mathsf{PubEval}(y, (\mathbf{B}_1, \ldots, \mathbf{B}_\ell)) \to \mathbf{B}_y$: On input a function $y \in \mathcal{Y}$ and matrices $\mathbf{B}_1, \ldots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n\times m}$, output a matrix $\mathbf{B}_y$.

$\mathsf{CTEval}(y, (x_i, \mathbf{B}_i, \mathbf{c}_i)_{i\in[\ell]}) \to \mathbf{c}_y$: On input a function $y \in \mathcal{Y}$, $x_1, \ldots, x_\ell \in \mathbb{Z}_q$, matrices $\mathbf{B}_1, \ldots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n\times m}$, and vectors $\mathbf{c}_i = [\mathbf{B}_i + x_i\mathbf{G}]^\top \mathbf{s} + \mathbf{z}_i \in \mathbb{Z}_q^m$ for some $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{z}_i \in \mathbb{Z}^m$ such that $\|\mathbf{z}\| \leq \delta$, output $\mathbf{c}_y \in \mathbb{Z}_q^m$.

$\mathsf{TrapEval}(y, (x_i^\star, \mathbf{R}_i)_{i\in[\ell]}, \mathbf{A}) \to \mathbf{R}_y$: On input a function $y \in \mathcal{Y}$, $x_1^\star, \ldots, x_\ell^\star \in \mathbb{Z}_q$, random matrices $\mathbf{R}_1, \ldots, \mathbf{R}_\ell \in \{-1,1\}^{m\times m}$, and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, output $\mathbf{R}_y$.

Lemma 9. *If the following conditions hold for a family of function $\mathcal{Y} = \{y : \mathbb{Z}_q^\ell \to \mathbb{Z}_q\}$ and $\alpha_\mathcal{Y} : \mathbb{Z} \to \mathbb{Z}$, we say that evaluation algorithms* ($\mathsf{PubEval}, \mathsf{CTEval}, \mathsf{TrapEval}$) *are $\alpha_\mathcal{Y}$-enabling for a function class $\mathcal{Y}$:*

- *For $\mathbf{B}_y \leftarrow \mathsf{PubEval}(y, (\mathbf{B}_1, \ldots, \mathbf{B}_\ell))$ and $\mathbf{c}_y \leftarrow \mathsf{CTEval}(y, (x_i, \mathbf{B}_i, \mathbf{c}_i)_{i\in[\ell]})$, there exists a vector $\mathbf{z}_y$ such that $\mathbf{c}_y = [\mathbf{B}_y + y(x_1, \ldots, x_\ell)\mathbf{G}]^\top \mathbf{s} + \mathbf{z}_y$ and $\|\mathbf{z}_y\| \leq \delta \cdot \alpha_\mathcal{Y}(n)$.*
- *For $\mathbf{B}_y \leftarrow \mathsf{PubEval}(y, (\mathbf{AR}_1 - x_1^\star\mathbf{G}, \ldots, \mathbf{AR}_\ell - x_\ell^\star\mathbf{G}))$ and $\mathbf{R}_y \leftarrow \mathsf{TrapEval}(y, (x_i^\star, \mathbf{R}_i)_{i\in[\ell]}, \mathbf{A})$, it holds that $\mathbf{AR}_y - y(x_1^\star, \ldots, x_\ell^\star)\mathbf{G} = \mathbf{B}_y$.*
- *If we set $\mathbf{R}_1, \ldots, \mathbf{R}_\ell \leftarrow_R \{-1,1\}^{m\times m}$, it holds that $\|\mathbf{R}_y\|_2 \leq \alpha_\mathcal{Y}(n)$ with overwhelming probability, where $\mathbf{R}_y \leftarrow \mathsf{TrapEval}(y, (x_i^\star, \mathbf{R}_i)_{i\in[\ell]}, \mathbf{A})$.*

5.2.8 Yamada's IBE Scheme

We review a multi-bit encryption variant of Yamada's IBE scheme denoted by Π_{Yam}.

$\mathsf{IBE.Setup}(1^\lambda) \to (\mathsf{IBE.mpk}, \mathsf{IBE.msk})$. Run $(\mathbf{A}, \mathbf{A}_\sigma^{-1}) \leftarrow \mathsf{TrapGen}(n, m, q)$, sample random matrices $\mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K \leftarrow_R \mathbb{Z}_q^{n\times m}$ and $\mathbf{U} \leftarrow_R \mathbb{Z}_q^{n\times \log|\mathcal{M}|}$, and outputs $\mathsf{IBE.mpk} = (\mathbf{A}, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K, \mathbf{U})$ and $\mathsf{IBE.msk} = \mathbf{A}_{\sigma_0}^{-1}$, where $\mathcal{M} = \{0,1\}^{\log|\mathcal{M}|}$ is a message space.

$\mathsf{IBE.Enc}(\mathsf{id}, \vec{\mu}) \to \mathsf{IBE.ct}_\mathsf{id}$. Parse $\mathsf{IBE.mpk} = (\mathbf{A}, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K, \mathbf{U})$ and $\vec{\mu} = (\mu_1, \ldots, \mu_{\log|\mathcal{M}|})$. Compute $\mathbf{D}_\mathsf{id} \leftarrow \mathsf{PubEval}(\mathsf{id}, (\mathbf{D}_1, \ldots, \mathbf{D}_K))$, sample $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$, $\mathbf{z}_0, \mathbf{z}_1 \leftarrow D_{\mathbb{Z}^{2m}, \alpha' q}$, and $\mathbf{z}_2 \leftarrow D_{\mathbb{Z}^{\log|\mathcal{M}|}, \alpha q}$, and output $\mathsf{IBE.ct}_\mathsf{id} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$, where

$$\mathbf{c}_0 = \mathbf{A}^\top \mathbf{s} + \mathbf{z}_0, \qquad \mathbf{c}_1 = [\mathbf{D}_0 + \mathbf{D}_\mathsf{id}]^\top \mathbf{s} + \mathbf{z}_1, \qquad \mathbf{c}_2 = \mathbf{U}^\top \mathbf{s} + \mathbf{z}_2 + \vec{\mu}\left\lfloor \frac{q}{2} \right\rfloor.$$

$\mathsf{IBE.KGen}(\mathsf{IBE.msk}, \mathsf{id}) \to \mathsf{IBE.sk}_\mathsf{id}$. Parse $\mathsf{IBE.mpk} = (\mathbf{A}, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K, \mathbf{U})$ and $\mathsf{IBE.msk} = \mathbf{A}_{\sigma_0}^{-1}$. Compute $\mathbf{D}_\mathsf{id} \leftarrow \mathsf{PubEval}(\mathsf{id}, (\mathbf{D}_1, \ldots, \mathbf{D}_k))$, $[\mathbf{A} \mid \mathbf{D}_0 + \mathbf{D}_\mathsf{id}]_{\sigma_0}^{-1}$ from $\mathbf{A}_{\sigma_0}^{-1}$, randomize it and output $\mathsf{IBE.sk}_\mathsf{id} = [\mathbf{A} \mid \mathbf{D}_0 + \mathbf{D}_\mathsf{id}]_{\sigma_1}^{-1}$.

$\mathsf{IBE.Dec}(\mathsf{IBE.sk}_{\mathsf{id}}, \mathsf{IBE.ct}_{\mathsf{id}}) \to \vec{\mu}/\perp$. Parse $\mathsf{IBE.sk}_{\mathsf{id}} = [\mathbf{A} \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]_{\sigma_1}^{-1}$ and $\mathsf{IBE.ct}_{\mathsf{id}} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$. Compute $\mathbf{D}_{\mathsf{id}} \leftarrow \mathsf{PubEval}(\mathsf{id}, (\mathbf{D}_1, \ldots, \mathbf{D}_K))$, $[\mathbf{A} \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]_{\sigma_1}^{-1}(\mathbf{U})$ from $[\mathbf{A} \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]_{\sigma_1}^{-1}$, and $\mathbf{c}' = \mathbf{c}_2 - ([\mathbf{A} \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]_{\sigma_1}^{-1}(\mathbf{U}))^\top \cdot [\mathbf{c}_0 \parallel \mathbf{c}_1]$. Parse $\mathbf{c}' = [c_1', \ldots, c_{\log|\mathcal{M}|}']$. For $i \in [\log|\mathcal{M}|]$, set $\mu_i = 0$ if $|c_i'| < q/4$ and $\mu_i = 1$ otherwise.

Theorem 4. *Yamada's* IBE *scheme* Π_{Yam} *satisfies the correctness and the adaptive* IND-CPA *security under the* $\mathsf{LWE}_{n,m,q,\alpha}$ *assumption.*

5.2.9 Boneh et al.'s ABE Scheme

We review a multi-bit encryption variant of Boneh et al.'s ABE scheme denoted by $\Pi_{\mathsf{BGG+}}$.

$\mathsf{ABE.Setup}(1^\lambda) \to (\mathsf{ABE.mpk}, \mathsf{ABE.msk})$. Run $(\mathbf{A}, \mathbf{A}_{\sigma_0}^{-1}) \leftarrow \mathsf{TrapGen}(n, m, q)$, sample random matrices $\mathbf{B}_1, \ldots, \mathbf{B}_J \leftarrow_R \mathbb{Z}_q^{n\times m}$ and $\mathbf{U} \leftarrow_R \mathbb{Z}_q^{n \times \log|\mathcal{M}|}$, and outputs $\mathsf{ABE.mpk} = (\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_J, \mathbf{U})$ and $\mathsf{ABE.msk} = \mathbf{A}_{\sigma_0}^{-1}$, where $\mathcal{M} = \{0,1\}^{\log|\mathcal{M}|}$ is a message space.

$\mathsf{ABE.Enc}(\vec{x}, \vec{\mu}) \to \mathsf{ABE.ct}_x$. Parse $\mathsf{ABE.mpk} = (\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_J, \mathbf{U})$, $\vec{x} = (x_1, \ldots, x_J)$, and $\vec{\mu} = (\mu_1, \ldots, \mu_{\log|\mathcal{M}|})$. Sample $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$, $\mathbf{R}_1, \ldots, \mathbf{R}_J \leftarrow_R \{-1,1\}^{m\times m}$, $\mathbf{z}_0 \leftarrow D_{\mathbb{Z}^m,\alpha q}$, and $\mathbf{z}_2 \leftarrow D_{\mathbb{Z}^{\log|\mathcal{M}|},\alpha q}$, and output $\mathsf{ABE.ct}_{\vec{x}} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$, where

$$\mathbf{c}_0 = \mathbf{A}^\top \mathbf{s} + \mathbf{z}_0, \qquad \mathbf{c}_1 = [\mathbf{B}_1 + x_1\mathbf{G} \mid \cdots \mid \mathbf{B}_J + x_J\mathbf{G}]^\top \mathbf{s} + [\mathbf{R}_1 \mid \cdots \mid \mathbf{R}_J]^\top \mathbf{z}_0,$$
$$\mathbf{c}_2 = \mathbf{U}^\top \mathbf{s} + \mathbf{z}_2 + \vec{\mu}\left\lfloor \frac{q}{2} \right\rfloor.$$

$\mathsf{ABE.KGen}(\mathsf{ABE.msk}, y) \to \mathsf{ABE.sk}_y$. Parse $\mathsf{ABE.msk} = \mathbf{A}_{\sigma_0}^{-1}$. Compute $\mathbf{B}_y \leftarrow \mathsf{PubEval}(y, (\mathbf{B}_1, \ldots, \mathbf{B}_J))$, $[\mathbf{A} \mid \mathbf{B}_y]_{\sigma_0}^{-1}$ from $\mathbf{A}_{\sigma_0}^{-1}$, randomize it and output $\mathsf{ABE.sk}_y = [\mathbf{A} \mid \mathbf{B}_y]_{\sigma_1}^{-1}$.

$\mathsf{ABE.Dec}(\mathsf{ABE.sk}_y, \mathsf{ABE.ct}_{\vec{x}}) \to \vec{\mu}/\perp$. Parse $\mathsf{ABE.sk}_y = [\mathbf{A} \mid \mathbf{B}_y]_{\sigma_1}^{-1}$, $\mathsf{ABE.ct}_{\vec{x}} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$, and further parse $\mathbf{c}_1 = [\mathbf{c}_{1,1} \parallel \cdots \parallel \mathbf{c}_{1,J}]$, where $\mathbf{c}_{1,1}, \ldots, \mathbf{c}_{1,J} \in \mathbb{Z}_q^m$. Compute $\mathbf{c}_{1,y} \leftarrow \mathsf{CTEval}(y, (x_j, \mathbf{B}_j, \mathbf{c}_{1,j})_{j\in[J]})$, $[\mathbf{A} \mid \mathbf{B}_y]_{\sigma_1}^{-1}(\mathbf{U})$ from $[\mathbf{A} \mid \mathbf{B}_y]_{\sigma_1}^{-1}$, and $\mathbf{c}' = \mathbf{c}_2 - ([\mathbf{A} \mid \mathbf{B}_y]_{\sigma_1}^{-1}(\mathbf{U}))^\top \cdot [\mathbf{c}_0 \parallel \mathbf{c}_{1,y}]$. Parse $\mathbf{c}' = [c_1', \ldots, c_{\log|\mathcal{M}|}']$. For $i \in [\log|\mathcal{M}|]$, set $\mu_i = 0$ if $|c_i'| < q/4$ and $\mu_i = 1$ otherwise.

Theorem 5. *Boneh et al.'s* ABE *scheme* Π_{ABE} *satisfies the correctness and the selective* IND-CPA *security under the* $\mathsf{LWE}_{n,m,q,\alpha}$ *assumption.*

5.3 Construction

We construct a DABE scheme defined in Section 5.1 by combining with Yamada's IBE scheme Π_{Yam} [Yam17] and Boneh et al.'s ABE scheme $\Pi_{\mathsf{BGG+}}$ [BGG+14].

$\mathsf{DABE.Setup}(1^\lambda) \to (\mathsf{DABE.mpk}, \mathsf{DABE.msk})$. Run $(\mathbf{A}_0, \mathbf{A}_{0,\sigma}^{-1}), (\mathbf{A}_1, \mathbf{A}_{1,\sigma}^{-1}) \leftarrow \mathsf{TrapGen}(n, m, q)$, sample random matrices $\mathbf{B}_1, \ldots, \mathbf{B}_J, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K \leftarrow_R \mathbb{Z}_q^{n\times m}$ and $\mathbf{U} \leftarrow_R \mathbb{Z}_q^{n\times\log|\mathcal{M}|}$, and outputs $\mathsf{DABE.mpk} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \ldots, \mathbf{B}_J, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K, \mathbf{U})$ and $\mathsf{DABE.msk} = (\mathbf{A}_{0,\sigma_0}^{-1}, \mathbf{A}_{1,\sigma_0}^{-1})$, where $\mathcal{M} = \{0,1\}^{\log|\mathcal{M}|}$ is a message space.

$\mathsf{DABE.Enc}(((\vec{x}, b), \mathsf{id}), \vec{\mu}) \to \mathsf{DABE.ct}_{(\vec{x},b),\mathsf{id}}$. Parse $\mathsf{DABE.mpk} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \ldots, \mathbf{B}_J, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K, \mathbf{U})$ and $\vec{\mu} = (\mu_1, \ldots, \mu_{\log|\mathcal{M}|})$. Sample $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$ and $\mathbf{R}_{1,1}, \ldots, \mathbf{R}_{1,J} \leftarrow_R \{-1,1\}^{m\times m}$. Proceed as follows:

Case of $b=0$. Sample $\mathbf{R}_2 \leftarrow_R \{-1,1\}^{m\times m}$, $\mathbf{z}_0 \leftarrow D_{\mathbb{Z}^{2m},\alpha q}$, and $\mathbf{z}_3 \leftarrow D_{\mathbb{Z}^{\log|\mathcal{M}|},\alpha q}$, and output $\mathsf{DABE.ct}_{(\vec{x},0),\mathsf{id}} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$;

$$\mathbf{c}_0 = \mathbf{A}_0^\top \mathbf{s} + \mathbf{z}_0, \qquad \mathbf{c}_1 = [\mathbf{B}_1 + x_1\mathbf{G} \mid \cdots \mid \mathbf{B}_J + x_J\mathbf{G}]^\top \mathbf{s} + [\mathbf{R}_{1,1} \mid \cdots \mid \mathbf{R}_{1,J}]^\top \mathbf{z}_0,$$
$$\mathbf{c}_2 = [\mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]^\top \mathbf{s} + \mathbf{R}_2^\top \mathbf{z}_0, \qquad \mathbf{c}_3 = \mathbf{U}^\top \mathbf{s} + \mathbf{z}_3 + \vec{\mu}\left\lfloor \frac{q}{2} \right\rfloor.$$

Case of $b=1$. Compute $\mathbf{D}_{\mathsf{id}} \leftarrow \mathsf{PubEval}(\mathsf{id}, (\mathbf{D}_1, \dots, \mathbf{D}_K))$, sample $\mathbf{z}_0, \mathbf{z}_2 \leftarrow D_{\mathbb{Z}^{2m},\alpha' q}$, and $\mathbf{z}_3 \leftarrow D_{\mathbb{Z}^{\log|\mathcal{M}|},\alpha q}$, and output $\mathsf{DABE.ct}_{(\vec{x},1),\mathsf{id}} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$;

$$\mathbf{c}_0 = \mathbf{A}_1^\top \mathbf{s} + \mathbf{z}_0, \qquad \mathbf{c}_1 = [\mathbf{B}_1 + x_1\mathbf{G} \mid \cdots \mid \mathbf{B}_J + x_J\mathbf{G}]^\top \mathbf{s} + [\mathbf{R}_{1,1} \mid \cdots \mid \mathbf{R}_{1,J}]^\top \mathbf{z}_0,$$
$$\mathbf{c}_2 = [\mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]^\top \mathbf{s} + \mathbf{z}_1, \qquad \mathbf{c}_4 = \mathbf{U}^\top \mathbf{s} + \mathbf{z}_3 + \vec{\mu}\left\lfloor \frac{q}{2} \right\rfloor.$$

$\mathsf{DABE.KGen}(\mathsf{DABE.sk}_Y, Y') \to \mathsf{DABE.sk}_{Y'}$. Parse $\mathsf{DABE.mpk} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \dots, \mathbf{B}_J, \mathbf{D}_0, \mathbf{D}_1, \dots, \mathbf{D}_K, \mathbf{U})$ and $\mathsf{DABE.msk} = (\mathbf{A}_{0,\sigma_0}^{-1}, \mathbf{A}_{1,\sigma_0}^{-1})$. Proceed as follows:

Case of $\mathsf{DABE.sk}_Y = \mathsf{DABE.msk}$ *and* $Y' = (y,b)$. Compute $\mathbf{B}_y \leftarrow \mathsf{PubEval}(y, (\mathbf{B}_1, \dots, \mathbf{B}_J))$, $[\mathbf{A}_b \mid \mathbf{B}_y]_{\sigma_0}^{-1}$ from $\mathbf{A}_{b,\sigma_0}^{-1}$, randomize it and output $\mathsf{DABE.sk}_{(y,b)} = [\mathbf{A}_b \mid \mathbf{B}_y]_{\sigma_1}^{-1}$.

Case of $Y = (y,0)$ *and* $Y' = (y,0,\mathsf{id})$. Compute $[\mathbf{A} \mid \mathbf{B}_y \mid \mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]_{\sigma_1}^{-1}$ from $\mathsf{DABE.sk}_{(y,0)} = [\mathbf{A} \mid \mathbf{B}_y]_{\sigma_1}^{-1}$, and output $\mathsf{DABE.sk}_{(y,0),\mathsf{id}} = [\mathbf{A} \mid \mathbf{B}_y \mid \mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]_{\sigma_1}^{-1}(\mathbf{U})$.

Case of $Y = (y,1)$ *and* $Y' = (y,1,\mathsf{id})$. Compute $\mathbf{D}_{\mathsf{id}} \leftarrow \mathsf{PubEval}(\mathsf{id}, (\mathbf{D}_1, \dots, \mathbf{D}_K))$, $[\mathbf{A} \mid \mathbf{B}_y \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]_{\sigma_1}^{-1}$ from $\mathsf{DABE.sk}_{(y,1)} = [\mathbf{A} \mid \mathbf{B}_y]_{\sigma_1}^{-1}$, and output $\mathsf{DABE.sk}_{(y,1),\mathsf{id}} = [\mathbf{A} \mid \mathbf{B}_y \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]_{\sigma_1}^{-1}(\mathbf{U})$.

$\mathsf{DABE.Dec}(\mathsf{DABE.sk}_{(y,b'),\mathsf{id}'}, \mathsf{DABE.ct}_{(\vec{x},b),\mathsf{id}}) \to \vec{\mu}/\bot$. Parse $\mathsf{DABE.ct}_{(\vec{x},b),\mathsf{id}} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ and further parse $\mathbf{c}_1 = [\mathbf{c}_{1,1} \parallel \cdots \parallel \mathbf{c}_{1,J}]$, where $\mathbf{c}_{1,1}, \dots, \mathbf{c}_{1,J} \in \mathbb{Z}_q^m$. Compute $\mathbf{c}_{1,y} \leftarrow \mathsf{CTEval}(y, (x_j, \mathbf{B}_j, \mathbf{c}_{1,j})_{j\in[J]})$ and $\mathbf{c}' = \mathbf{c}_3 - \mathsf{DABE.sk}_{(y,b),\mathsf{id}}^\top \cdot [\mathbf{c}_0 \parallel \mathbf{c}_{1,y} \parallel \mathbf{c}_2]$. Parse $\mathbf{c}' = [c'_1, \dots, c'_{\log|\mathcal{M}|}]$. For $i \in [\log|\mathcal{M}|]$, set $\mu_i = 0$ if $|c'_i| < q/4$ and $\mu_i = 1$ otherwise.

5.4 Security

Theorem 6. *If Boneh et al.'s* ABE *scheme* Π_{ABE} *satisfies the selective* IND-CPA *security, the proposed* DABE *scheme* Π_{DABE} *satisfies the selective* IND-CPA *security.*

Proof of Theorem 6. We construct a reduction algorithm $\mathcal{B}$ which interacts with $\mathcal{A}$ in the selective IND-CPA security game of Π_{DABE} and breaks the selective IND-CPA security of Π_{ABE}. At the beginning of the selective IND-CPA security game of DABE, $\mathcal{A}$ declares the challenge ciphertext attribute $((\vec{x}^\star, 0), \mathsf{id}^\star)$ to $\mathcal{C}$ in the selective IND-CPA security game of $\Pi_{\mathsf{BGG+}}$. Then, $\mathcal{B}$ declares $\vec{x}^\star$ as the challenge ciphertext attribute of the selective IND-CPA security game of Boneh et al.'s $\Pi_{\mathsf{BGG+}}$. After $\mathcal{B}$ receives $\mathsf{ABE.mpk} = (\mathbf{A}, \mathbf{B}_1, \dots, \mathbf{B}_J, \mathbf{U})$, it sets $\mathbf{A}_0 = \mathbf{A}$, runs $(\mathbf{A}_1, \mathbf{A}_{1,\sigma_0}^{-1}) \leftarrow \mathsf{TrapGen}(n,m,q)$, samples $\mathbf{R}_2 \leftarrow_R \{-1,1\}^{m\times m}$ and $\mathbf{D}_1, \dots, \mathbf{D}_K \leftarrow_R \mathbb{Z}_q^{n\times m}$, computes $\mathbf{D}_0 = \mathbf{A}_0\mathbf{R}_2 - \mathsf{FRD}(\mathsf{id}^\star)\mathbf{G}$, and sends $\mathsf{DABE.mpk} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \dots, \mathbf{B}_J, \mathbf{D}_0, \mathbf{D}_1, \dots, \mathbf{D}_K, \mathbf{U})$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s secret key reveal query on $(y,0)$ such that $f(\vec{x}^\star, y) = 0$ (resp. $((y,0),\mathsf{id})$ such that $f(\vec{x}^\star, y) = 0 \wedge \mathsf{id} = \mathsf{id}^\star$), $\mathcal{B}$ makes a secret key reveal query on y, receives $\mathsf{ABE.sk}_y = [\mathbf{A}_0 \mid \mathbf{B}_y]_{\sigma_1}^{-1}$ from $\mathcal{C}$, sets $\mathsf{DABE.sk}_{(y,0)} = [\mathbf{A}_0 \mid \mathbf{B}_y]_{\sigma_1}^{-1}$ (resp. $\mathsf{DABE.sk}_{(y,0),\mathsf{id}} = [\mathbf{A}_0 \mid \mathbf{B}_y \mid \mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]_{\sigma_1}^{-1}(\mathbf{U})$), and sends it to $\mathcal{A}$. Upon $\mathcal{A}$'s secret key reveal query on $((y,0),\mathsf{id})$ such that $\mathsf{id} \neq \mathsf{id}^\star$, computes $[\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{R}_2 + (\mathsf{FRD}(\mathsf{id}) - \mathsf{FRD}(\mathsf{id}^\star))\mathbf{G}]_{\sigma_1}^{-1} = [\mathbf{A}_0 \mid \mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]_{\sigma_1}^{-1}$ from $\mathbf{R}_2$, $[\mathbf{A}_0 \mid \mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]_{\sigma_1}^{-1}(\mathbf{U})$

from $[\mathbf{A}_0 \mid \mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]^{-1}_{\sigma_1}$, and sends $\mathsf{DABE.sk}_{(y,0),\mathsf{id}} = [\mathbf{A}_0 \mid \mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]^{-1}_{\sigma_1}(\mathbf{U})$ to $\mathcal{A}$. Upon $\mathcal{A}$'s secret key reveal query on $(y,1)$ or $((y,1),\mathsf{id})$, $\mathcal{B}$ answers in the same way as the real scheme since it knows $\mathbf{A}^{-1}_{1,\sigma_0}$.

Upon $\mathcal{A}$'s challenge query on $(\vec{\mu}^\star_0, \vec{\mu}^\star_1)$, $\mathcal{B}$ makes a challenge query on $(\vec{\mu}^\star_0, \vec{\mu}^\star_1)$ to $\mathcal{C}$, and receives the ABE challenge ciphertext $\mathsf{ABE.ct}^\star_{\vec{x}^\star} = (\mathbf{c}'_0, \mathbf{c}'_1, \mathbf{c}'_2)$. $\mathcal{B}$ sets $\mathbf{c}_0 = \mathbf{c}'_0, \mathbf{c}_1 = \mathbf{c}'_1, \mathbf{c}_3 = \mathbf{c}'_2$, computes $\mathbf{c}_2 = \mathbf{R}_2^\top \mathbf{c}'_0$, and sends the DABE challenge ciphertext $\mathsf{DABE.ct}^\star_{(\vec{x}^\star,0),\mathsf{id}^\star} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ to $\mathcal{A}$. After $\mathcal{B}$ receives $\widehat{\mathsf{coin}}$ from $\mathcal{A}$, $\mathcal{B}$ sends the same $\widehat{\mathsf{coin}}$ to $\mathcal{C}$ and terminates the game.

Due to the design of Π_{DABE}, all elements created by $\mathcal{B}$ follow the same distribution as the real scheme. Although $\mathcal{B}$ makes secret key reveal queries on y upon $\mathcal{A}$'s secret key reveal query on $(y,0)$ such that $f(\vec{x}^\star, y) = 0$ or $((y,0),\mathsf{id})$ such that $f(\vec{x}^\star, y) = 0 \wedge \mathsf{id} = \mathsf{id}^\star$, they are allowed in the security game of $\Pi_{\mathsf{BGG+}}$ due to the condition $f(\vec{x}^\star, y) = 0$. Although $\mathcal{B}$ modifies the way for creating $\mathbf{D}_0$, a variant of the leftover hash lemma (Lemma 8) ensures that $\mathbf{A}_0\mathbf{R}_2$ is statistically close to uniform. Thus, the distribution of $\mathbf{D}_0 = \mathbf{A}_0\mathbf{R}_2 - \mathsf{FRD}(\mathsf{id}^\star)\mathbf{G}$ is also statistically close to uniform. Although $\mathcal{B}$ modifies the way for answering $\mathcal{A}$'s secret key reveal queries on $((y,0),\mathsf{id})$ such that $\mathsf{id} \neq \mathsf{id}^\star$, $\mathsf{DABE.sk}_{(y,0),\mathsf{id}} = [\mathbf{A}_0 \mid \mathbf{D}_0 + \mathsf{FRD}(\mathsf{id})\mathbf{G}]^{-1}_{\sigma_1}(\mathbf{U})$ follow the same distribution as the real scheme due to Lemma 7. Moreover, $\mathcal{B}$ can compute $[\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{R}_2 + (\mathsf{FRD}(\mathsf{id}) - \mathsf{FRD}(\mathsf{id}^\star))\mathbf{G}]^{-1}_{\sigma_1}$ from $\mathbf{R}_2$ since the definition of the full-rank difference map ensures that $\mathsf{FRD}(\mathsf{id}) - \mathsf{FRD}(\mathsf{id}^\star)$ is full-rank if $\mathsf{id} \neq \mathsf{id}^\star$. If $\mathsf{coin} = 0$, $\mathbf{c}'_0 = \mathbf{A}_0^\top \mathbf{s} + \mathbf{z}_0$ holds. Then, we have

$$\mathbf{c}_2 = \mathbf{R}_2^\top \mathbf{c}'_0 = (\mathbf{A}_0\mathbf{R}_2)^\top \mathbf{s} + \mathbf{R}_2^\top \mathbf{z}_0 = [\mathbf{D}_0 + \mathsf{FRD}(\mathsf{id}^\star)\mathbf{G}]^\top \mathbf{s} + \mathbf{R}_2^\top \mathbf{z}_0$$

which follows the same distribution as the real scheme. If $\mathsf{coin} = 1$, $\mathbf{c}'_0 \leftarrow_R \mathbb{Z}_q^m$ holds. Then, a variant of the leftover hash lemma (Lemma 8) ensures that $\mathbf{c}_2 = \mathbf{R}_2^\top \mathbf{c}'_0$ is statistically close to uniform. Thus, $\mathcal{B}$ perfectly simulates the real security game from $\mathcal{A}$'s view. Since $\mathcal{B}$ wins the DABE security game with overwhelming probability if $\mathcal{A}$ wins the ABE security game, we complete the proof. □

Theorem 7. *If Yamada.'s* IBE *scheme* Π_{Yam} *satisfies the adaptive* OW-CPA *security, the proposed* DABE *scheme* Π_{DABE} *satisfies the second-level adaptive* OW-CPA *security.*

Proof of Theorem 7. We construct a reduction algorithm $\mathcal{B}$ which interacts with $\mathcal{A}$ in the second-level adaptive OW-CPA security game of Π_{DABE} and breaks the adaptive OW-CPA security of Π_{Yam}. At the beginning of the second-level adaptive OW-CPA security game of DABE, $\mathcal{A}$ declares the first-level challenge ciphertext attribute $(\vec{x}^\star, 1)$, where $\vec{x}^\star = (x^\star_1, \ldots, x^\star_J)$. After $\mathcal{B}$ receives $\mathsf{IBE.mpk} = (\mathbf{A}, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K, \mathbf{U})$ from $\mathcal{C}$ in the adaptive OW-CPA security game of Π_{Yam}, it sets $\mathbf{A}_1 = \mathbf{A}$, runs $(\mathbf{A}_0, \mathbf{A}^{-1}_{0,\sigma_0}) \leftarrow \mathsf{TrapGen}(n, m, q)$, samples $\mathbf{R}_{1,1}, \ldots, \mathbf{R}_{1,J} \leftarrow_R \{-1,1\}^{m\times m}$, computes $\mathbf{B}_1 = \mathbf{A}_1\mathbf{R}_{1,1} - x^\star_1\mathbf{G}, \ldots, \mathbf{B}_J = \mathbf{A}_1\mathbf{R}_{1,J} - x^\star_J\mathbf{G}$, and sends $\mathsf{DABE.mpk} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \ldots, \mathbf{B}_J, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_K, \mathbf{U})$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s secret key reveal query on $((y,1),\mathsf{id})$ such that $f(\vec{x}^\star, y) = 1$, $\mathcal{B}$ makes a secret key reveal query on id, receives $\mathsf{IBE.sk}_{\mathsf{id}} = [\mathbf{A}_1 \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]^{-1}_{\sigma_1}$ from $\mathcal{C}$, sets $\mathsf{DABE.sk}_{(y,1),\mathsf{id}} = [\mathbf{A}_1 \mid \mathbf{B}_y \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]^{-1}_{\sigma_1}(\mathbf{U})$, and sends it to $\mathcal{A}$. Upon $\mathcal{A}$'s secret key reveal query on $(y,1)$ (resp. $((y,1),\mathsf{id})$) such that $f(\vec{x}^\star, y) = 0$, $\mathcal{B}$ runs $\mathbf{R}_y \leftarrow \mathsf{TrapEval}(y, (x^\star_j, \mathbf{R}_{1,j})_{j\in[J]}, \mathbf{A}_1)$, computes $[\mathbf{A}_1 \mid \mathbf{B}_y]^{-1}_{\sigma_1}$ from $\mathbf{R}_y$, sets $\mathsf{DABE.sk}_{(y,1)} = [\mathbf{A}_1 \mid \mathbf{B}_y]^{-1}_{\sigma_1}$ (resp. $\mathsf{DABE.sk}_{(y,1),\mathsf{id}} = [\mathbf{A}_1 \mid \mathbf{B}_y \mid \mathbf{D}_0 + \mathbf{D}_{\mathsf{id}}]^{-1}_{\sigma_1}(\mathbf{U})$), and sends it to $\mathcal{A}$. Upon $\mathcal{A}$'s secret key reveal query on $(y,0)$ or $((y,0),\mathsf{id})$, $\mathcal{B}$ answers in the same way as the real scheme since it knows $\mathbf{A}^{-1}_{1,\sigma_0}$.

Upon $\mathcal{A}$'s challenge query on $\mathsf{id}^\star$, $\mathcal{B}$ makes a challenge query on $\mathsf{id}^\star$ to $\mathcal{C}$, and receives the IBE challenge ciphertext $\mathsf{IBE.ct}^\star_{\mathsf{id}^\star} = (\mathbf{c}'_0, \mathbf{c}'_1, \mathbf{c}'_2)$. $\mathcal{B}$ sets $\mathbf{c}_0 = \mathbf{c}'_0, \mathbf{c}_2 = \mathbf{c}'_1, \mathbf{c}_3 = \mathbf{c}'_2$, computes $\mathbf{c}_1 = [\mathbf{R}_{1,1} \mid \cdots \mid \mathbf{R}_{1,J}]^\top \mathbf{c}'_0$, and sends the DABE challenge ciphertext $\mathsf{DABE.ct}^\star_{(\vec{x}^\star,1),\mathsf{id}^\star} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ to $\mathcal{A}$. After $\mathcal{B}$ receives $\widehat{\mu}$ from $\mathcal{A}$, $\mathcal{B}$ sends the same $\widehat{\mu}$ to $\mathcal{C}$ and terminates the game.

Due to the design of the proposed DABE scheme, all elements created by $\mathcal{B}$ follow the same distribution as the real scheme. Although $\mathcal{B}$ makes secret key reveal queries on id upon $\mathcal{A}$'s secret key reveal query on $((y,1),\mathsf{id})$ such that $f(\vec{x}^\star,y)=1$, they are allowed in the security game of Π_{Yam} since the definition of the second-level adaptive $\mathsf{OW\text{-}CPA}$ security ensures that $\mathsf{id}\neq\mathsf{id}^\star$ holds. Although $\mathcal{B}$ modifies the way for creating $\mathbf{B}_1,\dots,\mathbf{B}_J$, a variant of the leftover hash lemma (Lemma 8) ensures that $\mathbf{A}_1\mathbf{R}_{1,1},\dots,\mathbf{A}_1\mathbf{R}_{1,J}$ are statistically close to uniform. Thus, the distribution of $\mathbf{B}_1=\mathbf{A}_1\mathbf{R}_{1,1}-x_1^\star\mathbf{G},\dots,\mathbf{B}_J=\mathbf{A}_1\mathbf{R}_{1,J}-x_J^\star\mathbf{G}$ are also statistically close to uniform. Although $\mathcal{B}$ modifies the way for answering $\mathcal{A}$'s secret key reveal queries on $(y,1)$ and $((y,1),\mathsf{id})$ such that $f(\vec{x}^\star,y)=0$, $\mathsf{DABE.sk}_{(y,1)}=[\mathbf{A}_1\mid\mathbf{B}_y]_{\sigma_1}^{-1}$ and $\mathsf{DABE.sk}_{(y,1),\mathsf{id}}=[\mathbf{A}_1\mid\mathbf{B}_y\mid\mathbf{D}_0+\mathbf{D}_{\mathsf{id}}]_{\sigma_1}^{-1}(\mathbf{U})$ follow the same distribution as the real scheme due to Lemmata 7 and 9. Moreover, $\mathcal{B}$ can compute $[\mathbf{A}_1\mid\mathbf{B}_y]_{\sigma_1}^{-1}$ from $\mathbf{R}_y$ since Lemma 9 ensures that $\mathbf{B}_y=\mathbf{A}_1\mathbf{R}_y-y(x_1^\star,\dots,x_J^\star)\mathbf{G}$ and $y(x_1^\star,\dots,x_J^\star)\neq 0$. Since it holds that $\mathbf{c}_0'=\mathbf{A}_1^\top\mathbf{s}+\mathbf{z}_0$, we have

$$\begin{aligned}\mathbf{c}_1&=[\mathbf{R}_{1,1}\mid\cdots\mid\mathbf{R}_{1,J}]^\top\mathbf{c}_0'\\&=[\mathbf{A}_1\mathbf{R}_{1,1}\mid\cdots\mid\mathbf{A}_1\mathbf{R}_{1,J}]^\top\mathbf{s}+[\mathbf{R}_{1,1}\mid\cdots\mid\mathbf{R}_{1,J}]^\top\mathbf{z}_0\\&=[\mathbf{B}_1+x_1^\star\mathbf{G}\mid\cdots\mid\mathbf{B}_J+x_J^\star\mathbf{G}]^\top\mathbf{s}+[\mathbf{R}_{1,1}\mid\cdots\mid\mathbf{R}_{1,J}]^\top\mathbf{z}_0\end{aligned}$$

which follows the same distribution as the real scheme. Thus, $\mathcal{B}$ perfectly simulates the real security game from $\mathcal{A}$'s view. Since $\mathcal{B}$ wins the DABE security game with overwhelming probability if $\mathcal{A}$ wins the IBE security game, we complete the proof. □

6 Generic Construction of ABKFHE

In this section, we propose a generic construction of ABKFHE scheme Π_{ABKFHE}. In Section 6.1, we provide a construction of Π_{ABKFHE}. In Section 6.2, we prove the selective $\mathsf{KH\text{-}CCA}$ security.

6.1 Construction

We extend the idea explained in Section 1.3.2 and propose a generic construction of ABKFHE from MFHE, DABE, and OTS.

$\mathsf{Setup}(1^\lambda)\to(\mathsf{mpk},\mathsf{msk})$. Run $\mathsf{MFHE.pp}\leftarrow\mathsf{MFHE.Setup}(1^\lambda)$ and $(\mathsf{DABE.mpk},\mathsf{DABE.msk})\leftarrow\mathsf{DABE.Setup}(1^\lambda)$. Choose a one-time signature scheme Π_{OTS}, Output $\mathsf{mpk}=(\mathsf{MFHE.pp},\mathsf{DABE.mpk},\Pi_{\mathsf{OTS}})$ and $\mathsf{msk}=\mathsf{DABE.msk}$.

$\mathsf{Enc}(\mathsf{mpk},x,\mu)\to\mathsf{ct}_x$. Parse $\mathsf{mpk}=(\mathsf{MFHE.pp},\mathsf{DABE.mpk},\Pi_{\mathsf{OTS}})$. Run

- $(\mathsf{MFHE.pk},\mathsf{MFHE.sk})\leftarrow\mathsf{MFHE.KGen}(1^\lambda)$,
- $\mathsf{MFHE.ct}\leftarrow\mathsf{MFHE.Enc}(\mathsf{MFHE.pk},\mu)$,
- $(\mathsf{vk},\mathsf{sigk})\leftarrow\mathsf{OTS.KGen}(1^\lambda)$,
- $\mathsf{DABE.ct}_{(x,0),\mathsf{vk}}\leftarrow\mathsf{DABE.Enc}(((x,0),\mathsf{vk}),\mathsf{MFHE.sk})$,
- $\sigma\leftarrow\mathsf{Sign}\left(\mathsf{sigk},(\mathsf{vk},\mathsf{MFHE.pk},\mathsf{DABE.ct}_{(x,0),\mathsf{vk}},\mathsf{MFHE.ct})\right)$.

Output

$$\mathsf{ct}_x=(\mathsf{vk},\mathsf{MFHE.pk},\mathsf{DABE.ct}_{(x,0),\mathsf{vk}},\mathsf{MFHE.ct},\sigma).$$

We say that a pre-evaluated ciphertext ct_x is valid if σ is a valid signature for $(\mathsf{vk},\mathsf{MFHE.pk},\mathsf{DABE.ct}_{(x,0),\mathsf{vk}},\mathsf{MFHE.ct})$.

$\mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y) \to (\mathsf{dk}_y, \mathsf{hk}_y)$. Pares $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ and $\mathsf{msk} = \mathsf{DABE.msk}$. Run

- $\mathsf{DABE.sk}_{(y,0)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y, 0))$,
- $\mathsf{DABE.sk}_{(y,1)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y, 1))$.

Output $\mathsf{dk}_y = \mathsf{DABE.sk}_{(y,0)}$ and $\mathsf{hk}_y = \mathsf{DABE.sk}_{(y,1)}$.

$\mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}, \mathsf{C}) \to \mathsf{ct}_{\mathbf{x},\mathsf{C}}/\bot$. Output $\bot$ if $f(\mathbf{x}, y) = 0$ holds or there are invalid ciphertexts $\mathsf{ct}^{(\ell)}_{x^{(\ell)}}$ for some $\ell \in [L]$. Otherwise, parse $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$, $\mathsf{hk}_y = \mathsf{DABE.sk}_{(y,1)}$, and $\mathsf{ct}^{(\ell)}_{x^{(\ell)}} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)})$ for $\ell \in [L]$. Run

- $\mathsf{MFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C})$,
- $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,
- $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y,1)}, ((y, 1), \mathsf{vk}))$,
- $\sigma \leftarrow \mathsf{Sign}\left(\mathsf{sigk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}\right)$.

Output

$$\mathsf{ct}_{\mathbf{x},\mathsf{C}} = \left((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma\right).$$

We say that an evaluated ciphertext $\mathsf{ct}_{x^{(\ell)},\mathsf{C}}$ is valid if $f(\mathbf{x}, y) = 1$ holds, $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ is a valid DABE secret key for $((y, 1), \mathsf{vk})$, and σ is a valid signature for $((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}})$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y, \mathsf{ct}_x/\mathsf{ct}_{\mathbf{x},\mathsf{C}}) \to \mu/\bot$. Parse $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ and $\mathsf{dk}_y = \mathsf{DABE.sk}_{(y,0)}$. Proceed as follows.

Case of Pre-evaluated Ciphertexts. Output $\bot$ if $f(x, y) = 0$ holds or ct_x is invalid. Otherwise, parse $\mathsf{ct}_x = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma)$. Run

* $\mathsf{DABE.sk}_{(y,0),\mathsf{vk}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y,0)}, ((y, 0), \mathsf{vk}))$,
* $\mathsf{MFHE.sk} \leftarrow \mathsf{DABE.Dec}(\mathsf{DABE.sk}_{(y,0),\mathsf{vk}}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}})$,

and output $\mu \leftarrow \mathsf{MFHE.Dec}(\mathsf{MFHE.sk}, \mathsf{MFHE.ct})$.

Case of Evaluated Ciphertexts. Output $\bot$ if $f(\mathbf{x}, y) = 0$ holds or $\mathsf{ct}_{\mathbf{x},\mathsf{C}}$ is invalid. Otherwise, parse $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma)$. For $\ell \in [L]$, run

* $\mathsf{DABE.sk}_{(y,0),\mathsf{vk}^{(\ell)}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y,0)}, ((y, 0), \mathsf{vk}^{(\ell)}))$,
* $\mathsf{MFHE.sk}^{(\ell)} \leftarrow \mathsf{DABE.Dec}(\mathsf{DABE.sk}_{(y,0),\mathsf{vk}^{(\ell)}}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})$,

and output $\mu \leftarrow \mathsf{MFHE.Dec}((\mathsf{MFHE.sk}^{(\ell)})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}})$.

Theorem 8. *The proposed* ABKFHE *scheme* Π_{ABKFHE} *satisfies correctness if the underlying* MFHE *scheme* Π_{MFHE}*,* DABE *scheme* Π_{DABE}*, and one-time signature scheme* Π_{OTS} *satisfy correctness.*

Proof of Theorem 8. For every $\mu \in \mathcal{M}$, $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $f(x, y) = 1$,

- $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$;
  - $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$,
  - $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$,
  - $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ and $\mathsf{msk} = \mathsf{DABE.msk}$,
- $\mathsf{ct}_x \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, \mu)$;
  - $(\mathsf{MFHE.pk}, \mathsf{MFHE.sk}) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$,
  - $\mathsf{MFHE.ct} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}, \mu)$,
  - $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,
  - $\mathsf{DABE.ct}_{(x,0),\mathsf{vk}} \leftarrow \mathsf{DABE.Enc}(((x, 0), \mathsf{vk}), \mathsf{MFHE.sk})$,
  - $\sigma \leftarrow \mathsf{Sign}\left(\mathsf{sigk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct})\right)$.
  - $\mathsf{ct}_x = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma)$,
- $\mathsf{dk}_y \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$;
  - $\mathsf{DABE.sk}_{(y,0)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y, 0))$,
  - $\mathsf{dk}_y = \mathsf{DABE.sk}_{(y,0)}$,

the correctness of Π_{OTS} ensures that $\mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1$ holds, the correctness of Π_{DABE} ensures that $\mathsf{DABE.Dec}(\mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y,0)}, ((y, 0), \mathsf{vk})), \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}) = \mathsf{MFHE.sk}$ holds, and the correctness of Π_{MFHE} ensures that $\mathsf{MFHE.Dec}(\mathsf{MFHE.sk}, \mathsf{MFHE.ct}) = \mu$ holds. Thus, $\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y, \mathsf{ct}_x) = \mu$ holds.

For every circuit $\mathsf{C} : \mathcal{M}^L \to \mathcal{M}$, $(\mu^{(1)}, \ldots, \mu^{(L)}) \in \mathcal{M}^L$, $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $f(x, y) = 1$,

- $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$;
  - $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$,
  - $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$,
  - $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ and $\mathsf{msk} = \mathsf{DABE.msk}$,
- $\mathsf{ct}^{(\ell)}_{x^{(\ell)}} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mu^{(\ell)})$ for $\ell \in [L]$;
  - $(\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.sk}^{(\ell)}) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$,
  - $\mathsf{MFHE.ct}^{(\ell)} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^{(\ell)}, \mu^{(\ell)})$,
  - $(\mathsf{vk}^{(\ell)}, \mathsf{sigk}^{(\ell)}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,
  - $\mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}} \leftarrow \mathsf{DABE.Enc}(((x^{(\ell)}, 0), \mathsf{vk}^{(\ell)}), \mathsf{MFHE.sk}^{(\ell)})$,
  - $\sigma^{(\ell)} \leftarrow \mathsf{Sign}\left(\mathsf{sigk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)})\right)$,
  - $\mathsf{ct}^{(\ell)}_{x^{(\ell)}} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)})$,
- $\mathsf{dk}_y \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$;
  - $\mathsf{DABE.sk}_{(y,0)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y, 0))$,

- $\mathsf{dk}_y = \mathsf{DABE.sk}_{(y,0)}$,

- $\mathsf{hk}_{y'} \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y')$;
  - $\mathsf{DABE.sk}_{(y',1)} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.msk}, (y', 1))$,
  - $\mathsf{hk}_{y'} = \mathsf{DABE.sk}_{(y,1)}$,

- $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \leftarrow \mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_{y'}, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C})$;
  - $\mathsf{MFHE.ct}_\mathsf{C} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$,
  - $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,
  - $\mathsf{DABE.sk}_{(y',1),\mathsf{vk}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y',1)}, ((y', 1), \mathsf{vk}))$,
  - $\sigma \leftarrow \mathsf{Sign}\left(\mathsf{sigk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_\mathsf{C}, \mathsf{DABE.sk}_{(y',1),\mathsf{vk}})\right)$,
  - $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = \left((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_\mathsf{C}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma\right)$,

the correctness of Π_{DABE} ensures that $\mathsf{DABE.sk}_{(y',1),\mathsf{vk}}$ is a valid DABE secret key for $((y', 1), \mathsf{vk})$, the correctness of Π_{OTS} ensures that $\mathsf{OTS.Ver}(\mathsf{vk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_\mathsf{C}, \mathsf{DABE.sk}_{(y',1),\mathsf{vk}}), \sigma) = 1$ holds, the correctness of Π_{DABE} ensures that $\mathsf{DABE.Dec}(\mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y,0)}, ((y, 0), \mathsf{vk}^{(\ell)})), \mathsf{DABE.ct}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}) = \mathsf{MFHE.sk}^{(\ell)}$ holds, and the correctness of Π_{MFHE} ensures that $\mathsf{MFHE.Dec}((\mathsf{MFHE.sk}^{(\ell)})_{\ell\in[L]}, \mathsf{MFHE.ct}) = \mathsf{C}((\mu^{(\ell)})_{\ell\in[L]})$ holds. Thus, $\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y, \mathsf{ct}_{\mathbf{x},\mathsf{C}}) = \mathsf{C}((\mu^{(\ell)})_{\ell\in[L]})$ holds. □

Theorem 9. *The proposed* ABKFHE *scheme* Π_{ABKFHE} *satisfies compactness if the underlying* MFHE *scheme satisfies compactness.*

Proof of Theorem 9. For every λ,

- $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$;
  - $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$,
  - $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$,
  - $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ and $\mathsf{msk} = \mathsf{DABE.msk}$,

- $\mathsf{ct}^{(\ell)}_{x^{(\ell)}} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mu^{(\ell)})$ for $\ell \in [L]$;
  - $(\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.sk}^{(\ell)}) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$,
  - $\mathsf{MFHE.ct}^{(\ell)} \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^{(\ell)}, \mu^{(\ell)})$,
  - $(\mathsf{vk}^{(\ell)}, \mathsf{sigk}^{(\ell)}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$,
  - $\mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}} \leftarrow \mathsf{DABE.Enc}(((x^{(\ell)}, 0), \mathsf{vk}^{(\ell)}), \mathsf{MFHE.sk}^{(\ell)})$,
  - $\sigma^{(\ell)} \leftarrow \mathsf{Sign}\left(\mathsf{sigk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)})\right)$,
  - $\mathsf{ct}^{(\ell)}_{x^{(\ell)}} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)})$,

- $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \leftarrow \mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_{y'}, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C})$;

- $\mathsf{MFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C})$,
- $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{OTS.KGen}(1^{\lambda})$,
- $\mathsf{DABE.sk}_{(y',1),\mathsf{vk}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y',1)}, ((y',1), \mathsf{vk}))$,
- $\sigma \leftarrow \mathsf{Sign}\left(\mathsf{sigk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{DABE.sk}_{(y',1),\mathsf{vk}})\right)$,
- $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = \left((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma\right)$,

the compactness of Π_{MFHE} ensures that $|\mathsf{MFHE.ct}_{\mathsf{C}}|$ is independent of the size and depth of C and at most $L \cdot \mathsf{poly}(\lambda)$, and $|(\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}|$ and $|(\mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma)|$ are independent of the size and depth of C and at most $L \cdot \mathsf{poly}(\lambda)$. Thus, $|\mathsf{ct}_{\mathbf{x},\mathsf{C}}|$ is independent of the size and depth of C and at most $L \cdot \mathsf{poly}(\lambda)$. $\square$

6.2 Security

Theorem 10. *The proposed* ABKFHE *scheme* Π_{ABKFHE} *satisfies the selective* KH-CCA *security if the underlying* MFHE *scheme* Π_{MFHE} *satisfies the* IND-CPA *security,* DABE *scheme* Π_{DABE} *satisfies the selective* IND-CPA *security and the second-level adaptive* OW-CPA *security, and* OTS *scheme* Π_{OTS} *satisfies the strong* EUF-CMA *security.*

We extend the intuition of Π_{IBKFHE} explained in Section 1.3.2 and prove Theorem 10 by using a sequence of games $\mathsf{Game}_0, \cdots, \mathsf{Game}_4$. Let $\mathsf{KFHE.ct}^{\star} = (\mathsf{vk}^{\star}, \mathsf{MFHE.pk}^{\star}, \mathsf{DABE.ct}^{\star}_{(x^{\star},0),\mathsf{vk}^{\star}}, \mathsf{MFHE.ct}^{\star}, \sigma^{\star})$ denote a challenge ciphertext. We can prove Theorem 3 when $\mathsf{MFHE.ct}^{\star}$ which is an encryption of $\mu^{\star}_{\mathsf{coin}}$ becomes indistinguishable from an encryption of a random string based on the IND-CPA security of Π_{MFHE} in Game_4. To prove the task, we change $\mathsf{DABE.ct}^{\star}_{(x^{\star},0),\mathsf{vk}^{\star}}$ which is an encryption of $\mathsf{MFHE.sk}^{\star}$ to be an encryption of a random string in Game_4, where the selective IND-CPA security of Π_{DABE} ensures $\mathsf{Game}_3 \approx_c \mathsf{Game}_4$. For this purpose, we have to ensure that the challenger $\mathcal{C}$ does not use DABE secret keys $\mathsf{DABE.sk}_{(y,0)}$ and $\mathsf{DABE.sk}_{(y,0),\mathsf{vk}^{\star}}$ such that $f(x^{\star}, y) = 1$ to answer all the adversary $\mathcal{A}$'s queries. Observe that $\mathsf{DABE.sk}_{(y,0)}$ such that $f(x^{\star}, y) = 0$ (resp. $\mathsf{DABE.sk}_{(y,1)}$) suffice to answer $\mathcal{A}$'s decryption key reveal queries (resp. homomorphic evaluation key reveal queries). In other words, what all we have to ensure is that $\mathcal{A}$ does not make decryption queries on pre-evaluated ciphertexts $\mathsf{ct}_x = (\mathsf{vk}, \cdots)$ such that $\mathsf{vk} = \mathsf{vk}^{\star}$ and evaluated ciphertexts $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \cdots)_{\ell \in [L]}, \cdots)$ such that $\mathsf{vk}^{\star} \in (\mathsf{vk}^{(\ell)})_{\ell \in [L]}$. We can prove the claim for pre-evaluated ciphertexts in Game_1 by following the CHK transformation [CHK04]. In particular, the strong EUF-CMA security of Π_{OTS} ensures $\mathsf{Game}_0 \approx_c \mathsf{Game}_1$. We prove the claim for evaluated ciphertexts in Game_3 by showing that the second-level adaptive OW-CPA security of Π_{DABE} ensures $\mathsf{Game}_2 \approx_c \mathsf{Game}_3$ since $\mathcal{A}$ cannot create valid DABE secret keys $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ such that $f(x^{\star}, y) = 1$. For this purpose, we have to ensure that $\mathcal{C}$ does not use DABE secret keys $\mathsf{DABE.sk}_{(y,1)}$ and $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ such that $f(x^{\star}, y) = 1$ to answer all the adversary $\mathcal{A}$'s queries. Observe that $\mathsf{DABE.sk}_{(y,0)}$ (resp. $\mathsf{DABE.sk}_{(y,1)}$ such that $f(x^{\star}, y) = 0$) suffice to answer $\mathcal{A}$'s decryption key reveal queries (resp. homomorphic evaluation key reveal queries). However, $\mathcal{C}$ may create $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ such that $f(x^{\star}, y) = 1$ to answer $\mathcal{A}$'s evaluation queries. Let $\mathsf{ct}^{\langle i \rangle}_{\mathbf{x},\mathsf{C}} = (\cdots, \mathsf{vk}^{\langle i \rangle}, \cdots)$ denote i-th answer to $\mathcal{A}$'s evaluation queries. We show that $\mathcal{A}$ cannot make a decryption query on an evaluated ciphertext $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = (\cdots, \mathsf{vk}, \cdots)$ such that $\mathsf{vk} \in (\mathsf{vk}^{\langle i \rangle})_{i \in [Q_{\mathsf{Eval}}]}$, where Q_{Eval} denotes the maximum number of $\mathcal{A}$'s evaluation queries and the strong Q_{Eval}-EUF-CMA security of Π_{OTS} and ensures $\mathsf{Game}_1 \approx_c \mathsf{Game}_2$. Then, we can conclude that $\mathcal{A}$ cannot create valid DABE secret keys $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ such that $f(x^{\star}, y) = 1$.

Proof of Theorem 10. We prove the theorem by using a sequence of games $\mathsf{Game}_0, \cdots, \mathsf{Game}_4$.

Game_0. This is the selective KH-CCA security game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$. Hereafter, let

$$\mathsf{ct}^\star_{x^\star} = (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star, \sigma^\star)$$

denote a challenge ciphertext, where $\mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}$ and $\mathsf{MFHE.ct}^\star$ are encryptions of $\mathsf{MFHE.sk}^\star$ and $\mu^\star_{\mathsf{coin}}$, respectively. Due to the definition of the selective KH-CCA security game, $\mathcal{C}$ stores the challenge ciphertext $\mathsf{ct}^\star_{x^\star}$ and its evaluation results in the list $\mathcal{L}$.

Game_1. This is the same as Game_0 except that upon $\mathcal{A}$'s evaluation queries and decryption queries on pre-evaluated ciphertexts. Upon $\mathcal{A}$'s evaluation queries on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}} = (\mathsf{vk}^{(\ell)}, \cdots, \sigma^{(\ell)}))_{\ell\in[L]}, \mathsf{C})$ such that $\mathsf{vk}^\star \in (\mathsf{vk}^{(\ell)})_{\ell\in[L]} \wedge \mathsf{ct}^\star_{x^\star} \notin (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}$, $\mathcal{C}$ always outputs $\perp$. Upon $\mathcal{A}$'s decryption queries on $(y, \mathsf{ct}_x = (\mathsf{vk}, \cdots, \sigma))$ such that $\mathsf{vk} = \mathsf{vk}^\star$, $\mathcal{C}$ always outputs $\perp$.

The output is not $\perp$ only if $\sigma^{(\ell)}$ and σ are valid signatures accepted by $\mathsf{vk}^\star$. The strong EUF-CMA security of Π_{OTS} ensures that $\mathcal{A}$ cannot forge a signature $\sigma^{(\ell)}$ or σ. Thus, $\mathsf{Game}_1 \approx_c \mathsf{Game}_2$ holds.

Lemma 10 ($\mathsf{Game}_0 \approx_c \mathsf{Game}_1$)**.** *If Π_{OTS} satisfies the strong* EUF-CMA *security,* Game_0 *and* Game_1 *are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 10. Let F_1 denote an event that $\mathcal{A}$ makes an evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)}))_{\ell\in[L]}, \mathsf{C})$ such that

$$\begin{aligned}&\mathsf{vk}^\star \in (\mathsf{vk}^{(\ell)})_{\ell\in[L]} \wedge \mathsf{ct}^\star_{x^\star} \notin (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]} \wedge \\ &\sum_{\ell\in[L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) = L\end{aligned}$$

or a decryption query on a pre-evaluated ciphertext $\mathsf{ct}_x = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma)$ such that

$$\mathsf{vk} = \mathsf{vk}^\star \wedge \mathsf{ct}_x \neq \mathsf{ct}^\star_{x^\star} \wedge \mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1.$$

If $\sum_{\ell\in[L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) < L$ holds upon $\mathcal{A}$'s evaluation query, there is an invalid pre-evaluated ciphertext in $(\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}$ and the design of Π_{ABKFHE} ensures that an answer to the query is $\perp$. If $\mathsf{ct}_x = \mathsf{ct}^\star_{x^\star}$ holds upon $\mathcal{A}$'s decryption query, the definition of the selective KH-CCA security ensures that an answer to the query is $\perp$. If $\mathsf{OTS.Ver}(\mathsf{vk}^\star, (\mathsf{vk}^\star, \mathsf{MFHE.pk}, \mathsf{IBE.ct}_{\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 0$ holds upon $\mathcal{A}$'s decryption query, the pre-evaluated ciphertext $\mathsf{KFHE.ct}$ is invalid and the design of Π_{KFHE} ensures that an answer to the query is $\perp$. Thus, $\mathsf{Game}_0 = \mathsf{Game}_1$ holds if F_1 does not occur. Therefore, it holds that $\Pr[E_0] \leq \Pr[E_1] + \Pr[F_1]$.

We construct a reduction algorithm $\mathcal{B}_1$ which interacts with $\mathcal{A}$ against Π_{ABKFHE} and breaks the strong EUF-CMA security of Π_{OTS}. After $\mathcal{B}_1$ receives $\mathsf{vk}^\star$ from $\mathcal{C}$ in the strong EUF-CMA security game of Π_{OTS}, it runs $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$ and $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$, and sends $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. Since $\mathcal{B}_1$ knows $\mathsf{msk} = \mathsf{DABE.msk}$, it can properly answer all $\mathcal{A}$'s secret key reveal queries, homomorphic evaluation key reveal queries, evaluation queries, and decryption queries on evaluated ciphertexts.

Upon $\mathcal{A}$'s challenge query on $(\mu_0^\star, \mu_1^\star)$, $\mathcal{B}_1$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$, runs $(\mathsf{MFHE.pk}^\star, \mathsf{MFHE.sk}^\star) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$, $\mathsf{MFHE.ct}^\star \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^\star, \mu_{\mathsf{coin}}^\star)$, and $\mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star} \leftarrow \mathsf{DABE.Enc}(((x^\star,0),\mathsf{vk}^\star), \mathsf{MFHE.sk}^\star)$, makes a sign query on $(\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star)$ to $\mathcal{C}$ and receives $\sigma^\star$, and sends $\mathsf{ct}^\star_{x^\star} = (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star, \sigma^\star)$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C})$, $\mathcal{B}_1$ can check whether F_1 occurs. If $\sum_{\ell\in[L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) < L$ holds, $\mathcal{B}_1$ sends $\perp$ to $\mathcal{A}$ due to the design of Π_{ABKFHE}. If $(\mathsf{vk}^\star \notin (\mathsf{vk}^{(\ell)})_{\ell\in[L]} \vee \mathsf{ct}^\star_{x^\star} \in (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}) \wedge \sum_{\ell\in[L]} \mathsf{OTS.Ver}(\mathsf{vk}^{(\ell)}, (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}), \sigma^{(\ell)}) = L$ holds, $\mathcal{B}_1$ sends the result of $\mathsf{Eval}(\mathsf{mpk}, \mathsf{DABE}(\mathsf{DABE.msk}, (y,1)), (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C})$ to $\mathcal{A}$. Upon $\mathcal{A}$'s decryption query on a pre-evaluated ciphertexts ct_x, $\mathcal{B}_1$ can check whether F_1 occurs. If $\mathsf{ct}_x = \mathsf{ct}^\star_{x^\star} \vee \mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 0$ holds, $\mathcal{B}_1$ sends $\perp$ to $\mathcal{A}$ due to the definition of the selective $\mathsf{KH\text{-}CCA}$ security and the design of Π_{ABKFHE}. If $\mathsf{vk} \neq \mathsf{vk}^\star \wedge \mathsf{ct}_x \neq \mathsf{ct}^\star_{x^\star} \wedge \mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1$ holds, $\mathcal{B}_1$ sends the result of $\mathsf{Dec}(\mathsf{mpk}, \mathsf{DABE.KGen}(\mathsf{DABEmsk}, (y,0)), \mathsf{ct}_x)$ to $\mathcal{A}$. Otherwise, if F_1 occurs, $\mathcal{B}_1$ knows $\mathsf{ct}_x = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma)$ such that $\mathsf{vk} = \mathsf{vk}^\star \wedge \mathsf{ct}_x \neq \mathsf{ct}^\star_{x^\star} \wedge \mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1$. Then, $\mathcal{B}_1$ sends $((\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma)$ to $\mathcal{C}$ as a pair of a message and a forged signature. Since the condition $\mathsf{ct}_x \neq \mathsf{ct}^\star_{x^\star}$ ensures that $((\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma)$ is not a pair of a queried message and a returned signature, while the condition $\mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma) = 1$ ensures that σ is a valid signature of a message $(\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct})$, $\mathcal{B}_1$ breaks the strong $\mathsf{EUF\text{-}CMA}$ security of Π_{OTS} with probability 1 if F_1 occurs. Therefore, it holds that

$$\Pr[E_0] \leq \Pr[E_1] + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{OTS}},\mathcal{B}_1}(\lambda).$$

$\square$

Game_2. Let Q_{Eval} denote the maximum number of $\mathcal{A}$'s evaluation queries on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C})$ such that $f(x^\star, y) = 1$ and $\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}} = (\cdots, \mathsf{vk}^{\langle i\rangle}, \cdots)$ denote i-th answer to them. This is the same as Game_1 except that upon $\mathcal{A}$'s decryption queries on evaluated ciphertexts $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = (\cdots, \mathsf{vk}, \cdots, \sigma)$ such that $\mathsf{vk} \in \{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]} \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \{\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}}\}_{i\in[Q_{\mathsf{Eval}}]}$, $\mathcal{C}$ always outputs $\perp$.

The output is not $\perp$ only if σ is a valid signature accepted by some $\{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]}$. The strong Q_{Eval}-$\mathsf{EUF\text{-}CMA}$ security of Π_{OTS} ensures that $\mathcal{A}$ cannot forge a signature σ. Thus, $\mathsf{Game}_1 \approx_c \mathsf{Game}_2$ holds.

Lemma 11 ($\mathsf{Game}_1 \approx_c \mathsf{Game}_2$)**.** *If Π_{OTS} satisfies the strong Q_{Eval}-$\mathsf{EUF\text{-}CMA}$ security, Game_1 and Game_2 are computationally indistinguishable for any PPT $\mathcal{A}$ making at most Q_{Eval} evaluation queries on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C})$ such that $f(x^\star, y) = 1$.*

Proof of Lemma 11. Let F_2 denote an event that $\mathcal{A}$ makes a decryption query on an evaluated ciphertext $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma)$ such that

$$\mathsf{vk} \in \{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]} \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \{\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}}\}_{i\in[Q_{\mathsf{Eval}}]} \wedge$$

$$\mathsf{OTS.Ver}(\mathsf{vk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}), \sigma) = 1.$$

If $\mathsf{OTS.Ver}(\mathsf{vk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}), \sigma) = 0$ holds, the evaluated ciphertext is invalid and the design of Π_{ABKFHE} ensures that an answer to the query is $\perp$. Thus, $\mathsf{Game}_1 = \mathsf{Game}_2$ holds if F_2 does not occur. Therefore, it holds that $\Pr[E_1] \leq \Pr[E_2] + \Pr[F_2]$.

We construct a reduction algorithm $\mathcal{B}_2$ which interacts with $\mathcal{A}$ against Π_{ABKFHE} and breaks the strong Q_{Eval}-EUF-CMA security of Π_{OTS}. After $\mathcal{B}_2$ receives $x^\star$ from $\mathcal{A}$, $\mathcal{B}_2$ receives $\{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]}$ from $\mathcal{C}$. Then, it runs $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$ and $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$, and sends $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. Since $\mathcal{B}_1$ knows $\mathsf{msk} = \mathsf{DABE.msk}$, it can properly answer all $\mathcal{A}$'s secret key reveal queries, homomorphic evaluation key reveal queries, evaluation queries on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C})$ such that $f(x^\star, y) = 0$, and decryption queries on pre-evaluated ciphertexts. $\mathcal{B}_3$ answers the challenge query in the same way as in Game_1.

Upon $\mathcal{A}$'s i-th evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}, \mathsf{C})$ such that $f(x^\star, y) = 1$, $\mathcal{B}_2$ runs $\mathsf{MFHE.ct}^{\langle i\rangle}_{\mathsf{C}} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell\in[L]}, \mathsf{C})$ and $\mathsf{DABE.sk}^{\langle i\rangle}_{(y,1),\mathsf{vk}^{\langle i\rangle}} \leftarrow \mathsf{DABE.KGen}(\mathsf{DABE.sk}_{(y,1)}, ((y,1), \mathsf{vk}^{\langle i\rangle}))$, makes a sign query on $(i, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}^{\langle i\rangle}_{\mathsf{C}}, \mathsf{DABE.sk}^{\langle i\rangle}_{(y,1),\mathsf{vk}^{\langle i\rangle}}))$ to $\mathcal{C}$ and receives $\sigma^{\langle i\rangle}$, and sends $\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}^{\langle i\rangle}_{\mathsf{C}}, \mathsf{vk}^{\langle i\rangle}, \mathsf{DABE.sk}^{\langle i\rangle}_{(y,1),\mathsf{vk}^{\langle i\rangle}}, \sigma^{\langle i\rangle})$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s decryption query on an evaluated ciphertexts $\mathsf{ct}_{\mathbf{x},\mathsf{C}}$, $\mathcal{B}_2$ can check whether F_2 occurs. If $\mathsf{OTS.Ver}(\mathsf{vk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}), \sigma) = 0$ holds, $\mathcal{B}_2$ sends $\perp$ to $\mathcal{A}$ due to the design of Π_{ABKFHE}. If $(\mathsf{vk} \notin \{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]} \vee \mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \{\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}}\}_{i\in[Q_{\mathsf{Eval}}]}) \wedge \mathsf{OTS.Ver}(\mathsf{vk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}), \sigma) = 1$, $\mathcal{B}_1$ sends the result of $\mathsf{Dec}(\mathsf{mpk}, \mathsf{DABE.KGen}(\mathsf{DABEmsk}, (y,0)), \mathsf{ct}_x)$ to $\mathcal{A}$. Otherwise, if F_2 occurs, $\mathcal{B}_1$ knows $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma)$ such that $\mathsf{vk} \in \{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]} \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \{\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}}\}_{i\in[Q_{\mathsf{Eval}}]} \wedge \mathsf{OTS.Ver}(\mathsf{vk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}), \sigma) = 1$. Then, $\mathcal{B}_2$ sends $(((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}), \sigma)$ to $\mathcal{C}$ as a pair of a message and a forged signature. Since the condition $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \{\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}}\}_{i\in[Q_{\mathsf{Eval}}]}$ ensures that $((\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma)$ is not a pair of a queried message and a returned signature, while the condition $\mathsf{vk} \in \{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]} \wedge \mathsf{OTS.Ver}(\mathsf{vk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}), \sigma) = 1$ ensures that σ is a valid signature of a message $((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell\in[L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}})$, $\mathcal{B}_2$ breaks the strong Q_{Eval}-EUF-CMA security of Π_{OTS} with probability 1 if F_2 occurs. Therefore, it holds that

$$\Pr[E_1] \leq \Pr[E_2] + \mathsf{Adv}^{Q_{\mathsf{Eval}}\text{-}\mathsf{EUF}\text{-}\mathsf{CMA}}_{\Pi_{\mathsf{OTS}},\mathcal{B}_2}(\lambda).$$

$\square$

Game_3. This is the same as Game_2 except that upon $\mathcal{A}$'s decryption queries on evaluated ciphertexts $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \cdots)_{\ell\in[L]}, \cdots, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \cdots)$ such that $f(x^\star, y) = 1 \wedge \mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell\in[L]}$, $\mathcal{C}$ always outputs $\perp$.

The output is not $\perp$ only if $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ is a valid DABE secret key. The definition of the selective KH-CCA security ensures that $\mathcal{A}$ does not make a homomorphic evaluation key reveal query on y such that $f(x^\star, y) = 1$ if it can make decryption queries. The second-level adaptive OW-CPA security of Π_{DABE} ensures that $\mathcal{A}$ cannot create a valid DABE secret key. Thus, $\mathsf{Game}_2 \approx_c \mathsf{Game}_3$ holds.

Lemma 12 ($\mathsf{Game}_2 \approx_c \mathsf{Game}_3$). *If Π_{DABE} satisfies the second-level adaptive* OW-CPA *security,* Game_1 *and* Game_2 *are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 12. As in Game_2, let Q_{Eval} denote the maximum number of $\mathcal{A}$'s evaluation queries on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}, \mathsf{C})$ such that $f(x^\star, y) = 1$ and $\mathsf{ct}^{\langle i \rangle}_{\mathbf{x},\mathsf{C}} = (\cdots, \mathsf{vk}^{\langle i \rangle}, \cdots)$ denote i-th answer to them Let F_3 denote an event that $\mathcal{A}$ makes a decryption query on an evaluated ciphertext $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma)$ such that

$$f(x^\star, y) = 1 \wedge \mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell \in [L]} \wedge (\mathsf{vk} \notin \{\mathsf{vk}^{\langle i \rangle}\}_{i \in [Q_{\mathsf{Eval}}]} \vee \mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \{\mathsf{ct}^{\langle i \rangle}_{\mathbf{x},\mathsf{C}}\}_{i \in [Q_{\mathsf{Eval}}]}) \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \mathcal{L}$$

and $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ is a valid DABE secret key. If $\mathsf{vk} \in \{\mathsf{vk}^{\langle i \rangle}\}_{i \in [Q_{\mathsf{Eval}}]} \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \{\mathsf{ct}^{\langle i \rangle}_{\mathbf{x},\mathsf{C}}\}_{i \in [Q_{\mathsf{Eval}}]}$ holds, an answer to the query is $\perp$ as we modified in Game_2. If $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \mathcal{L}$ holds, an answer to the query is $\perp$ due to the definition of the selective KH-CCA security. If $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ is an invalid DABE secret key, the evaluated ciphertext is invalid and and the design of Π_{ABKFHE} ensures that an answer to the query is $\perp$. Thus, $\mathsf{Game}_2 = \mathsf{Game}_3$ holds if F_3 does not occur. Therefore, it holds that $\Pr[E_2] \leq \Pr[E_3] + \Pr[F_3]$. We call $\mathcal{A}$'s decryption query a *critical decryption query* if F_3 occurs. Hereafter, let $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = (\cdots, \widehat{\mathsf{vk}}, \mathsf{DABE.sk}_{(y,1),\widehat{\mathsf{vk}}}, \cdots)$ denote an evaluated ciphertext on which $\mathcal{A}$ makes a critical decryption query.

We construct a reduction algorithm $\mathcal{B}_3$ which interacts with $\mathcal{A}$ against Π_{ABKFHE} and breaks the second-level adaptive OW-CPA security of Π_{DABE}. After $\mathcal{B}_3$ receives $x^\star$ from $\mathcal{A}$, it declares $(x^\star, 1)$ to $\mathcal{C}$ and receives $\mathsf{DABE.mpk}$. Then, it runs $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$, chooses a one-time signature scheme Π_{OTS}, and sends $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. Upon $\mathcal{A}$'s decryption key reveal query (resp. homomorphic evaluation key reveal query) on y, $\mathcal{B}_3$ makes a DABE secret key reveal query on $(y, 0)$ (resp. $(y, 1)$) to $\mathcal{C}$ and receives $\mathsf{DABE.sk}_{(y,0)}$ (resp. $\mathsf{DABE.sk}_{(y,1)}$), and sends it to $\mathcal{A}$. Upon $\mathcal{A}$'s decryption query on a pre-evaluated ciphertext (y, ct_x), $\mathcal{B}_3$ makes a DABE secret key reveal query on $(y, 0)$ to $\mathcal{C}$ and receives $\mathsf{DABE.sk}_{(y,0)}$, and answers in the same way as in Game_2. $\mathcal{B}_3$ answers $\mathcal{A}$'s challenge query in the same way as in Game_2.

Upon $\mathcal{A}$'s evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}} = (\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}}, \mathsf{MFHE.ct}^{(\ell)}, \sigma^{(\ell)}))_{\ell \in [L]}, \mathsf{C})$, $\mathcal{B}_3$ sends $\perp$ to $\mathcal{A}$ if $\mathsf{vk}^\star \in (\mathsf{vk}^{(\ell)})_{\ell \in [L]} \wedge \mathsf{ct}^\star_{x^\star} \notin (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}$ holds as we modified in Game_1. Otherwise, $\mathcal{B}_3$ runs $\mathsf{MFHE.ct}_{\mathsf{C}} \leftarrow \mathsf{MFHE.Eval}((\mathsf{MFHE.pk}^{(\ell)}, \mathsf{MFHE.ct}^{(\ell)})_{\ell \in [L]}, \mathsf{C})$ and $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$, makes a DABE secret key reveal query on $((y, 1), \mathsf{vk})$ to $\mathcal{C}$ and receives $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$, further runs $\sigma \leftarrow \mathsf{Sign}(\mathsf{sigk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}))$, and sends $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y,1),\mathsf{vk}}, \sigma)$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s decryption query on an evaluated ciphertext $(y', \mathsf{ct}_{\mathbf{x},\mathsf{C}})$, $\mathcal{B}_3$ can check whether F_3 occurs. If $\mathsf{vk} \in \{\mathsf{vk}^{\langle i \rangle}\}_{i \in [Q_{\mathsf{Eval}}]} \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \{\mathsf{ct}^{\langle i \rangle}_{\mathbf{x},\mathsf{C}}\}_{i \in [Q_{\mathsf{Eval}}]}$ holds, $\mathcal{B}_3$ sends $\perp$ to $\mathcal{A}$ as we modified in Game_2. $\mathcal{B}_3$ also sends $\perp$ to $\mathcal{A}$ if $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \mathcal{L}$ holds due to the definition of the selective KH-CCA security. If $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ is an invalid DABE secret key, $\mathcal{B}_3$ sends $\perp$ to $\mathcal{A}$ due to the design of Π_{ABKFHE}. If $f(x^\star, y) = 0 \vee \mathsf{vk}^\star \notin \{\mathsf{vk}^{(\ell)}\}_{\ell \in [L]}$ holds and $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ is a valid DABE secret key, $\mathcal{B}_3$

makes a DABE secret key reveal query on $(y,0)$ and receives $\mathsf{DABE.sk}_{(y,0)}$, then sends the result of $\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y = \mathsf{DABE.sk}_{(y,0)}, \mathsf{ct}_{\mathbf{x},\mathsf{C}})$ to $\mathcal{A}$. Otherwise, if F_3 occurs, $\mathcal{B}_3$ knows a valid DABE secret key $\mathsf{DABE.sk}_{(y,1),\widehat{\mathsf{vk}}}$. Then, $\mathcal{B}_3$ makes a DABE challenge query on $\widehat{\mathsf{vk}}$ to $\mathcal{C}$ and receives the DABE challenge ciphertext $\mathsf{DABE.ct}^\star_{(x^\star,1),\widehat{\mathsf{vk}}}$, sends the result of $\mathsf{DABE.Dec}(\mathsf{DABE.sk}_{(y,1),\widehat{\mathsf{vk}}}, \mathsf{DABE.ct}^\star_{(x^\star,1),\widehat{\mathsf{vk}}})$ to $\mathcal{C}$. If F_3 occurs, $f(x^\star, y) = 1$ holds. Thus, $\mathcal{B}_3$ can break the second-level adaptive $\mathsf{OW\text{-}CPA}$ security with overwhelming probability if $\mathcal{B}_3$ never makes a DABE secret key reveal query on $(y,1)$ or $((y,1),\widehat{\mathsf{vk}})$ such that $f(x^\star,y) = 1$. Although $\mathcal{B}_3$ makes a DABE secret key reveal query on $(y,1)$ upon $\mathcal{A}$'s homomorphic evaluation key reveal query on y, it holds that $f(x^\star,y) = 0$ since the definition of the selective $\mathsf{KH\text{-}CCA}$ security ensures that $\mathcal{A}$ cannot make decryption queries after $\mathcal{A}$'s homomorphic evaluation key reveal query on y such that $f(x^\star,y) = 1$.

What only we have to check is that $\mathcal{B}_3$ does not make a DABE secret key reveal query on $((y,1),\widehat{\mathsf{vk}})$ such that $f(x^\star,y) = 1$ upon $\mathcal{A}$'s evaluation queries. Observe that F_3 occurs when $\mathcal{A}$ makes a critical decryption query on an evaluated ciphertext $\mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \cdots)_{\ell\in[L]}, \cdots, \widehat{\mathsf{vk}}, \mathsf{DABE.sk}_{(y,1),\widehat{\mathsf{vk}}}, \cdots)$ such that $f(x^\star,y) = 1 \wedge \mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell\in[L]} \wedge (\widehat{\mathsf{vk}} \notin \{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]} \vee \mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \{\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}}\}_{i\in[Q_{\mathsf{Eval}}]}) \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \mathcal{L}$ and $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ is a valid DABE secret key. Moreover, $\mathcal{B}_3$ makes DABE secret key reveal queries on $((y,1),\mathsf{vk}^{\langle i\rangle})$ for $i \in [Q_{\mathsf{Eval}}]$ upon $\mathcal{A}$'s evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}} = (\mathsf{vk}^{(\ell)}, \cdots)_{\ell\in[L]}, \mathsf{C})$ only if $\mathsf{vk}^\star \notin (\mathsf{vk}^{(\ell)})_{\ell\in[L]} \vee \mathsf{ct}^\star_{x^\star} \in (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}$ holds. If $\mathcal{A}$'s critical decryption query satisfies $\widehat{\mathsf{vk}} \notin \{\mathsf{vk}^{\langle i\rangle}\}_{i\in[Q_{\mathsf{Eval}}]}$, $\mathcal{B}_3$ does not make a DABE secret key reveal query on $((y,1),\widehat{\mathsf{vk}})$. Hereafter, we focus on the other case that $\mathcal{A}$'s critical decryption query satisfies $f(x^\star,y) = 1 \wedge \mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell\in[L]} \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \{\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}}\}_{i\in[Q_{\mathsf{Eval}}]} \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \notin \mathcal{L}$ and $\mathsf{DABE.sk}_{(y,1),\mathsf{vk}}$ is a valid DABE secret key. If $\mathcal{A}$'s critical decryption query satisfies $\mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell\in[L]} \wedge \mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \{\mathsf{ct}^{\langle i\rangle}_{\mathbf{x},\mathsf{C}}\}_{i\in[Q_{\mathsf{Eval}}]}$, $\mathcal{A}$ has made an evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}} = (\mathsf{vk}^{(\ell)}, \cdots)_{\ell\in[L]}, \mathsf{C})$ such that $\mathsf{vk}^\star \in (\mathsf{vk}^{(\ell)})_{\ell\in[L]}$. Nevertheless, the evaluation query has to satisfy $\mathsf{ct}^\star_{x^\star} \in (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell\in[L]}$; in other words, the answer to the evaluation query has to satisfy $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \mathcal{L}$. Since F_3 never happens in this case, we can conclude that $\mathcal{B}_3$ does not make a DABE secret key reveal query on $((y,1),\widehat{\mathsf{vk}})$ such that $f(x^\star,y) = 1$. Therefore, it holds that

$$\Pr[E_2] \le \Pr[E_3] + \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\Pi_{\mathsf{DABE}},\mathcal{B}_3}(\lambda) + \mathsf{negl}(\lambda).$$

□

Game_4. This is the same as Game_3 except that $\mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}$ is an encryption of a random string sampled independently from $\mathsf{MFHE.sk}^\star$.

The selective $\mathsf{IND\text{-}CPA}$ security of the Π_{DABE} ensures that $\mathsf{Game}_3 \approx_c \mathsf{Game}_4$ holds. In short, the reduction algorithm runs $(\mathsf{vk}^\star, \mathsf{sigk}^\star) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$ at the beginning of the security game. After $\mathcal{A}$ declares the challenge attribute $x^\star$ in the selective $\mathsf{KH\text{-}CCA}$ security game, the reduction algorithm declares $(x^\star, 0, \mathsf{vk}^\star)$ as the challenge ciphertext attribute of DABE security game. In the challenge phase, the reduction algorithm runs $(\mathsf{MFHE.pk}^\star, \mathsf{MFHE.sk}^\star) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$, samples a random string $\mu^\star$ whose length is the same as $\mathsf{MFHE.sk}^\star$ but the distribution is independent of $\mathsf{MFHE.sk}^\star$. Then, the reduction algorithm declares $(\mathsf{MFHE.sk}^\star, \mu^\star)$ as the challenge messages in the DABE security game and receives the challenge ciphertext $\mathsf{DABE.ct}^\star_{(x^\star,0,\mathsf{vk}^\star)}$ from the DABE challenger. The reduction algorithm can create the other elements of the challenge ciphertext by itself. Due to the modifications in $\mathsf{Game}_1, \mathsf{Game}_2$, and Game_3, the reduction algorithm can answer all $\mathcal{A}$'s queries by making DABE secret key reveal queries on (y,b) or $((y,b),\mathsf{vk})$ such that $f(x^\star,y) = 0 \vee b = 1 \vee \mathsf{vk} \neq \mathsf{vk}^\star$. Thus, it holds that $\mathsf{Game}_3 \approx_c \mathsf{Game}_4$.

Lemma 13 ($\mathsf{Game}_3 \approx_c \mathsf{Game}_4$)**.** *If Π_{DABE} satisfies the selective* IND-CPA *security,* Game_3 *and* Game_4 *are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 13. We construct a reduction algorithm $\mathcal{B}_4$ which interacts with $\mathcal{A}$ against Π_{ABKFHE} and breaks the selective IND-CPA security of Π_{DABE}. At the beginning of the game, $\mathcal{B}_3$ runs $(\mathsf{vk}^\star, \mathsf{sigk}^\star) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$. After $\mathcal{B}_4$ receives $x^\star$ from $\mathcal{A}$, it declares $((x^\star, 0), \mathsf{vk}^\star)$ to $\mathcal{C}$ and receives $\mathsf{DABE.mpk}$. Then, it runs $\mathsf{MFHE.pp} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$, chooses a one-time signature scheme Π_{OTS}, and sends $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. Upon $\mathcal{A}$'s decryption key reveal query (resp. homomorphic evaluation key reveal query) on y, $\mathcal{B}_4$ makes a DABE secret key reveal query on $(y, 0)$ (resp. $(y, 1)$) to $\mathcal{C}$ and receives $\mathsf{DABE.sk}_{(y,0)}$ (resp. $\mathsf{DABE.sk}_{(y,1)}$), and sends it to $\mathcal{A}$. Upon $\mathcal{A}$'s evaluation query on $(y, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}, \mathsf{C})$, $\mathcal{B}_4$ makes a DABE secret key reveal query on $(y, 1)$ to $\mathcal{C}$ and receives $\mathsf{DABE.sk}_{(y,1)}$, then sends the result of $\mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_y = \mathsf{DABE.sk}_{(y,1)}, (\mathsf{ct}^{(\ell)}_{x^{(\ell)}})_{\ell \in [L]}, \mathsf{C})$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s decryption query on a pre-evaluated ciphertext $(y, \mathsf{ct}_x = (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}, \sigma))$, $\mathcal{B}_4$ sends $\perp$ to $\mathcal{A}$ if $\mathsf{vk} = \mathsf{vk}^\star$ holds as we modified in Game_1. $\mathcal{B}_4$ also sends $\perp$ to $\mathcal{A}$ if $\mathsf{OTS.Ver}(\mathsf{vk}, (\mathsf{vk}, \mathsf{MFHE.pk}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}, \mathsf{MFHE.ct}), \sigma)$ holds due to the definition of the selective KH-CCA security. Otherwise, $\mathcal{B}_4$ makes a DABE secret key reveal query on $((y, 0), \mathsf{vk})$ to $\mathcal{C}$ and receives $\mathsf{DABE.sk}_{(y,0),\mathsf{vk}}$, then sends the result of $\mathsf{MFHE.Dec}(\mathsf{DABE.Dec}(\mathsf{DABE.sk}_{(y,0),\mathsf{vk}}, \mathsf{DABE.ct}_{(x,0),\mathsf{vk}}), \mathsf{MFHE.ct})$ to $\mathcal{A}$. Upon $\mathcal{A}$'s decryption query on an evaluated ciphertext $(y, \mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y',1),\mathsf{vk}}, \sigma))$, $\mathcal{B}_4$ sends $\perp$ to $\mathcal{A}$ if $f(x^\star, y) = 1 \wedge \mathsf{vk}^\star \in \{\mathsf{vk}^{(\ell)}\}_{\ell \in [L]}$ holds as we modified in Game_3. $\mathcal{B}_3$ also sends $\perp$ to $\mathcal{A}$ if $\mathsf{ct}_{\mathbf{x},\mathsf{C}} \in \mathcal{L}$ holds due to the definition of the selective KH-CCA security. $\mathcal{B}_3$ also sends $\perp$ to $\mathcal{A}$ if $\mathsf{DABE.sk}_{(y',1),\mathsf{vk}}$ is an invalid DABE secret key or $\mathsf{OTS.Ver}(\mathsf{vk}, ((\mathsf{vk}^{(\ell)}, \mathsf{MFHE.pk}^{(\ell)}, \mathsf{DABE.ct}^{(\ell)}_{(x^{(\ell)},0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{MFHE.ct}_{\mathsf{C}}, \mathsf{vk}, \mathsf{DABE.sk}_{(y',1),\mathsf{vk}}), \sigma) = 0$ holds due to the design of Π_{ABKFHE}. Otherwise, $\mathcal{B}_4$ makes DABE secret key reveal queries on $((y, 0), \mathsf{vk}^{(\ell)})$ to $\mathcal{C}$ and receives $\mathsf{DABE.sk}_{(y,0),\mathsf{vk}^{(\ell)}}$ for $\ell \in [L]$, then sends the result of $\mathsf{MFHE.Dec}((\mathsf{DABE.Dec}(\mathsf{DABE.sk}_{(y,0),\mathsf{vk}^{(\ell)}})_{\ell \in [L]}, \mathsf{DABE.ct}^{(\ell)}_{(x,0),\mathsf{vk}^{(\ell)}}), \mathsf{MFHE.ct})$ to $\mathcal{A}$.

Upon $\mathcal{A}$'s challenge query on $(\mu_0^\star, \mu_1^\star)$, $\mathcal{B}_4$ samples $\mathsf{coin} \leftarrow_R \{0, 1\}$, runs $(\mathsf{MFHE.pk}^\star, \mathsf{MFHE.sk}^\star) \leftarrow \mathsf{MFHE.KGen}(1^\lambda)$ and $\mathsf{MFHE.ct}^\star \leftarrow \mathsf{MFHE.Enc}(\mathsf{MFHE.pk}^\star, \mu^\star_{\mathsf{coin}})$, makes a DABE challenge query on $(\mathsf{MFHE.sk}^\star, \mu^\star)$ to $\mathcal{C}$, where $\mu^\star$ is a random string with the same length as $\mathsf{MFHE.sk}^\star$, receives $\mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}$, further runs $\sigma^\star \leftarrow \mathsf{Sign}(\mathsf{sigk}^\star, (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star))$, and sends $\mathsf{ct}^\star_{x^\star} = (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star, \sigma^\star)$ to $\mathcal{A}$. After $\mathcal{B}_4$ receives $\widehat{\mathsf{coin}}$ from $\mathcal{A}$, $\mathcal{B}_4$ sends 0 to $\mathcal{C}$ if $\widehat{\mathsf{coin}} = \mathsf{coin}$ and 1 to $\mathcal{C}$ otherwise.

Although $\mathcal{B}_4$ makes a DABE secret key reveal queries on $(y, 0)$ to $\mathcal{C}$ upon $\mathcal{A}$'s decryption key reveal query on y, the definition of the selective KH-CCA security ensures that $f(x^\star, y) = 0$. Although $\mathcal{B}_4$ makes a DABE secret key reveal queries on $((y, 0), \mathsf{vk})$ to $\mathcal{C}$ upon $\mathcal{A}$'s decryption query on a pre-evaluated ciphertext $(y, \mathsf{ct}_x = (\mathsf{vk}, \cdots))$, the modification in Game_1 ensures that $\mathsf{vk} \neq \mathsf{vk}^\star$. Although $\mathcal{B}_4$ makes DABE secret key reveal queries on $((y, 0), \mathsf{vk}^{(\ell)})_{\ell \in [L]}$ to $\mathcal{C}$ upon $\mathcal{A}$'s decryption query on an evaluated ciphertext $(y, \mathsf{ct}_{\mathbf{x},\mathsf{C}} = ((\mathsf{vk}^{(\ell)}, \cdots)_{\ell \in [L]}, \cdots))$, the modification in Game_3 ensures that $f(x^\star, y) = 0 \vee \mathsf{vk} \neq \mathsf{vk}^\star$. Thus, it holds that

$$|\Pr[E_3] - \Pr[E_4]| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{DABE}}, \mathcal{B}_4}(\lambda).$$

□

Lemma 14 (Selective KH-CCA Security in Game_4)**.** *If Π_{MFHE} satisfies the* IND-CPA *security,* Π_{ABKFHE} *satisfies the selective* KH-CCA *security in* Game_4.

Proof of Lemma 14. We construct a reduction algorithm $\mathcal{B}_5$ which interacts with $\mathcal{A}$ against Π_{ABKFHE} and breaks the $\mathsf{IND\text{-}CPA}$ security of Π_{MFHE}. After $\mathcal{B}_5$ receives $(\mathsf{MFHE.pp}, \mathsf{MFHE.pk}^\star)$ from $\mathcal{C}$, it runs $(\mathsf{DABE.mpk}, \mathsf{DABE.msk}) \leftarrow \mathsf{DABE.Setup}(1^\lambda)$, chooses a one-time signature scheme Π_{OTS}, and sends $\mathsf{mpk} = (\mathsf{MFHE.pp}, \mathsf{DABE.mpk}, \Pi_{\mathsf{OTS}})$ to $\mathcal{A}$. Since $\mathcal{B}_5$ knows $\mathsf{DABE.msk}$, it can properly answer all $\mathcal{A}$'s decryption key reveal queries, homomorphic evaluation key reveal queries, evaluation queries, and decryption queries.

Upon $\mathcal{A}$'s challenge query on $(\mu_0^\star, \mu_1^\star)$, $\mathcal{B}_5$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$ and $\mu^\star \leftarrow_R \mathcal{M}$, makes a MFHE challenge query on the same $(\mu_0^\star, \mu_1^\star)$ to $\mathcal{C}$ and receives $\mathsf{MFHE.ct}^\star$, runs $(\mathsf{vk}^\star, \mathsf{sigk}^\star) \leftarrow \mathsf{OTS.KGen}(1^\lambda)$, $\mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star} \leftarrow \mathsf{DABE.Enc}(((x^\star,0),\mathsf{vk}^\star),\mu^\star)$, and $\sigma^\star \leftarrow \mathsf{Sign}(\mathsf{sigk}^\star, (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star))$, then sends $\mathsf{ct}^\star_{x^\star} = (\mathsf{vk}^\star, \mathsf{MFHE.pk}^\star, \mathsf{DABE.ct}^\star_{(x^\star,0),\mathsf{vk}^\star}, \mathsf{MFHE.ct}^\star, \sigma^\star)$ to $\mathcal{A}$. After $\mathcal{B}_5$ receives $\widehat{\mathsf{coin}}$ from $\mathcal{A}$, $\mathcal{B}_5$ sends the same $\widehat{\mathsf{coin}}$ to $\mathcal{C}$.

If $\mathsf{MFHE.ct}^\star$ is an encryption of $\mu_0^\star$ (resp. $\mu_1^\star$), $\mathsf{ct}^\star_{x^\star}$ is also an encryption of $\mu_0^\star$ (resp. $\mu_1^\star$). Therefore, it holds that

$$\left|\Pr[E_4] - \frac{1}{2}\right| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{MFHE}},\mathcal{B}_5}(\lambda).$$

□

We complete the proof of Theorem 3 since it holds that

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{KH\text{-}CCA}}_{\Pi_{\mathsf{ABKFHE}},\mathcal{A}}(\lambda)\\
&= \left|\Pr[E_0] - \frac{1}{2}\right|\\
&\leq \sum_{i\in[4]} |\Pr[E_{i-1}] - \Pr[E_i]| + \left|\Pr[E_4] - \frac{1}{2}\right|\\
&\leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{OTS}},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{Q_{\mathsf{Eval}}\text{-}\mathsf{EUF\text{-}CMA}}_{\Pi_{\mathsf{OTS}},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\Pi_{\mathsf{DABE}},\mathcal{B}_3}(\lambda) + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{DABE}},\mathcal{B}_4}(\lambda) + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi_{\mathsf{MFHE}},\mathcal{B}_5}(\lambda).
\end{aligned}$$

□

7 Emura et al.'s KHPKE Scheme under the Matrix DDH Assumption

In this section, we provide a simpler proof of Emura et al.'s KHPKE scheme Π_{KHPKE} [EHN+18] if it is instantiated under the matrix DDH assumption. In Section 7.1, we review cyclic groups and the matrix DDH assumption. In Section 7.2, we review Emura et al.'s KHPKE scheme instantiated under the matrix DDH assumption. In Section 7.3, we prove the KH-CCA security.

7.1 Cyclic Groups

Let $\widehat{\mathcal{G}}$ be a cyclic group generator that takes the security parameter 1^λ as input, and outputs $(p, \mathbb{G}, g)$, where p is a $\Theta(\lambda)$-bit prime number, $\mathbb{G}$ is a cyclic group of order p, and g is a generator of $\mathbb{G}$. For simplicity, let $\widehat{\mathcal{G}}(1^\lambda) := (p, \mathbb{G}, g)$ denote the output of $\widehat{\mathcal{G}}(1^\lambda)$. Let $1_{\mathbb{G}}$ denote an identity element of $\mathbb{G}$. For $a \in \mathbb{Z}_p$ and $\mathbf{a} = (a_1, \ldots, a_d) \in \mathbb{Z}_p^d$, let $[a] := g^a \in \mathbb{G}_1$ and $[\mathbf{a}] := ([a_1], \ldots, [a_d]) \in \mathbb{G}_1^d$. We use the same notation for a matrix $[\mathbf{A}]$. Let $\mathcal{D}_k$ be an efficiently sampleable matrix distribution [EHK+17] that outputs $(\mathbf{A}, \mathbf{a}^\perp) \in \mathbb{Z}_p^{(k+1)\times k} \times \mathbb{Z}_p^{k+1}$ such that $\mathbf{A}^\top \cdot \mathbf{a}^\perp = \mathbf{0}$ and $\mathbf{a}^\perp \neq \mathbf{0}$.

We use the following matrix DDH assumption to prove the KH-CCA security of Emura et al.'s KHPKE scheme Π_{KHPKE}.

Definition 24 (Matrix DDH Assumption)**.** *For a cyclic group $\widehat{\mathcal{G}}(1^\lambda) = (p, \mathbb{G}, g)$, an advantage for solving the matrix DDH problem by an algorithm $\mathcal{A}$ is defined to be*

$$\mathsf{Adv}^{\mathsf{mDDH}_{\mathbb{G}}}_{\mathcal{A}}(\lambda) := \left| \Pr\left[\mathcal{A}(\widehat{\mathcal{G}}(1^\lambda), [\mathbf{A}], [\mathbf{As}]) \to 1\right] - \Pr\left[\mathcal{A}(\widehat{\mathcal{G}}(1^\lambda), [\mathbf{A}], [\mathbf{v}]) \to 1\right] \right|,$$

where $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$, $\mathbf{s} \leftarrow_R \mathbb{Z}_p^k$, and $\mathbf{v} \leftarrow_R \mathbb{Z}_p^{k+1}$. We say that the matrix DDH assumption holds if it is negligible for all PPT $\mathcal{A}$.

7.2 Scheme

We describe Emura et al.'s KHPKE scheme [EHN+18] Π_{KHPKE} instantiated under the matrix DDH assumption.

$\mathsf{KHPKE.KGen}(1^\lambda) \to (\mathsf{KHPKE.pk}, \mathsf{KHPKE.dk}, \mathsf{KHPKE.hk})$. Run $(p, \mathbb{G}, g) \leftarrow \widehat{\mathcal{G}}(1^\lambda)$ and choose a collision-resistant hash function $H \leftarrow_R \mathcal{H}$, where $H : \{0,1\}^* \to \mathbb{Z}_p$. Sample $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$ and random vectors $(\mathbf{u}_\iota)_{\iota \in [0,3]} \leftarrow_R \mathbb{Z}_p^{k+1}$, then output

$$\mathsf{KHPKE.pk} := \left(\widehat{\mathcal{G}}(1^\lambda), [\mathbf{A}], ([\mathbf{A}^\top \mathbf{u}_\iota])_{\iota \in [0,3]}, H\right),$$

$\mathsf{KHPKE.dk} := (\mathbf{u}_\iota)_{\iota \in [0,3]}$, and $\mathsf{KHPKE.hk} := (\mathbf{u}_\iota)_{\iota \in [2]}$.

$\mathsf{KHPKE.Enc}(\mathsf{KHPKE.pk}, \mu) \to \mathsf{KHPKE.ct}$. Sample $\mathbf{s} \leftarrow_R \mathbb{Z}_p^k$ and output $\mathsf{KHPKE.ct} := (\mathsf{KHPKE.ct}_0, \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi, \mathsf{KHPKE.}\pi')$;

$$\mathsf{KHPKE.ct}_0 = [\mathbf{As}], \qquad \mathsf{KHPKE.ct}_\mu = \mu \cdot [\mathbf{s}^\top \mathbf{A}^\top \mathbf{u}_0] \qquad \mathsf{KHPKE.}\pi = [\mathbf{s}^\top \mathbf{A}^\top (\mathbf{u}_1 + h \cdot \mathbf{u}_2)],$$
$$\mathsf{KHPKE.}\pi' = [\mathbf{s}^\top \mathbf{A}^\top \mathbf{u}_3],$$

where $h = H(\mathsf{KHPKE.ct}_0, \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi')$.

$\mathsf{KHPKE.Eval}(\mathsf{KHPKE.pk}, \mathsf{KHPKE.hk}, (\mathsf{KHPKE.ct}^{(\ell)})_{\ell \in [L]}) \to \mathsf{KHPKE.ct}/\perp$. Parse $\mathsf{KHPKE.hk} = (\mathbf{u}_\iota)_{\iota \in [2]}$ and $\mathsf{KHPKE.ct}^{(\ell)} = (\mathsf{KHPKE.ct}_0^{(\ell)} = [\mathbf{c}^{(\ell)}], \mathsf{KHPKE.ct}_\mu^{(\ell)}, \mathsf{KHPKE.}\pi^{(\ell)}, \mathsf{KHPKE.}\pi'^{(\ell)})$. Output $\perp$ if there is some $\ell \in [L]$ which does not satisfy

$$\mathsf{KHPKE.}\pi^{(\ell)} = [(\mathbf{c}^{(\ell)})^\top \cdot (\mathbf{u}_1 + h^{(\ell)} \cdot \mathbf{u}_2)], \tag{5}$$

where $h^{(\ell)} = H(\mathsf{KHPKE.ct}_0^{(\ell)}, \mathsf{KHPKE.ct}_\mu^{(\ell)}, \mathsf{KHPKE.}\pi'^{(\ell)})$. Otherwise, run $\mathsf{KHPKE.ct}^{(0)} \leftarrow \mathsf{KHPKE.Enc}(\mathsf{KHPKE.pk}, 1_{\mathbb{G}})$ and output $\mathsf{KHPKE.ct} := (\mathsf{KHPKE.ct}_0, \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi, \mathsf{KHPKE.}\pi')$;

$$\mathsf{KHPKE.ct}_0 = \prod_{\ell \in [0,L]} \mathsf{KHPKE.ct}_0^{(\ell)}, \qquad \mathsf{KHPKE.ct}_\mu = \prod_{\ell \in [0,L]} \mathsf{KHPKE.ct}_\mu^{(\ell)},$$
$$\mathsf{KHPKE.}\pi = [\mathbf{c}^\top \cdot (\mathbf{u}_1 + h \cdot \mathbf{u}_2)], \qquad \mathsf{KHPKE.}\pi' = \prod_{\ell \in [0,L]} \mathsf{KHPKE.}\pi'^{(\ell)},$$

where $\mathsf{KHPKE.ct}_0 = [\mathbf{c}]$ and $h = H(\mathsf{KHPKE.ct}_0, \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi')$.

$\mathsf{KHPKE.Dec}(\mathsf{KHPKE.pk}, \mathsf{KHPKE.dk}, \mathsf{KHPKE.ct}) \to \mu/\perp$. Parse $\mathsf{KHPKE.dk} = (\mathbf{u}_\iota)_{\iota \in [0,3]}$ and $\mathsf{KHPKE.ct} = (\mathsf{KHPKE.ct}_0 = [\mathbf{c}], \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE.}\pi, \mathsf{KHPKE.}\pi')$. Output $\perp$ if $\mathsf{KHPKE.ct}$ does not simultaneously satisfy the condition (5) and

$$\mathsf{KHPKE.}\pi' = [\mathbf{c}^\top \mathbf{u}_3]. \tag{6}$$

Otherwise, output $\mathsf{KHPKE.ct}_\mu / [\mathbf{c}^\top \mathbf{u}_0]$.

7.3 Security

In this section, we prove that Emura et al.'s KHPKE scheme Π_{KHPKE} [EHN+18] instantiated under the matrix DDH assumption satisfies the $\mathsf{KH\text{-}CCA}$ security.

Theorem 11. *Π_{KHPKE} satisfies the $\mathsf{KH\text{-}CCA}$ security under the matrix DDH assumption.*

We provide a simpler proof than the original paper [EHN+18]. Indeed, although Section 4.3 of [EHN+13], which is an ePrint version of [EHN+18], which discusses the $\mathsf{KH\text{-}CCA}$ security takes 15 pages long, Section 7.3 of this paper takes only 6 pages long. We want to claim that we do not provide an essential improvement on Emura et al.'s proof. We obtain a simpler proof by focusing on the matrix DDH assumption, while Emura et al. proved the $\mathsf{KH\text{-}CCA}$ security from a universal$_2$ hash proof system [CS02]. However, the refined proof enables us to understand the essence of a proof of our proposed ABKHE scheme in Section 8.

Although we already explained the intuition of a proof in Section 1.3.3, we provide a more detailed overview. We call $\mathcal{A}$'s decryption query on $\mathsf{KHPKE.ct} = (\mathsf{KHPKE.ct}_0 = [\mathbf{c}], \ldots)$ a *critical decryption query* if $\mathsf{KHPKE.ct}$ satisfies the conditions (5) and (6), $\mathsf{KHPKE.ct} \notin \mathcal{L}$ holds, and $\mathbf{c}$ does not live in the span of $\mathbf{A}$. Let $\mathsf{KHPKE.ct}^\star = (\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE.}\pi^\star, \mathsf{KHPKE.}\pi'^\star)$ denote a challenge ciphertext for a message $\mu_{\mathsf{coin}}^\star$, where $h^\star = H(\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE.}\pi'^\star)$. Let D denote the number of ciphertexts in $\mathcal{L}$ at the end of the game, where the challenge ciphertext $\mathsf{KHPKE.ct}^\star$ is the first ciphertext and $\mathcal{A}$ makes $D-1$ dependent evaluation queries. Let $\mathsf{KHPKE.ct}^{[d]} = (\mathsf{KHPKE.ct}_0^{[d]}, \mathsf{KHPKE.ct}_\mu^{[d]}, \mathsf{KHPKE.}\pi^{[d]}, \mathsf{KHPKE.}\pi'^{[d]})$ denote d-th ciphertext in $\mathcal{L}$ and treat it as an encryption of $\mu^{[d]}$, where $\mathsf{KHPKE.ct}^{[1]} = \mathsf{KHPKE.ct}^\star$ and $\mu^{[1]} = \mu_{\mathsf{coin}}^\star$.

We prove Theorem 11 by using a sequence of games $\mathsf{Game}_0, \mathsf{Game}_1, \mathsf{Game}_2, \mathsf{Game}_{3,1}, \mathsf{Game}_{4,1}, \mathsf{Game}_{5,1}, \mathsf{Game}_{3,2}, \ldots, \mathsf{Game}_{3,D}, \mathsf{Game}_{4,D}$, where it holds that $\mathsf{Game}_0 \approx_c \mathsf{Game}_1 = \mathsf{Game}_2 \approx_c \mathsf{Game}_{3,1}$ and $\mathsf{Game}_{5,d-1} \approx_c \mathsf{Game}_{3,d} \approx \mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{5,d}$ for $d \in [D]$. Observe that $\mathcal{A}$ which is given the challenge ciphertext $\mathsf{KHPKE.ct}^\star = (\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE.}\pi^\star, \mathsf{KHPKE.}\pi'^\star)$ can randomize it and compute $(\overline{\mathsf{KHPKE.ct}_0^\star}, \overline{\mathsf{KHPKE.ct}_\mu^\star}, \overline{\mathsf{KHPKE.}\pi'^\star})$ such that the decryption result is $\mu_{\mathsf{coin}}^\star$ by ignoring the condition (5) and $(\overline{\mathsf{KHPKE.ct}_0^\star}, \overline{\mathsf{KHPKE.ct}_\mu^\star}, \overline{\mathsf{KHPKE.}\pi'^\star}) \neq (\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE.}\pi'^\star)$ holds. If it holds that $H(\overline{\mathsf{KHPKE.ct}_0^\star}, \overline{\mathsf{KHPKE.ct}_\mu^\star}, \overline{\mathsf{KHPKE.}\pi'^\star}) = h^\star$, a decryption result of a ciphertext $(\overline{\mathsf{KHPKE.ct}_0^\star}, \overline{\mathsf{KHPKE.ct}_\mu^\star}, \mathsf{KHPKE.}\pi^\star, \overline{\mathsf{KHPKE.}\pi'^\star})$ is $\mu_{\mathsf{coin}}^\star$ without ignoring the condition (5) and $(\overline{\mathsf{KHPKE.ct}_0^\star}, \overline{\mathsf{KHPKE.ct}_\mu^\star}, \mathsf{KHPKE.}\pi^\star, \overline{\mathsf{KHPKE.}\pi'^\star}) \neq \mathsf{KHPKE.ct}^\star$ holds. Thus, $\mathcal{A}$ can break the $\mathsf{KH\text{-}CCA}$ security by making a decryption query on $(\overline{\mathsf{KHPKE.ct}_0^\star}, \overline{\mathsf{KHPKE.ct}_\mu^\star}, \mathsf{KHPKE.}\pi^\star, \overline{\mathsf{KHPKE.}\pi'^\star})$. In Game_1, we use the collision resistance of H and prevent the attack. In Game_2, we change how to compute $\mathsf{KHPKE.ct}^{[d]}$ for $d \in [2, D]$ so that the distribution of $\mathsf{KHPKE.ct}^{[d]}$ does not depend on $\mathsf{KHPKE.ct}^{[1]}, \ldots, \mathsf{KHPKE.ct}^{[d-1]}$. Since the change is conceptual, Game_1 and Game_2 follow the same distribution from $\mathcal{A}$'s view.

In Game_2, all ciphertexts $\mathsf{KHPKE.ct}^{[1]} = \mathsf{KHPKE.ct}^\star, \ldots, \mathsf{KHPKE.ct}^{[D]} \in \mathcal{L}$ depend on $\mu_{\mathsf{coin}}^\star$. In $\mathsf{Game}_{3,d}, \mathsf{Game}_{4,d}, \mathsf{Game}_{5,d}$ for $d \in [D]$, we change distributions of ciphertexts $\mathsf{KHPKE.ct}^{[1]}, \ldots, \mathsf{KHPKE.ct}^{[D]}$ so that all the ciphertexts $\mathsf{KHPKE.ct}^{[1]}, \ldots, \mathsf{KHPKE.ct}^{[D]}$ are independent of $\mu_{\mathsf{coin}}^\star$. We can complete the change by following security proofs of $\mathsf{CCA1}$-secure Cramer-Shoup-lite and the $\mathsf{CCA2}$-secure Cramer-Shoup cryptosystem [CS98].

Proof of Theorem 11. We use the following sequence of games.

Game_0. This is the $\mathsf{KH\text{-}CCA}$ security game. Hereafter, let $\mathsf{KHPKE.ct}^\star = (\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE.}\pi^\star, \mathsf{KHPKE.}\pi'^\star)$ denote a challenge ciphertext for a message $\mu_{\mathsf{coin}}^\star$, where $h^\star = H(\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE.}\pi'^\star)$.

Game_1. This is the same as Game_0 except that a collision does not occur for a hash function H among all ciphertexts that appeared in the security game.

The collision resistance of H ensures that $\mathsf{Game}_0 \approx_c \mathsf{Game}_1$ holds.

Game_2. This is the same as Game_1 except the answers to dependent evaluation queries so that the distribution of ciphertexts $\mathsf{KHPKE.ct}^{[1]} = \mathsf{KHPKE.ct}^\star, \ldots, \mathsf{KHPKE.ct}^{[D]} \in \mathcal{L}$ are independent. In Game_1, $\mathcal{C}$ runs Eval algorithm with inputs $\mathsf{KHPKE.ct}^{[1]}, \ldots, \mathsf{KHPKE.ct}^{[d-1]}$ that are answers to $\mathcal{A}$'s challenge query and dependent evaluation queries, and creates an evaluated ciphertext $\mathsf{KHPKE.ct}^{[d]}$. In Game_2, upon $\mathcal{A}$'s challenge query, $\mathcal{C}$ runs $\mathsf{KHPKE.Enc}$ algorithm and creates two ciphertexts $\mathsf{KHPKE.ct}^\star$ and $\widetilde{\mathsf{KHPKE.ct}}^\star$ in the same way as in Game_1, sends $\widetilde{\mathsf{KHPKE.ct}}^\star$ to $\mathcal{A}$ as the challenge ciphertext, and stores both ciphertexts $(\mathsf{KHPKE.ct}^\star, \widetilde{\mathsf{KHPKE.ct}}^\star) \in \mathcal{L}$. Upon $\mathcal{A}$'s first dependent evaluation query, $\mathcal{C}$ runs $\mathsf{KHPKE.Eval}$ algorithm with inputs $\widetilde{\mathsf{KHPKE.ct}}^{[1]}$ in place of $\mathsf{KHPKE.ct}^{[1]}$ that is the answer to $\mathcal{A}$'s challenge query and creates two evaluated ciphertexts $\mathsf{KHPKE.ct}^{[2]}$ and $\widetilde{\mathsf{KHPKE.ct}}^{[2]}$ in the same way as in Game_1, sends $\widetilde{\mathsf{KHPKE.ct}}^{[2]}$ to $\mathcal{A}$ as the answer to the evaluation query, and stores both ciphertexts $(\mathsf{KHPKE.ct}^{[2]}, \widetilde{\mathsf{KHPKE.ct}}^{[2]}) \in \mathcal{L}$. In the same way, upon $\mathcal{A}$'s $(d-1)$-th dependent evaluation query, $\mathcal{C}$ runs Eval algorithm with inputs $\widetilde{\mathsf{KHPKE.ct}}^{[1]}, \ldots, \widetilde{\mathsf{KHPKE.ct}}^{[d-1]}$ in place of $\mathsf{KHPKE.ct}^{[1]}, \ldots, \mathsf{KHPKE.ct}^{[d-1]}$ that are the answers to $\mathcal{A}$'s challenge query and dependent evaluation queries, and creates two evaluated ciphertexts $\mathsf{KHPKE.ct}^{[d]}$ and $\widetilde{\mathsf{KHPKE.ct}}^{[d]}$ in the same way as in Game_1, sends $\widetilde{\mathsf{KHPKE.ct}}^{[d]}$ to $\mathcal{A}$ as the answer to the evaluation query, and stores both ciphertexts $(\mathsf{KHPKE.ct}^{[d]}, \widetilde{\mathsf{KHPKE.ct}}^{[d]}) \in \mathcal{L}$. In Game_1 and Game_2, all ciphertexts $\mathsf{KHPKE.ct}^{[d]}$ and $\widetilde{\mathsf{KHPKE.ct}}^{[d]}$ follow the same distribution for $d \in [D]$.

From now on, we change a distribution of d-th ciphertext $\mathsf{KHPKE.ct}^{[d]} = (\cdots, \mathsf{KHPKE.ct}_\mu^{[d]}, \cdots) \in \mathcal{L}$ for $d \in [D]$ one by one so that $\mathsf{KHPKE.ct}_\mu^{[d]}$ is independent of the other elements of $\mathsf{KHPKE.ct}^{[d]}$ and distributed uniformly at random over $\mathbb{G}$. For this purpose, we use the following sequence of games $\mathsf{Game}_{3,d}, \mathsf{Game}_{4,d}, \mathsf{Game}_{5,d}$ for $d \in [D]$, where $\mathsf{Game}_{5,0} = \mathsf{Game}_2$ and the proof terminates in $\mathsf{Game}_{4,D}$.

$\mathsf{Game}_{3,d}$. This is the same as $\mathsf{Game}_{5,d-1}$ except $\mathcal{C}$'s answer to the challenge query if $d = 1$ and a dependent evaluation query if $d \in [2, D]$. If $d = 1$, $\mathcal{C}$ creates the challenge ciphertext $\mathsf{KHPKE.ct}^\star = (\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE.}\pi^\star, \mathsf{KHPKE.}\pi'^\star)$;

$$\begin{aligned}&\mathsf{KHPKE.ct}_0^\star = [\mathbf{c}], \qquad \mathsf{KHPKE.ct}_\mu^\star = \mu_{\mathsf{coin}}^\star \cdot [\mathbf{c}^\top \mathbf{u}_0] \qquad \mathsf{KHPKE.}\pi^\star = [\mathbf{c}^\top(\mathbf{u}_1 + h^\star \cdot \mathbf{u}_2)], \\ &\mathsf{KHPKE.}\pi'^\star = [\mathbf{c}^\top \mathbf{u}_3],\end{aligned} \tag{7}$$

where $\mathbf{c} \leftarrow_R \mathbb{Z}_p^{k+1}$ and $h^\star = H(\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE.}\pi'^\star)$. If $d \in [2, D]$, $\mathcal{C}$ creates $\mathsf{KHPKE.ct}^{(0)}$ to compute $\mathsf{KHPKE.ct}^{[d]}$ in the same way as (7) except that $\mu_{\mathsf{coin}}^\star$ is replaced with $1_{\mathbb{G}}$. We note that $\mathcal{C}$ creates $\widetilde{\mathsf{KHPKE.ct}}^{[1]}, \ldots, \widetilde{\mathsf{KHPKE.ct}}^{[D]}$ in the same way as in Game_2.

We can prove $\mathsf{Game}_{5,d-1} \approx_c \mathsf{Game}_{3,d}$ under the matrix DDH assumption.

Lemma 15 ($\mathsf{Game}_{5,d-1} \approx_c \mathsf{Game}_{3,d}$)**.** *If the matrix DDH assumption holds,* $\mathsf{Game}_{5,d-1}$ *and* $\mathsf{Game}_{3,d}$ *are computationally indistinguishable for any PPT* $\mathcal{A}$.

Proof of Lemma 15. We show that for any PPT adversary $\mathcal{A}$ that breaks the $\mathsf{KH\text{-}CCA}$ security of Π_{KHPKE}, there exists a reduction algorithm $\mathcal{B}_1$ that solves the matrix DDH assumption, where

$$|\Pr[E_{5,d-1}] - \Pr[E_{3,d}]| \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{mDDH}_{\mathbb{G}}}(\lambda). \tag{8}$$

We prove only for $d = 1$ since proofs for the other cases are essentially the same. $\mathcal{B}_1$ receives $(\mathcal{G}(1^\lambda), [\mathbf{A}], [\mathbf{v}])$ which is an instance of the matrix DDH problem, where $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$, $\mathbf{v} = \mathbf{A}\mathbf{s}$ for $\mathbf{s} \leftarrow_R \mathbb{Z}_p^k$ or $\mathbf{v} \leftarrow_R \mathbb{Z}_p^{k+1}$. $\mathcal{B}_1$ chooses a collision-resistant hash function $H \leftarrow_R \mathcal{H}$, samples random vectors $(\mathbf{u}_\iota)_{\iota \in [0,3]} \leftarrow_R \mathbb{Z}_p^{k+1}$, then sends $\mathsf{KHPKE.pk} = (\widehat{\mathcal{G}}(1^\lambda), [\mathbf{A}], ([\mathbf{A}^\top \mathbf{u}_\iota])_{\iota \in [0,3]}, H)$ to $\mathcal{A}$. Since $\mathcal{B}_1$ knows $(\mathbf{u}_\iota)_{\iota \in [0,3]}$, it can answer all $\mathcal{A}$'s homomorphic evaluation key reveal query, decryption queries, and evaluation queries.

Upon $\mathcal{A}$'s challenge query on $(\mu_0^\star, \mu_1^\star)$, $\mathcal{B}_1$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$ and creates the challenge ciphertext $\mathsf{KHPKE.ct}^\star = (\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE}.\pi^\star, \mathsf{KHPKE}.\pi'^\star)$;

$$\begin{aligned}&\mathsf{KHPKE.ct}_0^\star = [\mathbf{v}], \qquad \mathsf{KHPKE.ct}_\mu^\star = \mu_{\mathsf{coin}}^\star \cdot [\mathbf{v}^\top \mathbf{u}_0], \qquad \mathsf{KHPKE}.\pi^\star = [\mathbf{v}^\top(\mathbf{u}_1 + h^\star \cdot \mathbf{u}_2)], \\ &\mathsf{KHPKE}.\pi'^\star = [\mathbf{v}^\top \mathbf{u}_3],\end{aligned} \tag{9}$$

where $h^\star = H(\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE}.\pi'^\star)$. The challenge ciphertext $\mathsf{KHPKE.ct}^\star$ is distributed as in $\mathsf{Game}_{5,0}$ (resp. $\mathsf{Game}_{3,1}$) if $\mathbf{v} = \mathbf{A}\mathbf{s}$ (resp. $\mathbf{v} \leftarrow_R \mathbb{Z}_p^{k+1}$). Thus, the inequality (8) holds. $\square$

$\mathsf{Game}_{4,d}$. This is the same as $\mathsf{Game}_{3,d}$ except $\mathcal{C}$'s answer to the challenge query if $d = 1$ and a $(d-1)$-th dependent evaluation query if $d \in [2, D]$ by setting $\mathsf{KHPKE.ct}_\mu^{[d]} \leftarrow_R \mathbb{G}$. Since the d-th ciphertext $\mathsf{KHPKE.ct}^{[d]} \in \mathcal{L}$ becomes independent of $\mu_{\mathsf{coin}}^\star$ in $\mathsf{Game}_{4,d}$, $\mathcal{A}$'s advantage in $\mathsf{Game}_{4,D}$ is exactly 0.

Lemma 16 ($\mathsf{Game}_{3,d} \approx \mathsf{Game}_{4,d}$)**.** *It holds that*

$$\Pr[E_{3,d}] = \Pr[E_{4,d}]$$

with overwhelming probability.

We will prove Lemma 16 at the end of the proof.

$\mathsf{Game}_{5,d}$. This is the same as $\mathsf{Game}_{4,d}$ except $\mathcal{C}$'s answer to the challenge query if $d = 1$ and a dependent evaluation query if $d \in [2, D]$. If $d = 1$, $\mathcal{C}$ creates the challenge ciphertext $\mathsf{KHPKE.ct}^\star = (\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE}.\pi^\star, \mathsf{KHPKE}.\pi'^\star)$ in the same way as the real scheme except that $\mathsf{KHPKE.ct}_\mu^\star \leftarrow_R \mathbb{G}$ is unchanged. If $d \in [2, D]$, $\mathcal{C}$ creates $\mathsf{KHPKE.ct}^{(0)} = (\mathsf{KHPKE.ct}_0^{(0)}, \mathsf{KHPKE.ct}_\mu^{(0)}, \mathsf{KHPKE}.\pi^{(0)}, \mathsf{KHPKE}.\pi'^{(0)})$ to compute $\mathsf{KHPKE.ct}^{[d]}$ in the same way as the real scheme except that $\mathsf{KHPKE.ct}_\mu^{(0)} \leftarrow_R \mathbb{G}$ is unchanged.

We can prove $\mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{5,d}$ under the matrix DDH assumption.

Lemma 17 ($\mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{5,d}$)**.** *If the matrix DDH assumption holds,* $\mathsf{Game}_{4,d}$ *and* $\mathsf{Game}_{5,d}$ *are computationally indistinguishable for any PPT* $\mathcal{A}$*.*

We can prove Lemma 17 essentially in the same way as Lemma 15. For example, the only difference for $d = 1$ is that the reduction algorithm creates the challenge ciphertext in the same way as (9) except $\mathsf{KHPKE.ct}_\mu^\star \leftarrow_R \mathbb{G}$ if $d = 1$. Then, the reduction algorithm simulates $\mathsf{Game}_{4,d}$ if $\mathbf{v} \leftarrow_R \mathbb{Z}_p^{k+1}$ and $\mathsf{Game}_{5,d}$ if $\mathbf{v} = \mathbf{A}\mathbf{s}$.

To conclude the proof of Theorem 11, we prove Lemma 16.

Proof of Lemma 16. We prove only for $d = 1$ since proofs for the other cases are essentially the same. For this purpose, we show that even when $\mathcal{A}$ is computationally unbounded, $\mathsf{Game}_{3,d} \equiv \mathsf{Game}_{4,d}$ holds with overwhelming probability. For this purpose, we construct a simulator that behaves as $\mathcal{C}$ in $\mathsf{Game}_{3,d}$ from $\mathcal{A}$'s view. The simulator runs $(p, \mathbb{G}, g) \leftarrow \widehat{\mathcal{G}}(1^\lambda)$ and chooses a collision-resistant hash function $H \leftarrow_R \mathcal{H}$. The simulator samples $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$, random vectors $\widehat{\mathbf{u}}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \leftarrow_R \mathbb{Z}_p^{k+1}$, and random $\tilde{\alpha}_0 \leftarrow_R \mathbb{Z}_p$, then sets $\mathbf{u}_0 = \widehat{\mathbf{u}}_0 + \tilde{\alpha}_0 \mathbf{a}^\perp$. Nevertheless, the simulator does not use $\mathbf{u}_0$ but $\widehat{\mathbf{u}}_0$ to simulate the game except for creating $\mathsf{KHPKE.ct}^{[d]} \in \mathcal{L}$. At first, the simulator sends $\mathsf{KHPKE.pk} = (\widehat{\mathcal{G}}(1^\lambda), [\mathbf{A}], [\mathbf{A}^\top \widehat{\mathbf{u}}_0], [\mathbf{A}^\top \mathbf{u}_1], [\mathbf{A}^\top \mathbf{u}_2], [\mathbf{A}^\top \mathbf{u}_3], H)$ to $\mathcal{A}$. $\mathsf{KHPKE.pk}$ is properly distributed since it holds that

$$[\mathbf{A}^\top \widehat{\mathbf{u}}_0] = [\mathbf{A}^\top (\mathbf{u}_0 - \tilde{\alpha}_0 \mathbf{a}^\perp)] = [\mathbf{A}^\top \mathbf{u}_0] \cdot [\mathbf{A}^\top \mathbf{a}^\perp]^{-\tilde{\alpha}_0} = [\mathbf{A}^\top \mathbf{u}_0]. \tag{10}$$

The simulator answers $\mathcal{A}$'s homomorphic evaluation key reveal query and evaluation queries by using $\mathbf{u}_1, \mathbf{u}_2$ as in $\mathsf{Game}_{3,d}$, while it answers $\mathcal{A}$'s decryption queries by using $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ and $\widehat{\mathbf{u}}_0$. We will discuss the validity later.

Upon $\mathcal{A}$'s challenge query on $(\mu_0^\star, \mu_1^\star)$, the simulator samples $\mathsf{coin} \leftarrow_R \{0, 1\}$ and creates the challenge ciphertext $\mathsf{KHPKE.ct}^\star = (\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE}.\pi^\star, \mathsf{KHPKE}.\pi'^\star)$ in the same way as in $\mathsf{Game}_{3,d}$;

$$\mathsf{KHPKE.ct}_0^\star = [\mathbf{c}], \qquad \mathsf{KHPKE.ct}_\mu^\star = \mu_{\mathsf{coin}}^\star \cdot [\mathbf{c}^\top \mathbf{u}_0], \qquad \mathsf{KHPKE}.\pi^\star = [\mathbf{c}^\top (\mathbf{u}_1 + h^\star \cdot \mathbf{u}_2)],$$
$$\mathsf{KHPKE}.\pi'^\star = [\mathbf{c}^\top \mathbf{u}_3],$$

where $h^\star = H(\mathsf{KHPKE.ct}_0^\star, \mathsf{KHPKE.ct}_\mu^\star, \mathsf{KHPKE}.\pi'^\star)$. Observe that $\mathsf{KHPKE.ct}_\mu^\star$ is the only element that the simulator uses $\mathbf{u}_0$ to create and

$$\mathsf{KHPKE.ct}_\mu^\star = \mu_{\mathsf{coin}}^\star \cdot [\mathbf{c}^\top (\widehat{\mathbf{u}}_0 + \tilde{\alpha}_0 \mathbf{a}^\perp)] = \mu_{\mathsf{coin}}^\star \cdot [\mathbf{c}^\top \widehat{\mathbf{u}}_0] \cdot [\mathbf{c}^\top \mathbf{a}^\perp]^{\tilde{\alpha}_0}$$

holds. Since $[\mathbf{c}^\top \mathbf{a}^\perp]$ is a generator of $\mathbb{G}$ with overwhelming probability and $\mathsf{KHPKE.ct}_\mu^\star$ is the only element which depends on $\tilde{\alpha}_0$ in the security game, $\mathsf{KHPKE.ct}_\mu^\star$ is distributed uniformly at random over $\mathbb{G}$ as in $\mathsf{Game}_{4,d}$.

Finally, we check that the simulator's answers to decryption queries are valid although $\widehat{\mathbf{u}}_0 \neq \mathbf{u}_0$ is used. For this purpose, we divide $\mathcal{A}$'s attack strategies into two types called Type-1 and Type-2 which are defined as follows:

- $\mathcal{A}$ is called Type-1 if it makes a homomorphic evaluation key reveal query in Phase 1.
- $\mathcal{A}$ is called Type-2 if it does not make a homomorphic evaluation key reveal query in Phase 1.

By definition, Type-1 and Type-2 are mutually exclusive and cover all possible strategies of $\mathcal{A}$. We show that the simulator's answers against $\mathcal{A}$ of Type-1 (resp. Type-2) are valid by following the proof of the CCA1-secure Cramer-Shoup-lite (resp. CCA2-secure Cramer-Shoup cryptosystem) [CS98].

Case of Type-1. Since $\mathcal{A}$ of Type-1 makes a homomorphic evaluation key reveal query in Phase 1, it is allowed to make decryption queries only in Phase 1. Upon $\mathcal{A}$'s decryption query on $\mathsf{KHPKE.ct} = (\mathsf{KHPKE.ct}_0 = [\mathbf{c}'], \mathsf{KHPKE.ct}_\mu, \mathsf{KHPKE}.\pi, \mathsf{KHPKE}.\pi')$, the simulator's answer is valid when $\mathbf{c}'^\top \mathbf{u}_0 = \mathbf{c}'^\top \widehat{\mathbf{u}}_0$ holds. Thus, the answer is invalid when $\mathbf{c}'$ does not live in the span of $\mathbf{A}$ and the answer is not $\perp$. In other words, the simulator cannot answer $\mathcal{A}$'s critical decryption queries validly. When the computationally unbounded $\mathcal{A}$ receives $\mathsf{KHPKE.pk}$, it can compute $\widehat{\mathbf{u}}_3$ such that $\mathbf{A}^\top \mathbf{u}_3 = \mathbf{A}^\top \widehat{\mathbf{u}}_3$, where $\mathbf{u}_3 = \widehat{\mathbf{u}}_3 + \tilde{\alpha}_3 \mathbf{a}^\perp$. If the answer to $\mathcal{A}$'s decryption query is not $\perp$, $\mathsf{KHPKE}.\pi' = [\mathbf{c}'^\top \mathbf{u}_3]$ holds due to the condition (6). If $\mathbf{c}'$ does not live in the span of $\mathbf{A}$, a computationally unbounded $\mathcal{A}$'s ability to make a critical decryption query is equivalent to the knowledge of $\tilde{\alpha}_3 \in \mathbb{Z}_p$. Although $\mathcal{A}$ of Type-1 can learn $\tilde{\alpha}_3$ when it receives the challenge ciphertext $\mathsf{KHPKE.ct}^\star$, it is not allowed to make decryption queries in Phase 2. The only way for $\mathcal{A}$ to learn $\tilde{\alpha}_3$ is making decryption queries in Phase 1 such that $\mathbf{c}'$ does not live in the span of $\mathbf{A}$. Although $\mathcal{A}$ can eliminate a candidate of $\tilde{\alpha}_3 \in \mathbb{Z}_p$ by making a decryption query and the answer is $\perp$, there are exponentially many candidates and $\mathcal{A}$ is allowed to make only a polynomial number of queries. Thus, the simulator's answers to decryption queries are valid with probability $1 - Q_{\mathsf{Dec}}/q$, where Q_{Dec} denotes the number of $\mathcal{A}$'s decryption queries.

Case of Type-2. Since $\mathcal{A}$ of Type-2 does not make a homomorphic evaluation key reveal query in Phase 1, it is allowed to make decryption queries until it makes a homomorphic evaluation key reveal query in Phase 2. When the computationally unbounded $\mathcal{A}$ receives $\mathsf{KHPKE.pk}$, it can compute $\widehat{\mathbf{u}}_\iota$ for $\iota \in [2]$ such that $\mathbf{A}^\top \mathbf{u}_\iota = \mathbf{A}^\top \widehat{\mathbf{u}}_\iota$, where $\mathbf{u}_\iota = \widehat{\mathbf{u}}_\iota + \tilde{\alpha}_\iota \mathbf{a}^\perp$. When the computationally unbounded $\mathcal{A}$ receives the challenge ciphertext $\mathsf{KHPKE.ct}^\star$, it learns the value of $\tilde{\alpha}_1 + h^\star \cdot \tilde{\alpha}_2$ since it holds that

$$\mathsf{KHPKE}.\pi^\star = [\mathbf{c}^\top(\widehat{\mathbf{u}}_1 + \tilde{\alpha}_1 \mathbf{a}^\perp) + h^\star \cdot (\widehat{\mathbf{u}}_2 + \tilde{\alpha}_2 \mathbf{a}^\perp)] = [\mathbf{c}^\top \widehat{\mathbf{u}}_1 + h^\star \cdot \widehat{\mathbf{u}}_2] \cdot [\mathbf{c}^\top \mathbf{a}^\perp]^{\tilde{\alpha}_1 + h^\star \cdot \tilde{\alpha}_2}.$$

If the answer to $\mathcal{A}$'s decryption query is not $\perp$, $\mathsf{KHPKE}.\pi = [\mathbf{c}'^\top(\mathbf{u}_1 + h \cdot \mathbf{u}_2)]$ holds due to the condition (5). If $\mathbf{c}'$ does not live in the span of $\mathbf{A}$, $\mathcal{A}$ learns the value of $\tilde{\alpha}_1 + h \cdot \tilde{\alpha}_2$, where the change in Game_1 ensures that $h \neq h^\star$ holds. Then, a computationally unbounded $\mathcal{A}$'s ability to make a critical decryption query is equivalent to the knowledge of $(\tilde{\alpha}_1, \tilde{\alpha}_2) \in \mathbb{Z}_p^2$. $\mathcal{A}$ cannot learn $\tilde{\alpha}_1 + h \cdot \tilde{\alpha}_2$ for any h from answers to dependent evaluation queries since the change in Game_2 ensures that the discrete logarithm of $\mathsf{KHPKE.ct}_0^{[d]}$ lives in the span of $\mathbf{A}$. (If $d \in [2, D]$, the change in $\mathsf{Game}_{5,d-1}$ is also required to ensure the fact.) Although $\mathcal{A}$ of Type-2 can learn α_1, α_2 when it makes a homomorphic evaluation key reveal query in Phase 2, it is not allowed to make decryption queries after the query. The only way for $\mathcal{A}$ to learn $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ is making decryption queries and evaluation queries such that $\mathbf{c}'$ does not live in the span of $\mathbf{A}$. Although $\mathcal{A}$ can eliminate a candidate of $\tilde{\alpha}_1 + h \cdot \tilde{\alpha}_2$ for some h by making a decryption query or an evaluation query and the answer is $\perp$, there are exponentially many candidates and $\mathcal{A}$ is allowed to make only polynomial number of queries. Thus, the simulator's answers to decryption queries are valid with probability $1 - (Q_{\mathsf{Dec}} + Q_{\mathsf{Eval}})/q$, where Q_{Dec} (resp. Q_{Eval}) denotes the number of $\mathbf{A}$'s decryption (resp. evaluation) queries. ☐

☐

8 Pairing-based Construction of ABKHE

In this section, we propose a pairing-based ABKHE scheme Π_{ABKHE} from a pair encoding scheme (PES) by combining with an ABE schemes over dual system groups Π_{DSG} [AC16, AC17, CGW15] and Emura et al.'s KHPKE scheme Π_{KHPKE}. In Section 8.1, we review bilinear groups and the PES. In Section 8.2, we provide a construction of Π_{ABKHE}. In Section 8.3, we prove the adaptive KH-CCA security.

8.1 Preliminaries on Pairing-based Cryptography

8.1.1 Bilinear Groups

Let $\mathcal{G}$ be a bilinear group generator which takes the security parameter 1^λ as input, and outputs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$, where p is a $\Theta(\lambda)$-bit prime number, $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic groups of order p, g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map. For simplicity, let $\mathcal{G}(1^\lambda) := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ denote the output of $\mathcal{G}(1^\lambda)$. Let 1_T denote an identity element of $\mathbb{G}_T$. As in Section 7.1, we use the notations $[\mathbf{A}]_1, [\mathbf{A}]_2$, and $[\mathbf{A}]_T$ for $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$, respectively. For matrices $\mathbf{A}$ and $\mathbf{B}$ of compatible dimensions, let $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{A}^\top \mathbf{B}]_T$.

For a matrix distribution $\mathcal{D}_k$ which we explained in Section 7.1, we use the following property.

Lemma 18 (Basis Lemma [CGW15]). *For $(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, $\mathbf{a}^\perp$ does not live in the span of $\mathbf{B}$, $\mathbf{b}^\perp$ does not live in the span of $\mathbf{A}$, and ${\mathbf{a}^\perp}^\top \mathbf{b} \neq \mathbf{0}$ simultaneously hold with probability $1 - 1/p$.*

We use the following complexity assumptions to prove the adaptive KH-CCA security of the proposed ABKHE scheme.

Definition 25 (m-fold Matrix DDH Assumption). *For bilinear groups $\mathcal{G}(1^\lambda) = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ and a polynomially bounded m, an advantage for solving the m-fold matrix DDH problem over $\mathbb{G}_1$ by an algorithm $\mathcal{A}$ is defined to be*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{mDDH}_{\mathbb{G}_1}}(\lambda) := \left| \Pr\left[\mathcal{A}(\mathcal{G}(1^\lambda), [\mathbf{A}]_1, [\mathbf{AS}]_1) \to 1\right] - \Pr\left[\mathcal{A}(\mathcal{G}(1^\lambda), [\mathbf{A}]_1, [\mathbf{V}]_1) \to 1\right] \right|,$$

where $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$, $\mathbf{S} \leftarrow_R \mathbb{Z}_p^{k \times m}$, and $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1) \times m}$. We say that the m-fold matrix DDH assumption over $\mathbb{G}_1$ holds if it is negligible for all PPT $\mathcal{A}$. We also define the m-fold matrix DDH assumption over $\mathbb{G}_2$.

Remark 8. *A 1-fold matrix DDH assumption is the matrix DDH assumption as in Definition 24. For a polynomially bounded m, the m-fold matrix DDH assumption is computationally equivalent to the matrix DDH assumption [AC16, EHK+17].*

Definition 26 ((d_1, d_2)-q-ratio Assumption [AC17]). *For bilinear groups $\mathcal{G}(1^\lambda) = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$, let*

$$\mathcal{D}_1 := ([u_i]_1)_{i \in [0, d_2]} \cup \left\{ \left[\frac{u_i}{u_j v_k} \right]_1 \right\}_{i,j \in [d_2], i \neq j, k \in [d_1]}, \quad \mathcal{D}_2 := ([v_i]_2)_{i \in [d_1]} \cup \left\{ \left[\frac{v_i}{v_j u_k} \right]_2 \right\}_{i,j \in [d_1], i \neq j, k \in [d_2]},$$

where $u_0, u_1, \ldots, u_{d_2}, v_1, \ldots, v_{d_1} \leftarrow_R \mathbb{Z}_p^*$. An advantage for solving the (d_1, d_2)-q-ratio problem by an algorithm $\mathcal{A}$ is defined to be

$$\mathsf{Adv}_{\mathcal{A}}^{(d_1,d_2)\text{-}q\text{-ratio}}(\lambda) := \left| \Pr\left[\mathcal{A}(\mathcal{G}(1^\lambda), \mathcal{D}_1, \mathcal{D}_2, [1/u_0]_2) \to 1\right] - \Pr\left[\mathcal{A}(\mathcal{G}(1^\lambda), \mathcal{D}_1, \mathcal{D}_2, [u']_2) \to 1\right] \right|,$$

where $u' \leftarrow_R \mathbb{Z}_p$. We say that the (d_1, d_2)-q-ratio assumption holds if it is negligible for all PPT $\mathcal{A}$.

8.1.2 Pair Encoding Scheme

We review a pair encoding scheme (PES) by following [AC16, AC17, Att14, Tak21]. A PES for a predicate $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ consists of the following four polynomial time algorithms $(\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ defined as follows.

$\mathsf{Param}(\mathsf{par}) \to n$. On input par, Param outputs $n \in \mathbb{Z}_p$ that specifies the number of common variables denoted by $\mathbf{b} \coloneqq (b_1, \ldots, b_n)$.

$\mathsf{EncC}(x, p) \to (w_1, w_2, \mathbf{c})$. On input $x \in \mathcal{X}$ and p, EncC outputs a vector of w_3 ciphertext-encoding polynomials $\mathbf{c} = (c_1, \ldots, c_{w_3})$ in non-lone ciphertext-encoding variables s_0 and $\mathbf{s} = (s_1, s_1, \ldots, s_{w_1})$ and lone ciphertext-encoding variables $\hat{\mathbf{s}} = (\hat{s}_1, \ldots, \hat{s}_{w_2})$. The t-th polynomial is given by

$$c_t \coloneqq \sum_{i \in [w_2]} \eta_{t,i} \hat{s}_i + \sum_{i \in [0, w_1], j \in [n]} \eta_{t,i,j} s_i b_j$$

for $t \in [w_3]$, where $\eta_{t,i}, \eta_{t,i,j} \in \mathbb{Z}_p$.

$\mathsf{EncK}(y, p) \to (m_1, m_2, \mathbf{k})$. On input $y \in \mathcal{Y}$ and p, EncK outputs a vector of m_3 key-encoding polynomials $\mathbf{k} = (k_1, \ldots, k_{m_3})$ in non-lone key-encoding variables $\mathbf{r} = (r_1, \ldots, r_{m_1})$ and lone key-encoding variables α and $\hat{\mathbf{r}} = (\hat{r}_1, \ldots, \hat{r}_{m_2})$. The t'-th polynomial is given by

$$k_{t'} \coloneqq \phi_{t'} \alpha + \sum_{i' \in [m_2]} \phi_{t',i'} \hat{r}_{i'} + \sum_{i' \in [m_1], j \in [n]} \phi_{t',i',j} r_{i'} b_j$$

for $t' \in [m_3]$, where $\phi_{t'}, \phi_{t',i'}, \phi_{t',i',j} \in \mathbb{Z}_p$.

$\mathsf{Pair}(x, y, p) \to (\mathbf{E}, \overline{\mathbf{E}})$. On input $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and p, Pair outputs two matrices $\mathbf{E}$ and $\overline{\mathbf{E}}$ of size $(w_1 + 1) \times m_3$ and $w_3 \times m_1$, respectively.

Definition 27. $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ *for a predicate f is correct if for all (p, par), $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $f(x, y) = 1$, it holds that*

$$\mathbf{s}^\top \mathbf{E} \mathbf{k} - \mathbf{c}^\top \overline{\mathbf{E}} \mathbf{r} = \sum_{i \in [0, w_1], t' \in [m_3]} E_{i,t'} s_i k_{t'} - \sum_{t \in [w_3], i' \in [m_1]} \overline{E}_{t,i'} c_t r_{i'} = \alpha s_0,$$

where $E_{i,t'}$ denote a (i, t')-th element of $\mathbf{E}$ and $\overline{E}_{t,i'}$ denote a (t, i')-th element of $\overline{\mathbf{E}}$.

Remark 9. *For example, a PES for IBE has two common variables (b_1, b_2), one ciphertext-encoding polynomial $c = s(b_1 + \mathsf{id} \cdot b_2)$ and one key-encoding polynomial $k = \alpha + r(b_1 + \mathsf{id} \cdot b_2)$. The scheme is correct since it holds that $sk - cr = \alpha s$.*

We review the definitions of the perfect security [Att14] and the symbolic security [AC17]. Intuitively, the perfect security ensures that given non-lone variables $s_0, \mathbf{s}, \mathbf{r}$, ciphertext-encoding polynomials $\mathbf{c} = (c_1, \ldots, c_{w_3})$, and key-encoding polynomials $\mathbf{k} = (k_1, \ldots, k_{m_3})$, the distributions do not change regardless of the value of α.

Definition 28 (Perfect Security [Att14])**.** *A $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ for a predicate $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ satisfies the perfect security if for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $f(x, y) = 0$, it holds that*

$$\begin{aligned} &\begin{pmatrix} s_0, \mathbf{s}, \mathbf{r} \\ (\sum_{i \in [w_2]} \eta_{t,i} \hat{s}_i + \sum_{i \in [0, w_1], j \in [n]} \eta_{t,i,j} s_i b_j)_{t \in [w_3]} \\ (\sum_{i' \in [m_2]} \phi_{t',i'} \hat{r}_{i'} + \sum_{i' \in [m_1], j \in [n]} \phi_{t',i',j} r_{i'} b_j)_{t' \in [m_3]} \end{pmatrix} \\ \equiv &\begin{pmatrix} s_0, \mathbf{s}, \mathbf{r} \\ (\sum_{i \in [w_2]} \eta_{t,i} \hat{s}_i + \sum_{i \in [0, w_1], j \in [n]} \eta_{t,i,j} s_i b_j)_{t \in [w_3]} \\ (\phi_{t'} \alpha + \sum_{i' \in [m_2]} \phi_{t',i'} \hat{r}_{i'} + \sum_{i' \in [m_1], j \in [n]} \phi_{t',i',j} r_{i'} b_j)_{t' \in [m_3]} \end{pmatrix} \end{aligned} \tag{11}$$

where $s_0 \leftarrow_R \mathbb{Z}_p$, $\mathbf{s} \leftarrow_R \mathbb{Z}_p^{w_1}$, $\mathbf{r} \leftarrow_R \mathbb{Z}_p^{m_1}$, $\hat{\mathbf{s}} \leftarrow_R \mathbb{Z}_p^{w_2}$, $\hat{\mathbf{r}} \leftarrow_R \mathbb{Z}_p^{m_2}$, $\mathbf{b} \leftarrow_R \mathbb{Z}_p^n$,$\alpha \leftarrow_R \mathbb{Z}_p$, and the boxed part denote a change between the left and the right distribution.

Theorem 12 ([AC16, CG17, CGW15, Tak21]). *If there is a* $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ *for a predicate f satisfying the perfect security, there is an adaptively secure* ABE *scheme for the same predicate f under the standard matrix DDH assumption over $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Next, we describe the symbolic security which captures more expressive predicates f than the perfect security.

Definition 29 (Symbolic Security [AC17]). *A* $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ *for a predicate $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ satisfies (d_1, d_2)-selective symbolic security for positive integers d_1 and d_2 if for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $f(x, y) = 0$, there exist three deterministic polynomial-time algorithms* EncB, EncS, *and* EncR*;*

$\mathsf{EncB}(x) \to (\mathbf{B}_1, \ldots, \mathbf{B}_n) \in (\mathbb{Z}_p^{d_1 \times d_2})^n$

$\mathsf{EncR}(x, y) \to (\mathbf{r}_1, \ldots, \mathbf{r}_{m_1}, \mathbf{a}, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2}) \in (\mathbb{Z}_p^{d_1})^{m_1} \times (\mathbb{Z}_p^{d_2})^{m_2+1}$

$\mathsf{EncS}(x) \to (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{w_1}, \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2}) \in (\mathbb{Z}_p^{d_2})^{w_1+1} \times (\mathbb{Z}_p^{d_1})^{w_2}$

such that $\langle \mathbf{s}_0, \mathbf{a} \rangle \neq 0$, and if we substitute

$$s_i : \mathbf{s}_i^\top, \qquad \hat{s}_i : \hat{\mathbf{s}}_i^\top, \qquad s_i b_j : \mathbf{B}_j \mathbf{s}_i^\top, \qquad r_{i'} : \mathbf{r}_{i'}, \qquad \alpha : \mathbf{a}, \qquad \hat{r}_{i'} : \hat{\mathbf{r}}_{i'}, \qquad r_{i'} b_j : \mathbf{r}_{i'} \mathbf{B}_j,$$

for $z \in [w_2], i \in [0, w_1], j \in [n], z' \in [m_2]$, and $i' \in [m_1]$ in all ciphertext-encoding polynomials output by $\mathsf{EncC}(x, p)$ *and all key-encoding polynomials output by* $\mathsf{EncK}(y, p)$*, then they evaluate to* $\mathbf{0}$.

Similarly, the PES *satisfies (d_1, d_2)-co-selective symbolic security if there exist* EncB, EncR, *and* EncS *as above except that inputs of these three algorithms are y, y, and (x, y), respectively. Finally, the* PES *satisfies (d_1, d_2)-symbolic security if it satisfies (d_1', d_2')-selective symbolic security such that $d_1' \leq d_1, d_2' \leq d_2$ and (d_1'', d_2'')-selective symbolic security such that $d_1'' \leq d_1, d_2'' \leq d_2$.*

Theorem 13 ([AC17]). *If there is a* $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{Pair})$ *for a predicate f satisfying the (d_1, d_2)-symbolic security, there is an adaptively secure* ABE *scheme for the same predicate f under the (d_1, d_2)-q-ratio assumption.*

8.2 Construction

We construct an ABKHE scheme Π_{ABKHE} from $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{Pair})$ for a predicate $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$. Let Π_{DSG} denote an ABE scheme from PES over dual system groups [AC16, AC17, CGW15]. Briefly speaking, Π_{ABKHE} is based on Π_{DSG} with three master secret keys $(\mathbf{u}_\iota)_{\iota \in [0,2]}$ by combining with Emura et al.'s KHPKE scheme Π_{KHPKE} [EHN+18]. A ciphertext of Π_{ABKHE} is described as $\mathsf{ct}_x = (\mathsf{ABE.ct}_x, \pi)$, where $\mathsf{ABE.ct}_x$ is a ciphertext of Π_{DSG} and π will play the same role as $\mathsf{KHPKE}.\pi$ in Π_{KHPKE}. Let $\mathsf{sk}_{y,\iota}$ denote a secret key of Π_{DSG} for a master secret key $\mathbf{u}_\iota$. Then, a decryption key and a homomorphic evaluation key are described as $\mathsf{dk}_y = (\mathsf{sk}_{y,\iota})_{\iota \in [0,2]}$ and $\mathsf{dk}_y = (\mathsf{sk}_{y,\iota})_{\iota \in [2]}$, respectively.

By following ABE scheme Π_{DSG} from PES over dual system groups [AC16, AC17, CGW15], mpk contains group elements $[\mathbf{A}]_1, [\mathbf{B}]_2, ([\mathbf{W}_j^\top \mathbf{A}]_1, [\mathbf{W}_j \mathbf{B}]_2)_{j \in [n]}$, while msk contains group elements $([\mathbf{u}_\iota]_2)_{\iota \in [0,2]}$. Then, an ABE ciphertext $\mathsf{ABE.ct}_x$ is computed by $[\mathbf{A}\mathbf{s}_i]_1$, $[\mathbf{A}\mathbf{s}_{w_1+i}]_1$, and $[\mathbf{W}_j^\top \mathbf{A}\mathbf{s}_i]_1$ that represent non-lone ciphertext-encoding variables s_i, lone ciphertext-encoding variables $\hat{s}_i$, and multiplications of common variables and non-lone ciphertext-encoding variable $s_i b_j$, respectively. Similarly, an ι-th secret key $\mathsf{sk}_{y,\iota}$ is computed by $[\mathbf{B}\mathbf{r}_{\iota,i'}]_2$, $[\mathbf{u}_\iota]_2$ and $[\mathbf{B}\mathbf{r}_{\iota,m_1+i'}]_2$, and $[\mathbf{W}_j \mathbf{B}\mathbf{r}_{\iota,i'}]_2$ that represent non-lone key-encoding variables $r_{i'}$, lone key-encoding variables α and $\hat{r}_{i'}$, and multiplications of common variables and non-lone key-encoding variable $r_{i'} b_j$, respectively.

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$. Run $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda)$ and $n \leftarrow \mathsf{Param}(\mathsf{par})$, and choose a collision-resistant hash function $H \leftarrow_R \mathcal{H}$, where $H : \{0,1\}^* \to \mathbb{Z}_p$. Sample $(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, uniformly random matrices $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$, and random vectors $(\mathbf{u}_\iota)_{\iota\in[0,2]} \leftarrow_R \mathbb{Z}_p^{k+1}$, then output

$$\mathsf{mpk} \coloneqq \Big(\mathcal{G}(1^\lambda), [\mathbf{A}]_1, [\mathbf{B}]_2, ([\mathbf{W}_j^\top \mathbf{A}]_1, [\mathbf{W}_j \mathbf{B}]_2)_{j\in[n]}, ([\mathbf{A}^\top \mathbf{u}_\iota]_T)_{\iota\in[0,2]}, H\Big)$$

and $\mathsf{msk} \coloneqq ([\mathbf{u}_\iota]_2)_{\iota\in[0,2]}$.

$\mathsf{Enc}(\mathsf{mpk}, x, \mu) \to \mathsf{ct}_x$. Run $\mathsf{EncC}(x, p)$ to obtain w_3 key-encoding polynomials $(c_1, \ldots, c_{w_3})$, sample $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{w_1+w_2} \leftarrow_R \mathbb{Z}_p^k$, and output $\mathsf{ct}_x \coloneqq ((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}_{1,t})_{t\in[w_3]}, \mathsf{ct}_T, \pi)$;

$$\mathsf{ct}_{0,i} \coloneqq [\mathbf{A}\mathbf{s}_i]_1, \qquad \mathsf{ct}_{1,t} \coloneqq \prod_{i\in[w_2]} [\mathbf{A}\mathbf{s}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1], j\in[n]} [\mathbf{W}_j^\top \mathbf{A}\mathbf{s}_i]_1^{\eta_{t,i,j}},$$
$$\mathsf{ct}_T \coloneqq \mu \cdot [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{u}_0]_T, \quad \pi \coloneqq [\mathbf{s}_0^\top \mathbf{A}^\top (\mathbf{u}_1 + h \cdot \mathbf{u}_2)]_T,$$

where $h = H((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, \mathsf{ct}_T)$.

$\mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y) \to (\mathsf{dk}_y, \mathsf{hk}_y)$. Run $\mathsf{EncK}(y, p)$ to obtain m_3 key-encoding polynomials $(k_1, \ldots, k_{m_3})$, sample $\mathbf{r}_{\iota,1}, \ldots, \mathbf{r}_{\iota,m_1+m_2} \leftarrow_R \mathbb{Z}_p^k$, and compute $\mathsf{sk}_{y,\iota} \coloneqq ((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]})$ for $\iota \in [0,2]$;

$$\begin{aligned} \mathsf{sk}_{\iota,0,i'} &\coloneqq [\mathbf{B}\mathbf{r}_{\iota,i'}]_2, \\ \mathsf{sk}_{\iota,1,t'} &\coloneqq [\mathbf{u}_\iota]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{\iota,m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_1], j\in[n]} [\mathbf{W}_j \mathbf{B}\mathbf{r}_{\iota,i'}]_2^{\phi_{t',i',j}}. \end{aligned} \tag{12}$$

Output $\mathsf{dk}_y \coloneqq (\mathsf{sk}_{y,\iota})_{\iota\in[0,2]}$ and $\mathsf{hk}_y \coloneqq (\mathsf{sk}_{y,\iota})_{\iota\in[2]}$.

$\mathsf{Eval}(\mathsf{mpk}, \mathsf{hk}_y, (\mathsf{ct}_x^{(\ell)})_{\ell\in[L]}) \to \mathsf{ct}_x/\perp$. Output $\perp$ if $f(x,y) = 0$ holds. Otherwise, parse $\mathsf{hk}_y = ((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]})_{\iota\in[2]}$ and $\mathsf{ct}_x^{(\ell)} = ((\mathsf{ct}_{0,i}^{(\ell)})_{i\in[0,w_1]}, (\mathsf{ct}_{1,t}^{(\ell)})_{t\in[w_3]}, \mathsf{ct}_T^{(\ell)}, \pi^{(\ell)})$, run $(\mathbf{E}, \overline{\mathbf{E}}) \leftarrow \mathsf{Pair}(x, y, p)$, and check whether the following conditions simultaneously hold for all $\ell \in [L]$:

- Compute $\mathsf{sk}_y \coloneqq ((\mathsf{sk}_{0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{1,t})_{t'\in[m_3]})$ in the same way as (12) except that $\mathbf{u}_\iota$ is replaced with a zero vector. It holds that

$$\prod_{i\in[0,w_1], t'\in[m_3]} e(\mathsf{ct}_{0,i}^{(\ell)}, \mathsf{sk}_{1,t'})^{E_{i,t'}} = \prod_{t\in[w_3], i'\in[m_1]} e(\mathsf{ct}_{1,t}^{(\ell)}, \mathsf{sk}_{0,i'})^{\overline{E}_{t,i'}}. \tag{13}$$

- It holds that

$$\frac{\prod_{i\in[0,w_1], t'\in[m_3]} e(\mathsf{ct}_{0,i}^{(\ell)}, \mathsf{sk}_{1,1,t'} \cdot \mathsf{sk}_{2,1,t'}^{h^{(\ell)}})^{E_{i,t'}}}{\prod_{t\in[w_3], i'\in[m_1]} e(\mathsf{ct}_{1,t}^{(\ell)}, \mathsf{sk}_{1,0,i'} \cdot \mathsf{sk}_{2,0,i'}^{h^{(\ell)}})^{\overline{E}_{t,i'}}} = \pi, \tag{14}$$

  where $h^{(\ell)} = H((\mathsf{ct}_{0,i}^{(\ell)})_{i\in[0,w_1]}, \mathsf{ct}_T^{(\ell)})$.

If one of the conditions does not hold for some $\ell \in [L]$, output $\perp$. Otherwise, run $\mathsf{ct}_x^{(0)} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, 1_T)$ and output $\mathsf{ct}_x := ((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}_{1,t})_{t\in[w_3]}, \mathsf{ct}_T, \pi)$;

$$\mathsf{ct}_{0,i} := \prod_{\ell\in[0,L]} \mathsf{ct}_{0,i}^{(\ell)}, \qquad \mathsf{ct}_{1,t} := \prod_{\ell\in[0,L]} \mathsf{ct}_{1,t}^{(\ell)}, \qquad \mathsf{ct}_T := \prod_{\ell\in[0,L]} \mathsf{ct}_T^{(\ell)},$$

$$\pi := \frac{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}, \mathsf{sk}_{1,1,t} \cdot \mathsf{sk}_{2,1,t'}^h)^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}, \mathsf{sk}_{1,0,i'} \cdot \mathsf{sk}_{2,0,i'}^h)^{\overline{E}_{t,i'}}},$$

where $h = H((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, \mathsf{ct}_T)$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_y, \mathsf{ct}_x) \to \mu/\perp$. Output $\perp$ if $f(x,y) = 0$ holds. Otherwise, parse $\mathsf{dk}_y = ((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]})_{\iota\in[0,2]}$ and $\mathsf{ct}_x = ((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}_{1,t})_{t\in[w_3]}, \mathsf{ct}_T, \pi)$, run $(\mathbf{E}, \overline{\mathbf{E}}) \leftarrow \mathsf{Pair}(x, y, p)$, and check whether the conditions (13) and (14) simultaneously hold. If one of the conditions does not hold, output $\perp$. Otherwise, output

$$\mathsf{ct}_T \cdot \frac{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}, \mathsf{sk}_{0,0,i'})^{\overline{E}_{t,i'}}}{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}, \mathsf{sk}_{0,1,t'})^{E_{i,t'}}}.$$

Theorem 14. *The proposed* ABKHE *scheme* Π_{ABKHE} *satisfies correctness if the* PES = (Param, EncC, EncK, Pair) *for* f *satisfies the correctness.*

Proof of Theorem 14. If it holds that

$$\frac{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}, \mathsf{sk}_{\iota,1,t'})^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}, \mathsf{sk}_{\iota,0,i'})^{\overline{E}_{t,i'}}} = [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{u}_\iota]_T \tag{15}$$

for any $\mathsf{ct}_x = ((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}_{1,t})_{t\in[w_3]}, \mathsf{ct}_T, \pi) \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, \mu)$ and $(((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]}))_{\iota\in[0,2]}, ((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]}))_{\iota\in[2]}) \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, y)$ such that $f(x,y) = 1$, we can complete the proof. We will prove the quality (15) at the end of the proof.

The equality (15) implies the condition (13) by setting $\mathbf{u}_\iota$ as a zero vector. The equality (15) also implies the condition (14) since it holds that

$$\begin{aligned}
&\frac{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}^{(\ell)}, \mathsf{sk}_{1,1,t'} \cdot \mathsf{sk}_{2,1,t'}^{h^{(\ell)}})^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}^{(\ell)}, \mathsf{sk}_{1,0,i'} \cdot \mathsf{sk}_{2,0,i'}^{h^{(\ell)}})^{\overline{E}_{t,i'}}} \\
&= \frac{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}^{(\ell)}, \mathsf{sk}_{1,1,t'})^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}^{(\ell)}, \mathsf{sk}_{1,0,i'})^{\overline{E}_{t,i'}}} \cdot \left(\frac{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}^{(\ell)}, \mathsf{sk}_{2,1,t'})^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}^{(\ell)}, \mathsf{sk}_{2,0,i'})^{\overline{E}_{t,i'}}} \right)^{h^{(\ell)}} \\
&= [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{u}_1]_T \cdot [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{u}_2]_T^{h^{(\ell)}} = \pi.
\end{aligned}$$

Thus, the Eval and Dec do not output $\perp$.

For $(\mathsf{ct}_x^{(\ell)})_{\ell\in[L]}$ which is an input of Eval and $\mathsf{ct}_x^{(0)}$ which is created during Eval, let

$$\begin{aligned}
&\mathsf{ct}_{0,i}^{(\ell)} = [\mathbf{A}\mathbf{s}_i^{(\ell)}]_1, \qquad \mathsf{ct}_{1,t}^{(\ell)} = \prod_{i\in[w_2]} [\mathbf{A}\mathbf{s}_{w_1+i}^{(\ell)}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1],j\in[n]} [\mathbf{W}_j^\top \mathbf{A}\mathbf{s}_i^{(\ell)}]_1^{\eta_{t,i,j}}, \\
&\mathsf{ct}_T^{(\ell)} = \mu^{(\ell)} \cdot [(\mathbf{s}_0^{(\ell)})^\top \mathbf{A}^\top \mathbf{u}_0]_T, \quad \pi^{(\ell)} = [(\mathbf{s}_0^{(\ell)})^\top \mathbf{A}^\top (\mathbf{u}_1 + h^{(\ell)} \cdot \mathbf{u}_2)]_T,
\end{aligned}$$

where $h^{(\ell)} = H((\mathsf{ct}_{0,i}^{(\ell)})_{i\in[0,w_1]}, \mathsf{ct}_T^{(\ell)})$ for $\ell \in [0, L]$. Let $\mathsf{ct}_x = ((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}_{1,t})_{t\in[w_3]}, \mathsf{ct}_T, \pi)$ denote an output of Eval and $\mathbf{s}_i = \sum_{\ell\in[0,L]} \mathbf{s}_i^{(\ell)}$. Then, we have

$$\begin{aligned}
\mathsf{ct}_{0,i} &= \prod_{\ell\in[0,L]} [\mathbf{A}\mathbf{s}_i^{(\ell)}]_1 = [\mathbf{A}\mathbf{s}_i]_1, \\
\mathsf{ct}_{1,t} &= \prod_{\ell\in[0,L]} \mathsf{ct}_{1,t}^{(\ell)} = \prod_{i\in[w_2]} [\mathbf{A}\mathbf{s}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1],j\in[n]} [\mathbf{W}_j^\top \mathbf{A}\mathbf{s}_i]_1^{\eta_{t,i,j}}, \\
\mathsf{ct}_T &= \prod_{\ell\in[0,L]} \mathsf{ct}_T^{(\ell)} = \left(\prod_{\ell\in[L]} \mu^{(\ell)}\right) \cdot [\mathbf{s}_0^\top \mathbf{A}^\top \mathbf{u}_0]_T.
\end{aligned}$$

Moreover, as the case of the condition (14), we have

$$\pi = \frac{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}, \mathsf{sk}_{1,1,t} \cdot \mathsf{sk}_{2,1,t'}^h)^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}, \mathsf{sk}_{1,0,i'} \cdot \mathsf{sk}_{2,0,i'}^h)^{\overline{E}_{t,i'}}} = [\mathbf{s}_0^\top \mathbf{A}^\top (\mathbf{u}_1 + h \cdot \mathbf{u}_2)]_T,$$

where $h = H((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, \mathsf{ct}_T)$. Thus, an output of Eval follow the same distribution as an output of Enc for a plaintext $\prod_{\ell\in[L]} \mu^{(\ell)}$. Finally, the equality (15) implies that an output of Dec is μ.

To conclude the proof, we prove the equality (15). Observe that the left-hand side the equality (15) satisfies

$$\begin{aligned}
&\frac{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}, \mathsf{sk}_{\iota,1,t'})^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}, \mathsf{sk}_{\iota,0,i'})^{\overline{E}_{t,i'}}} \\
&= \frac{\prod_{i\in[0,w_1],t'\in[m_3]} e([\mathbf{A}\mathbf{s}_i]_1, [\mathbf{u}_\iota]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{\iota,m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\mathbf{W}_j \mathbf{B}\mathbf{r}_{\iota,i'}]_2^{\phi_{t',i',j}})^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\prod_{i\in[w_2]} [\mathbf{A}\mathbf{s}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1],j\in[n]} [\mathbf{W}_j^\top \mathbf{A}\mathbf{s}_i]_1^{\eta_{t,i,j}}, [\mathbf{B}\mathbf{r}_{\iota,i'}]_2)^{\overline{E}_{t,i'}}}.
\end{aligned}$$

Moreover, the discrete logarithm of the value with base $e(g_1, g_2)$ is

$$\begin{aligned}
&\sum_{i\in[0,w_1],t'\in[m_3]} E_{i,t'} \cdot \mathbf{s}_i^\top \mathbf{A}^\top \cdot \left(\phi_{t'}\mathbf{u}_\iota + \sum_{i'\in[m_2]} \phi_{t',i'} \mathbf{B}\mathbf{r}_{\iota,m_1+i'} + \sum_{i'\in[m_1],j\in[n]} \phi_{t',i',j} \mathbf{W}_j \mathbf{B}\mathbf{r}_{\iota,i'}\right) \\
&- \sum_{t\in[w_3],i'\in[m_1]} \overline{E}_{t,i'} \cdot \left(\sum_{i\in[w_2]} \eta_{t,i} \mathbf{s}_{w_1+i}^\top \mathbf{A}^\top + \sum_{i\in[0,w_1],j\in[n]} \eta_{t,i,j} \mathbf{s}_i^\top \mathbf{A}^\top \mathbf{W}_j\right) \cdot \mathbf{B}\mathbf{r}_{\iota,i'}.
\end{aligned}$$

Thus, if we substitute

$$\begin{array}{llll}
s_i : \mathbf{A}\mathbf{s}_i, & \hat{s}_i : \mathbf{A}\mathbf{s}_{w_1+i}, & s_i b_j : \mathbf{W}_j^\top \mathbf{A}\mathbf{s}_i, & \\
\alpha : \mathbf{u}_\iota, & r_{i'} : \mathbf{B}\mathbf{r}_{\iota,i'}, & \hat{r}_{i'} : \mathbf{B}\mathbf{r}_{\iota,m_1+i'}, & r_{i'} b_j : \mathbf{W}_j \mathbf{B}\mathbf{r}_{\iota,i'},
\end{array}$$

the correctness of PES implies the equality (15). □

8.3 Security

In this section, we prove that the proposed ABKHE scheme Π_{ABKHE} satisfies the adaptive KH-CCA security.

Theorem 15. *If the* $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{Pair})$ *for* f *satisfies the perfect security and the symbolic security,* Π_{ABKHE} *satisfies the adaptive* KH-CCA *security under the matrix DDH assumption and the* q*-ratio assumption, respectively.*

We will prove Theorem 15 in the case of perfect security since proof for symbolic security is essentially the same.

8.3.1 Semi-functional Distributions

To prove Theorem 15, we prepare auxiliary *semi-functional* distributions for a ciphertext and an ABE secret key by following [AC16, CGW15].

Semi-functional Ciphertext. A semi-functional ciphertext ct_x for x encrypting μ is defined as $\mathsf{ct}_x = ((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}_{1,t})_{t\in[w_3]}, \mathsf{ct}_T, \pi)$;

$$\mathsf{ct}_{0,i} \coloneqq [\mathbf{c}_i]_1, \qquad \mathsf{ct}_{1,t} \coloneqq \prod_{i\in[w_2]} [\mathbf{c}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1],j\in[n]} [\mathbf{W}_j^\top \mathbf{c}_i]_1^{\eta_{t,i,j}},$$
$$\mathsf{ct}_T \coloneqq \mu \cdot [\mathbf{c}_0^\top \mathbf{u}_0]_T, \qquad \pi \coloneqq [\mathbf{c}_0^\top(\mathbf{u}_1 + h\cdot\mathbf{u}_2)]_T,$$

where $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{w_1+w_2} \leftarrow_R \mathbb{Z}_p^{k+1}$ and $h = H((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, \mathsf{ct}_T)$.

Semi-functional Secret Key. An ι-th semi-functional secret key $\mathsf{sk}_{y,\iota}$ for y is defined as $\mathsf{sk}_{y,\iota} = ((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]})$;

$$\begin{aligned}\mathsf{sk}_{\iota,0,i'} &= [\mathbf{B}\mathbf{r}_{\iota,i'}]_2,\\ \mathsf{sk}_{\iota,1,t'} &= [\mathbf{u}_\iota + \alpha_{\iota,y}\mathbf{a}^\perp]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{\iota,m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\mathbf{W}_j\mathbf{B}\mathbf{r}_{\iota,i'}]_2^{\phi_{t',i',j}},\end{aligned}$$

where $\mathbf{r}_{\iota,1}, \ldots, \mathbf{r}_{\iota,m_1+m_2} \leftarrow_R \mathbb{Z}_p^k$ and $\alpha_{\iota,y} \leftarrow_R \mathbb{Z}_p$.

Intuitively, a normal ciphertext (resp. secret key) is a special case of a semi-functional ciphertext (resp. secret key) only if $\mathbf{c}_i$ lives in the span of $\mathbf{A}$ and $\mathbf{c}_0^\top\mathbf{a}^\perp = 0$ holds (resp. $\alpha_{\iota,y} = 0$), while such situations occur only with negligible probability. For a semi-functional $\mathsf{ct}_x = ((\mathsf{ct}_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}_{1,t})_{t\in[w_3]}, \mathsf{ct}_T, \pi)$ and a semi-functional $\mathsf{sk}_{y,\iota} = ((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]})$, the equation (15) becomes

$$\frac{\prod_{i\in[0,w_1],t'\in[m_3]} e(\mathsf{ct}_{0,i}, \mathsf{sk}_{\iota,1,t'})^{E_{i,t'}}}{\prod_{t\in[w_3],i'\in[m_1]} e(\mathsf{ct}_{1,t}, \mathsf{sk}_{\iota,0,i'})^{\overline{E}_{t,i'}}} = [\mathbf{c}_0^\top(\mathbf{u}_\iota + \alpha_{\iota,y}\mathbf{a}^\perp)]_T = [\mathbf{c}_0^\top\mathbf{u}_\iota]_T \cdot [\mathbf{c}_0^\top\mathbf{a}^\perp]_T^{\alpha_{\iota,y}}.$$

Therefore, the correctness does not hold since it holds that $\mathbf{c}_0^\top\mathbf{a}^\perp \neq 0 \wedge \tilde{\alpha}_\iota \neq 0$ which implies $[\mathbf{c}_0^\top\mathbf{a}^\perp]_T^{\alpha_{\iota,y}} \neq 1_T$ with overwhelming probability. On the other hand, the correctness holds if either ct_x or $\mathsf{sk}_{y,\iota}$ follows a normal distribution.

8.3.2 Proof of Theorem 15

Although we already explained the intuition of a proof in Section 1.3.3, we provide a more detailed overview. We call $\mathcal{A}$'s decryption query on $(y, \mathsf{ct}_x = ((\mathsf{ct}_{0,i} = \mathbf{c}_i)_{i\in[0,w_1]}, \cdots))$ a *critical decryption query* if ct_x is valid, $\mathsf{ct}_x \notin \mathcal{L}$ holds, and $\mathbf{c}_0$ does not live in the span of $\mathbf{A}$. We call $\mathcal{A}$'s homomorphic evaluation key reveal query on y a *critical homomorphic evaluation key reveal query* if $f(x^\star, y) = 1$

Table 2: Distributions of ciphertexts $\mathsf{ct}^{[1]}_{x^\star} = \mathsf{ct}^\star_{x^\star}, \ldots, \mathsf{ct}^{[D]}_{x^\star} \in \mathcal{L}$ in $\mathsf{Game}_{3,d}, \ldots, \mathsf{Game}_{9,d}$

	$\mathsf{ct}^{[1]}_{x^\star}, \ldots, \mathsf{ct}^{[d-1]}_{x^\star}$	$\mathsf{ct}^{[d]}_{x^\star}$	$\mathsf{ct}^{[d+1]}_{x^\star}, \ldots, \mathsf{ct}^{[D]}_{x^\star}$
$\mathsf{Game}_{3,d}$	normal encryptions of random strings	semi-functional encryption of $\mu^{[d]}$	normal encryptions of $\mu^{[d+1]}, \ldots, \mu^{[D]}$
$\mathsf{Game}_{4,d}$	normal encryptions of random strings	semi-functional encryption of $\mu^{[d]}$	normal encryptions of $\mu^{[d+1]}, \ldots, \mu^{[D]}$
$\mathsf{Game}_{5,d}$	normal encryptions of random strings	semi-functional encryption of $\mu^{[d]}$	normal encryptions of $\mu^{[d+1]}, \ldots, \mu^{[D]}$
$\mathsf{Game}_{6,d}$	normal encryptions of random strings	semi-functional encryption of a random string	normal encryptions of $\mu^{[d+1]}, \ldots, \mu^{[D]}$
$\mathsf{Game}_{7,d}$	normal encryptions of random strings	semi-functional encryption of a random string	normal encryptions of $\mu^{[d+1]}, \ldots, \mu^{[D]}$
$\mathsf{Game}_{8,d}$	normal encryptions of random strings	semi-functional encryption of a random string	normal encryptions of $\mu^{[d+1]}, \ldots, \mu^{[D]}$
$\mathsf{Game}_{9,d}$	normal encryptions of random strings	normal encryption of a random string	normal encryptions of $\mu^{[d+1]}, \ldots, \mu^{[D]}$

holds. Let $\mathsf{ct}^\star_{x^\star} = ((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}^\star_{1,t})_{t\in[w_3]}, \mathsf{ct}^\star_T, \pi^\star)$ denote a challenge ciphertext for a challenge ciphertext attribute $x^\star$ and a message $\mu^\star_{\mathsf{coin}}$, where $h^\star = H((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, \mathsf{ct}^\star_T)$. Let D denote the number of ciphertexts in $\mathcal{L}$ at the end of the game, where the challenge ciphertext $\mathsf{ct}^\star_{x^\star}$ is the first ciphertext and $\mathcal{A}$ makes $D-1$ dependent evaluation queries. Let $\mathsf{ct}^{[d]}_{x^\star} = ((\mathsf{ct}^{[d]}_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}^{[d]}_{1,t})_{t\in[w_3]}, \mathsf{ct}^{[d]}_T, \pi^{[d]})$ denote d-th ciphertext in $\mathcal{L}$ and treat it as an encryption of $\mu^{[d]}$, where $\mathsf{ct}^{[1]}_{x^\star} = \mathsf{ct}^\star_{x^\star}$ and $\mu^{[1]} = \mu^\star_{\mathsf{coin}}$.

We prove Theorem 15 by using a sequence of games $\mathsf{Game}_0, \mathsf{Game}_1, \mathsf{Game}_2, \mathsf{Game}_{3,1}, \ldots, \mathsf{Game}_{9,1}, \mathsf{Game}_{3,2}, \ldots, \mathsf{Game}_{3,D}, \ldots, \mathsf{Game}_{6,D}$, where it holds that $\mathsf{Game}_0 \approx_c \mathsf{Game}_1 = \mathsf{Game}_2 \approx_c \mathsf{Game}_{3,1}$ and $\mathsf{Game}_{9,d-1} \approx_c \mathsf{Game}_{3,d} \approx_c \cdots \approx_c \mathsf{Game}_{5,d} \approx \mathsf{Game}_{6,d} \approx_c \cdots \approx_c \mathsf{Game}_{9,d}$. The roles of Game_1 and Game_2 are essentially the same as in the proof of Theorem 11. Given the challenge cipehrtext $\mathsf{ct}^\star_{x^\star}$, $\mathcal{A}$ can randomize it and compute $((\overline{\mathsf{ct}^\star}_{0,i})_{i\in[0,w_1]}, (\overline{\mathsf{ct}^\star}_{1,t})_{t\in[w_3]}, \overline{\mathsf{ct}^\star}_T)$ such that the decryption result is $\mu^\star_{\mathsf{coin}}$ by ignoring the condition (14) and $((\overline{\mathsf{ct}^\star}_{0,i})_{i\in[0,w_1]}, (\overline{\mathsf{ct}^\star}_{1,t})_{t\in[w_3]}, \overline{\mathsf{ct}^\star}_T) \neq ((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}^\star_{1,t})_{t\in[w_3]}, \mathsf{ct}^\star_T)$ holds. If it holds that $H((\overline{\mathsf{ct}^\star}_{0,i})_{i\in[0,w_1]}, \overline{\mathsf{ct}^\star}_T) = h^\star$, a decryption result of a cipehrtext $((\overline{\mathsf{ct}^\star}_{0,i})_{i\in[0,w_1]}, (\overline{\mathsf{ct}^\star}_{1,t})_{t\in[w_3]}, \overline{\mathsf{ct}^\star}_T, \pi^\star)$ is $\mu^\star_{\mathsf{coin}}$ and $((\overline{\mathsf{ct}^\star}_{0,i})_{i\in[0,w_1]}, (\overline{\mathsf{ct}^\star}_{1,t})_{t\in[w_3]}, \overline{\mathsf{ct}^\star}_T, \pi^\star) \neq \mathsf{ct}^\star_{x^\star}$ holds. In Game_1, we use the collision resistance of H and prevent the attack. In Game_2, we change how to compute $\mathsf{ct}^{[d]}_{x^\star}$ for $d \in [2, D]$ so that the distribution of $\mathsf{ct}^{[d]}_{x^\star}$ does not depend on $\mathsf{ct}^{[d']}_{x^\star}$ for $d' \in [d-1]$. Game_1 and Game_2 follow the same distribution from $\mathcal{A}$'s view.

The role of $\mathsf{Game}_{3,d}, \mathsf{Game}_{6,d}$, and $\mathsf{Game}_{9,d}$ in a proof of Π_{ABKHE} (Theorem 15) are similar to $\mathsf{Game}_{3,d}$, $\mathsf{Game}_{4,d}$, and $\mathsf{Game}_{5,d}$ in the proof of Π_{KHPKE} (Theorem 11). As illustrated in Table 2, we change the distributions of ciphertexts $\mathsf{ct}_{x^\star}^{[d]} \in \mathcal{L}$ in $\mathsf{Game}_{3,d}, \mathsf{Game}_{6,d}$, and $\mathsf{Game}_{9,d}$, where $\mathsf{ct}_{x^\star}^{[1]}, \ldots, \mathsf{ct}_{x^\star}^{[d-1]}$ (resp. $\mathsf{ct}_{x^\star}^{[d+1]}, \ldots, \mathsf{ct}_{x^\star}^{[D]}$) are always normal encryptions of random strings (resp. normal encryptions of $\mu^{[d+1]}, \ldots, \mu^{[D]}$) in $\mathsf{Game}_{3,d}, \ldots, \mathsf{Game}_{9,d}$. In particular, $\mathsf{ct}_{x^\star}^{[d]}$ is a normal encryption of $\mu^{[d]}$ in $\mathsf{Game}_{9,d-1}$, while it becomes a semi-functional encryption of $\mu^{[d]}$ in $\mathsf{Game}_{3,d}$, a semi-functional encryption of a random string in $\mathsf{Game}_{6,d}$, and a normal encryption of a random string in $\mathsf{Game}_{9,d}$. As the proof of Lemma 15, the $(w_1 + w_2)$-fold matrix DDH assumption over $\mathbb{G}_1$ ensures that $\mathsf{Game}_{9,d-1} \approx_c \mathsf{Game}_{3,d}$ holds by following the dual system technique of Π_{DSG} [AC16, CGW15]. However, unlike the case of Π_{KHPKE} (Lemma 15), we cannot immediately prove $\mathsf{Game}_{3,d} \approx \mathsf{Game}_{6,d}$ in the sense that computationally unbounded $\mathcal{A}$ can distinguish a semi-functional encryption of $\mu^{[d]}$ and that of a random string. In the proof of Π_{KHPKE} (Lemma 15), we proved the indistinguishability based on the fact that $\mathbf{u}_0$ was not revealed to $\mathcal{A}$ and $\mathcal{A}$ cannot make critical decryption queries. In contrast, the computationally unbounded $\mathcal{A}$ against Π_{ABKHE} can make a decryption key reveal query (resp. homomorphic evaluation key reveal query) on y such that $f(x^\star, y) = 0$ and recover $\mathbf{u}_0$ (resp. recover $\mathbf{u}_1, \mathbf{u}_2$ and make a critical decryption query).

To resolve the issue, we want to use the dual system technique of Π_{DSG} [AC16, CGW15] and change some of ABE secret keys $\mathsf{sk}_{y,\iota}$ such that $f(x^\star, y) = 0$ to be semi-functional so that the computationally unbounded $\mathcal{A}$ cannot recover $\mathbf{u}_0, \mathbf{u}_1$, and $\mathbf{u}_2$. What we have to care is that we cannot change all ABE secret keys $\mathsf{sk}_{y,\iota}$ which $\mathcal{A}$ receives to be semi-functional since $\mathcal{A}$ against Π_{ABKHE} can receive $\mathsf{sk}_{y,\iota}$ such that $f(x^\star, y) = 1$ unlike the case of Π_{DSG}. In particular, the definition of the adaptive $\mathsf{KH\text{-}CCA}$ security ensures that $\mathcal{A}$ cannot make decryption key reveal queries on y such $f(x^\star, y) = 1$; thus, all $\mathsf{sk}_{y,0}$ $\mathcal{A}$ receives satisfy $f(x^\star, y) = 0$. In contrast, $\mathcal{A}$ can make homomorphic evaluation key reveal queries on y and receives $\mathsf{sk}_{y,1}, \mathsf{sk}_{y,2}$ such that $f(x^\star, y) = 1$. Thus, we try to change only the required ABE secret keys $\mathsf{sk}_{y,\iota}$ to be semi-functional. To this end, we divide $\mathcal{A}$'s attack strategies into two types called Type-1 and Type-2 which are defined as follows:

- $\mathcal{A}$ is called Type-1 if it makes a critical homomorphic evaluation key reveal query in Phase 1.
- $\mathcal{A}$ is called Type-2 if it does not make a critical homomorphic evaluation key reveal query in Phase 1.

By definition, Type-1 and Type-2 are mutually exclusive and cover all possible strategies of $\mathcal{A}$. During the proof of Π_{KHPKE} (Lemma 16), we used a similar division and proved the indistinguishability of $\mathsf{KHPKE.ct}^{[d]}$. In contrast, we use the division and employ distinct game sequences depending on $\mathcal{A}$'s types. In $\mathsf{Game}_{4,d}$, we change all $\mathsf{sk}_{y,0}$ $\mathcal{A}$ receives to be semi-functional regardless of Type-1 and Type-2. Since $f(x^\star, y) = 0$ holds as we explained above, the $(m_1 + m_2)$-fold matrix DDH assumption over $\mathbb{G}_2$ ensures that $\mathsf{Game}_{3,d} \approx_c \mathsf{Game}_{4,d}$ holds by following the dual system technique of Π_{DSG} [AC16, CGW15]. In $\mathsf{Game}_{5,d}$, we change $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$ $\mathcal{A}$ receives to be semi-functional until $\mathcal{A}$ makes the first homomorphic evaluation key reveal query only if $\mathcal{A}$ is Type-2. Observe that we cannot apply the same change to $\mathcal{A}$ of Type-1 since we do not know whether $f(x^\star, y) = 0$ holds upon $\mathcal{A}$'s homomorphic evaluation key reveal queries in Phase 1. In contrast, the definition of Type-2 ensures that $\mathcal{A}$ of Type-2 makes critical homomorphic evaluation key reveal queries only in Phase 2. Thus, we can check when $\mathcal{A}$ makes the first critical homomorphic evaluation key reveal query. Then, the $(m_1 + m_2)$-fold matrix DDH assumption over $\mathbb{G}_2$ ensures that $\mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{5,d}$ holds by following the dual system technique of Π_{DSG} [AC16, CGW15]. Finally, we can conclude that $\mathcal{A}$ cannot recover $\mathbf{u}_0$ since all $\mathsf{sk}_{y,0}$ $\mathcal{A}$ receives are semi-functional, while the above changes ensure that $\mathcal{A}$ cannot make critical decryption queries. Thus, we can prove $\mathsf{Game}_{5,d} \approx \mathsf{Game}_{6,d}$. Afterward, we change all $\mathsf{sk}_{y,\iota}$ to be normal in $\mathsf{Game}_{7,d}$ and $\mathsf{Game}_{8,d}$. Then, we change a distribution of $\mathsf{ct}_{x^\star}^{[d]}$ in $\mathsf{Game}_{9,d}$. We can prove $\mathsf{Game}_{6,d} \approx_c \mathsf{Game}_{7,d} \approx_c \mathsf{Game}_{8,d}$ (resp. $\mathsf{Game}_{8,d} \approx_c \mathsf{Game}_{9,d}$) in the

Table 3: Distributions of ABE secret keys $\mathsf{sk}_{y,0}, \mathsf{sk}_{y,1}$, and $\mathsf{sk}_{y,2}$ in $\mathsf{Game}_{3,d}, \ldots, \mathsf{Game}_{9,d}$

	$\mathsf{sk}_{y,0}$	$\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$ until the critical hk_y query	$\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$ after the critical hk_y query
$\mathsf{Game}_{3,d}$	normal	normal	normal
$\mathsf{Game}_{4,d}$	semi-functional	normal	normal
$\mathsf{Game}_{5,d}$	semi-functional	semi-functional	normal
$\mathsf{Game}_{6,d}$	semi-functional	semi-functional	normal
$\mathsf{Game}_{7,d}$	semi-functional	normal	normal
$\mathsf{Game}_{8,d}$	normal	normal	normal
$\mathsf{Game}_{9,d}$	normal	normal	normal

same way as $\mathsf{Game}_{5,d} \approx_c \mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{3,d}$ (resp. $\mathsf{Game}_{3,d} \approx_c \mathsf{Game}_{9,d-1}$). Table 3 summarizes distributions of $\mathsf{sk}_{y,\iota}$ in each game.

Proof of Theorem 15. We use the following sequence of games.

Game_0. This is the adaptive KH-CCA security game. Hereafter, let $\mathsf{ct}^\star_{x^\star} = ((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}^\star_{1,t})_{t\in[w_3]}, \mathsf{ct}^\star_T, \pi^\star)$ denote a challenge ciphertext for a challenge ciphertext attribute $x^\star$ and a message $\mu^\star_{\mathsf{coin}}$, where $h^\star = H((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, \mathsf{ct}^\star_T)$.

Game_1. This is the same as Game_0 except that a collision does not occur for a hash function H among all ciphertexts that appeared in the security game.

The collision resistance of H ensures that $\mathsf{Game}_0 \approx_c \mathsf{Game}_1$ holds.

Game_2. This is the same as Game_1 except the answers to dependent evaluation queries so that the distributions of $\mathsf{ct}^{[1]}_{x^\star} = \mathsf{ct}^\star_{x^\star}, \ldots, \mathsf{ct}^{[D]}_{x^\star} \in \mathcal{L}$ are independent. In Game_1, $\mathcal{C}$ runs Eval algorithm with inputs $\mathsf{ct}^{[1]}, \ldots, \mathsf{ct}^{[d-1]}$ that are answers to $\mathcal{A}$'s challenge query and dependent evaluation queries, and creates an evaluated ciphertext $\mathsf{ct}^{[d]}$. In Game_2, upon $\mathcal{A}$'s challenge query, $\mathcal{C}$ runs Enc algorithm and creates two ciphertexts $\mathsf{ct}^\star$ and $\widetilde{\mathsf{ct}^\star}$ in the same way as in Game_1, sends $\mathsf{ct}^\star$ to $\mathcal{A}$ as the challenge ciphertext, and stores both ciphertexts $(\mathsf{ct}^\star, \widetilde{\mathsf{ct}^\star}) \in \mathcal{L}$. Upon $\mathcal{A}$'s first dependent evaluation query, $\mathcal{C}$ runs Eval algorithm with inputs $\widetilde{\mathsf{ct}^{[1]}}$ in place of $\mathsf{ct}^{[1]}$ that is the answer to $\mathcal{A}$'s challenge query, and creates two evaluated ciphertexts $\mathsf{ct}^{[2]}$ and $\widetilde{\mathsf{ct}^{[2]}}$ in the same way as in Game_1, sends $\mathsf{ct}^{[2]}$ to $\mathcal{A}$ as the answer to the evaluation query, and stores both ciphertexts $(\mathsf{ct}^{[2]}, \widetilde{\mathsf{ct}^{[2]}}) \in \mathcal{L}$. In the same way, upon $\mathcal{A}$'s $(d-1)$-th dependent evaluation query, $\mathcal{C}$ runs Eval algorithm with inputs $\widetilde{\mathsf{ct}^{[1]}}, \ldots, \widetilde{\mathsf{ct}^{[d-1]}}$ in place of $\mathsf{ct}^{[1]}, \ldots, \mathsf{ct}^{[d-1]}$ that are the answers to $\mathcal{A}$'s challenge query and dependent evaluation queries, and creates two evaluated ciphertexts $\mathsf{ct}^{[d]}$ and $\widetilde{\mathsf{ct}^{[d]}}$ in the same way as in Game_1, sends $\mathsf{ct}^{[d]}$ to $\mathcal{A}$ as the answer to the evaluation query, and stores both ciphertexts $(\mathsf{ct}^{[d]}, \widetilde{\mathsf{ct}^{[d]}}) \in \mathcal{L}$. In Game_1 and Game_2, all ciphertexts $\mathsf{ct}^{[d]}$ and $\widetilde{\mathsf{ct}^{[d]}}$ follow the same distribution for $d \in [D]$.

From now on, we change a distribution of d-th ciphertext $\mathsf{ct}^{[d]}_{x^\star} = (\cdots, \mathsf{ct}^{[d]}_T, \cdots) \in \mathcal{L}$ for $d \in [D]$ one by one so that $\mathsf{ct}^{[d]}_T$ is independent of the other elements of $\mathsf{ct}^{[d]}_{x^\star}$ and distributed uniformly at random over $\mathbb{G}_T$. For this purpose, we use the following sequence of games $\mathsf{Game}_{3,d}, \ldots, \mathsf{Game}_{9,d}$ for $d \in [D]$, where $\mathsf{Game}_{9,0} = \mathsf{Game}_2$ and the proof terminates in $\mathsf{Game}_{6,D}$.

$\mathsf{Game}_{3,d}$. This is the same as $\mathsf{Game}_{9,d-1}$ except $\mathcal{C}$'s answer to the challenge query if $d = 1$ and a dependent evaluation query if $d \in [2, D]$. In particular, $\mathcal{C}$ creates the challenge ciphertext $\mathsf{ct}^{\star}_{x^\star}$ as a semi-functional encryption of $\mu^{\star}_{\mathsf{coin}}$ if $d = 1$ and $\mathsf{ct}^{(0)}$ as a semi-functional encryption of 1_T if $d \in [2, D]$, while $\mathcal{C}$ creates $\widetilde{\mathsf{ct}}^{[1]}, \ldots, \widetilde{\mathsf{ct}}^{[D]}$ in the same way as in $\mathsf{Game}_{2,d}$.

We can prove $\mathsf{Game}_{9,d-1} \approx_c \mathsf{Game}_{3,d}$ under the matrix DDH assumption over $\mathbb{G}_1$ by following the dual system technique of Π_{DSG} [AC16, CGW15].

Lemma 19 ($\mathsf{Game}_{9,d-1} \approx_c \mathsf{Game}_{3,d}$). *If the matrix DDH assumption over $\mathbb{G}_1$ holds, $\mathsf{Game}_{9,d-1}$ and $\mathsf{Game}_{3,d}$ are computationally indistinguishable for any PPT $\mathcal{A}$.*

We will prove Lemma 19 in Section 8.3.3.

$\mathsf{Game}_{4,d}$. This is the same as $\mathsf{Game}_{3,d}$ except that $\mathcal{C}$ answers semi-functional $\mathsf{sk}_{y,0}$ upon $\mathcal{A}$'s decryption key reveal queries on y. We note that $\mathcal{C}$ still uses normal $\mathsf{sk}_{y,0}$ to answer $\mathcal{A}$'s decryption queries as in $\mathsf{Game}_{3,d}$.

Since $f(x^\star, y) = 0$ holds due to the definition of the adaptive $\mathsf{KH\text{-}CCA}$ security game, we can prove $\mathsf{Game}_{3,d} \approx_c \mathsf{Game}_{4,d}$ under the matrix DDH assumption over $\mathbb{G}_2$ by following the dual system technique of Π_{DSG} [AC16, CGW15].

Lemma 20 ($\mathsf{Game}_{3,d} \approx_c \mathsf{Game}_{4,d}$). *If the PES satisfies the perfect security and the matrix DDH assumption over $\mathbb{G}_2$ holds, $\mathsf{Game}_{3,d}$ and $\mathsf{Game}_{4,d}$ are computationally indistinguishable for any PPT $\mathcal{A}$.*

We will prove Lemma 20 in Section 8.3.4. Intuitively, Lemma 20 implies that the dual system technique of Π_{DSG} [AC16, CGW15] is required that $\mathcal{A}$ cannot create any semi-functional ciphertexts ct_x in Phase 1. Otherwise, it can distinguish normal and semi-functional $\mathsf{sk}_{y,0}$ such that $f(x, y) = 1$, where the fact contradicts to the proofs of Π_{DSG} [AC16, CGW15].

$\mathsf{Game}_{5,d}$. If $\mathcal{A}$ follows the Type-1 strategy, this is the same as $\mathsf{Game}_{4,d}$. Otherwise, this is the same as $\mathsf{Game}_{4,d}$ except that $\mathcal{C}$ answers semi-functional $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$ upon $\mathcal{A}$'s decryption key reveal queries and homomorphic evaluation key reveal queries on y until the first critical homomorphic evaluation key reveal query. We note that $\mathcal{C}$ still uses normal $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$ to answer $\mathcal{A}$'s decryption queries and evaluation queries as in $\mathsf{Game}_{4,d}$.

Since $f(x^\star, y) = 0$ holds due to the definitions of the adaptive $\mathsf{KH\text{-}CCA}$ security and $\mathcal{A}$'s Type-2 strategy, we can prove $\mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{5,d}$ under the matrix DDH assumption over $\mathbb{G}_2$ by following the dual system technique of Π_{DSG} [AC16, CGW15].

Lemma 21 ($\mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{5,d}$). *If the PES satisfies the perfect security and the matrix DDH assumption over $\mathbb{G}_2$ holds, $\mathsf{Game}_{4,d}$ and $\mathsf{Game}_{5,d}$ are computationally indistinguishable for any PPT $\mathcal{A}$.*

We will prove Lemma 21 in Section 8.3.4.

$\mathsf{Game}_{6,d}$. This is the same as $\mathsf{Game}_{5,d}$ except $\mathcal{C}$'s answer to the challenge query if $d = 1$ and a $(d-1)$-th dependent evaluation query if $d \in [2, D]$ by setting $\mathsf{ct}^{[d]}_T \leftarrow_R \mathbb{G}_T$. Since the d-th ciphertext $\mathsf{ct}^{[d]}_{x^\star} \in \mathcal{L}$ is independent of $\mu^{\star}_{\mathsf{coin}}$ in $\mathsf{Game}_{6,d}$, $\mathcal{A}$'s advantage in $\mathsf{Game}_{6,D}$ is exactly 0.

Lemma 22 ($\mathsf{Game}_{5,d} \approx \mathsf{Game}_{6,d}$). *It holds that*

$$|\Pr[E_{5,d} - \Pr[E_{6,d}]]| \leq \mathsf{negl}(\lambda)$$

with overwhelming probability.

We will prove Lemma 22 at the end of the proof.

$\mathsf{Game}_{7,d}$. If $\mathcal{A}$ follows the Type-1 strategy, this is the same as $\mathsf{Game}_{6,d}$. Otherwise, this is the same as $\mathsf{Game}_{6,d}$ except that $\mathcal{C}$ always answers normal $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$ upon $\mathcal{A}$'s decryption key reveal queries and homomorphic evaluation key reveal queries on y.

By following the proof of $\mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{5,d}$ (Lemma 21), $\mathsf{Game}_{6,d} \approx_c \mathsf{Game}_{7,d}$ holds under the matrix DDH assumption over $\mathbb{G}_2$.

Lemma 23 ($\mathsf{Game}_{6,d} \approx_c \mathsf{Game}_{7,d}$). *If the* PES *satisfies the perfect security and the matrix DDH assumption over* $\mathbb{G}_2$ *holds,* $\mathsf{Game}_{6,d}$ *and* $\mathsf{Game}_{7,d}$ *are computationally indistinguishable for any PPT* $\mathcal{A}$.

We will prove Lemma 23 in Section 8.3.4.

$\mathsf{Game}_{8,d}$. This is the same as $\mathsf{Game}_{7,d}$ except that $\mathcal{C}$ always answers normal $\mathsf{sk}_{y,0}$ upon $\mathcal{A}$'s decryption key reveal queries on y.

By following the proof of $\mathsf{Game}_{3,d} \approx_c \mathsf{Game}_{4,d}$ (Lemma 20), $\mathsf{Game}_{7,d} \approx_c \mathsf{Game}_{8,d}$ holds under the matrix DDH assumption over $\mathbb{G}_2$.

Lemma 24 ($\mathsf{Game}_{7,d} \approx_c \mathsf{Game}_{8,d}$). *If the* PES *satisfies the perfect security and the matrix DDH assumption over* $\mathbb{G}_2$ *holds,* $\mathsf{Game}_{7,d}$ *and* $\mathsf{Game}_{8,d}$ *are computationally indistinguishable for any PPT* $\mathcal{A}$.

We will prove Lemma 24 in Section 8.3.4.

$\mathsf{Game}_{9,d}$. This is the same as $\mathsf{Game}_{8,d}$ except $\mathcal{C}$'s answer to the challenge query if $d = 1$ and a dependent evaluation query if $d \in [2, D]$. In particular, $\mathcal{C}$ sets $\mathsf{ct}_{x^\star}^{[d]}$ as a normal encryption of a random string $\mu^{[d]} \leftarrow_R \mathbb{G}_T$.

By following the proof of $\mathsf{Game}_{9,d-1} \approx_c \mathsf{Game}_{3,d}$ (Lemma 19), $\mathsf{Game}_{8,d} \approx_c \mathsf{Game}_{9,d}$ holds under the matrix DDH assumption over $\mathbb{G}_1$.

Lemma 25 ($\mathsf{Game}_{8,d} \approx_c \mathsf{Game}_{9,d}$). *If the matrix DDH assumption over* $\mathbb{G}_1$ *holds,* $\mathsf{Game}_{8,d}$ *and* $\mathsf{Game}_{9,d}$ *are computationally indistinguishable for any PPT* $\mathcal{A}$.

We will prove Lemma 25 in Section 8.3.3.

To conclude the proof of Theorem 15, we prove Lemma 22.

Proof of Lemma 22. We prove only for $d = 1$ since proofs for the other cases are essentially the same. For this purpose, we construct a simulator that behaves as $\mathcal{C}$ in $\mathsf{Game}_{5,d}$ from $\mathcal{A}$'s view. The simulator runs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda)$ and $n \leftarrow \mathsf{Param}(\mathsf{par})$, and choose a collision-resistant hash function $H \leftarrow_R \mathcal{H}$, where $H : \{0,1\}^* \to \mathbb{Z}_p$. The simulator samples $(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, uniformly random matrices $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$, random vectors $(\mathbf{u}_\iota)_{\iota \in [0,2]} \leftarrow_R \mathbb{Z}_p^{k+1}$, and random $\tilde{\alpha}_0 \leftarrow_R \mathbb{Z}_p$, then sets $\mathbf{u}_0 = \widehat{\mathbf{u}}_0 + \tilde{\alpha}_0 \mathbf{a}^\perp$. Nevertheless, the simulator does not use $\mathbf{u}_0$ but $\widehat{\mathbf{u}}_0$ to simulate the game except for creating $\mathsf{ct}^{[d]}_{x^\star} \in \mathcal{L}$. At first, the simulator sends $\mathsf{mpk} = (\mathcal{G}(1^\lambda), [\mathbf{A}]_1, [\mathbf{B}]_2, ([\mathbf{W}_j^\top \mathbf{A}]_1, [\mathbf{W}_j\mathbf{B}]_2)_{j\in[n]}, ([\mathbf{A}^\top \mathbf{u}_\iota]_T)_{\iota\in[0,2]}, H)$ to $\mathcal{A}$. mpk is properly distributed since it holds that

$$[\mathbf{A}^\top \widehat{\mathbf{u}}_0] = [\mathbf{A}^\top(\mathbf{u}_0 - \tilde{\alpha}_0 \mathbf{a}^\perp)] = [\mathbf{A}^\top \mathbf{u}_0] \cdot [\mathbf{A}^\top \mathbf{a}^\perp]^{-\tilde{\alpha}_0} = [\mathbf{A}^\top \mathbf{u}_0]. \tag{16}$$

The simulator answers $\mathcal{A}$'s homomorphic evaluation key reveal queries and evaluation queries by using $\mathbf{u}_1, \mathbf{u}_2$ as in $\mathsf{Game}_{5,d}$, while it answers $\mathcal{A}$'s decryption key reveal queries and decryption queries by using $\mathbf{u}_1, \mathbf{u}_2$ and $\widehat{\mathbf{u}}_0$. We will discuss the validity later.

Upon $\mathcal{A}$'s challenge query on $(x^\star, \mu_0^\star, \mu_1^\star)$, the simulator samples $\mathsf{coin} \leftarrow_R \{0,1\}$ and creates the challenge ciphertext $\mathsf{ct}^\star_{x^\star} = ((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}^\star_{1,t})_{t\in[w_3]}, \mathsf{ct}^\star_T, \pi^\star)$ in the same way as in $\mathsf{Game}_{5,d}$;

$$\mathsf{ct}^\star_{0,i} = [\mathbf{c}_i]_1, \qquad \mathsf{ct}^\star_{1,t} = \prod_{i\in[w_2]} [\mathbf{c}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1], j\in[n]} [\mathbf{W}_j^\top \mathbf{c}_i]_1^{\eta_{t,i,j}},$$
$$\mathsf{ct}^\star_T = \mu^\star_{\mathsf{coin}} \cdot [\mathbf{c}_0^\top \mathbf{u}_0]_T, \qquad \pi^\star = [\mathbf{c}_0^\top(\mathbf{u}_1 + \tilde{h} \cdot \mathbf{u}_2)]_T,$$

where $h^\star = H((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, \mathsf{ct}^\star_T)$. Observe that $\mathsf{ct}^\star_T$ is the only element which the simulator uses $\mathbf{u}_0$ to create and

$$\mathsf{ct}^\star_T = \mu^\star_{\mathsf{coin}} \cdot [\mathbf{c}_0^\top(\widehat{\mathbf{u}}_0 + \tilde{\alpha}_0 \mathbf{a}^\perp)]_T = \mu^\star_{\mathsf{coin}} \cdot [\mathbf{c}_0^\top \widehat{\mathbf{u}}_0]_T \cdot [\mathbf{c}_0^\top \mathbf{a}^\perp]_T^{\tilde{\alpha}_0}$$

holds. Since $[\mathbf{c}^\top \mathbf{a}^\perp]$ is a generator of $\mathbb{G}$ with overwhelming probability and $\mathsf{ct}^\star_T$ is the only element which depends on $\tilde{\alpha}_0$ in the security game, $\mathsf{ct}^\star_T$ is distributed uniformly at random over $\mathbb{G}_T$ as in $\mathsf{Game}_{6,d}$.

Finally, we check that the simulator's answers to decryption key reveal queries and decryption queries are valid although $\widehat{\mathbf{u}}_0 \neq \mathbf{u}_0$ is used. The modification in $\mathsf{Game}_{4,d}$ ensures that all $\mathsf{sk}_{0,\iota} = ((\mathsf{sk}_{0,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{0,1,t'})_{t'\in[m_3]})$ which $\mathcal{A}$ receives follow semi-functional distributions. Then, it holds that

$$\begin{aligned}\mathsf{sk}_{0,1,t'} &= [\widehat{\mathbf{u}}_0 + \alpha_{0,y}\mathbf{a}^\perp]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{\iota,m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2], j\in[n]} [\mathbf{W}_j \mathbf{B}\mathbf{r}_{\iota,i'}]_2^{\phi_{t',i',j}} \\ &= [\mathbf{u}_0 + (\alpha_{0,y} - \tilde{\alpha}_0)\mathbf{a}^\perp]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{\iota,m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2], j\in[n]} [\mathbf{W}_j \mathbf{B}\mathbf{r}_{\iota,i'}]_2^{\phi_{t',i',j}},\end{aligned}$$

where $\alpha_{0,y} - \tilde{\alpha}_0$ is distributed uniformly at random over $\mathbb{Z}_p$ as in $\mathsf{Game}_{5,d}$ due to the randomness of $\alpha_{0,y}$. We check the validities of decryption queries depending on whether $\mathcal{A}$ follows Type-1 or Type-2.

Case of Type-1. Since $\mathcal{A}$ of Type-1 makes a critical homomorphic evaluation key reveal query in Phase 1, it is allowed to make decryption queries only in Phase 1. Upon $\mathcal{A}$'s decryption query on $\mathsf{ct}_x = ((\mathsf{ct}_{0,i} = [\mathbf{c}'_i]_1)_{i\in[0,w_1]}, (\mathsf{ct}_{1,t})_{t\in[w_3]}, \mathsf{ct}_T, \pi)$, the simulator's answer is valid when $\mathbf{c}_0'^\top \mathbf{u}_0 = \mathbf{c}_0'^\top \widehat{\mathbf{u}}_0$ holds. Thus, the answer is invalid only when $\mathbf{c}'_0$ does not live in the span of $\mathbf{A}$ and the answer is not $\perp$. In other words, the simulator cannot answer $\mathcal{A}$'s critical decryption queries in a valid way. Since the dual system technique of Π_{DSG} [AC16, CGW15] implies that $\mathcal{A}$ cannot create semi-functional ciphertexts by itself, the only way for $\mathcal{A}$ to create semi-functional ciphertexts is evaluating the challenge ciphertext $\mathsf{ct}^\star_{x^\star}$. Thus, $\mathcal{A}$ of Type-1 which is allowed to make decryption queries only in Phase 1 cannot make critical decryption queries. Thus, $\mathsf{Game}_{5,d} \approx \mathsf{Game}_{6,d}$ holds.

Case of Type-2. Since $\mathcal{A}$ of Type-2 does not make a critical homomorphic evaluation key reveal query in Phase 1, it is allowed to make decryption queries until it makes the first critical homomorphic evaluation key reveal query in Phase 2. When the computationally unbounded $\mathcal{A}$ receives

mpk, it can compute $\widehat{\mathbf{u}}_\iota$ for $\iota \in [2]$ such that $\mathbf{A}^\top \mathbf{u}_\iota = \mathbf{A}^\top \widehat{\mathbf{u}}_\iota$, where $\mathbf{u}_\iota = \widehat{\mathbf{u}}_\iota + \tilde{\alpha}_\iota \mathbf{a}^\perp$. Since the modification in $\mathsf{Game}_{5,d}$ ensures that all $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$ which $\mathcal{A}$ of Type-2 receives follow semi-functional distributions, $\tilde{\alpha}_1$ and $\tilde{\alpha}_2$ are distributed uniformly at random over $\mathbb{Z}_p$ from $\mathcal{A}$'s view. When the computationally unbounded $\mathcal{A}$ receives the challenge ciphertext $\mathsf{ct}^\star_{x^\star}$, it learns the value of $\tilde{\alpha}_1 + h^\star \cdot \tilde{\alpha}_2$ since it holds that

$$\pi^\star = [\mathbf{c}_0^\top (\widehat{\mathbf{u}}_1 + \tilde{\alpha}_1 \mathbf{a}^\perp) + h^\star \cdot (\widehat{\mathbf{u}}_2 + \tilde{\alpha}_2 \mathbf{a}^\perp)]_T = [\mathbf{c}_0^\top \widehat{\mathbf{u}}_1 + h^\star \cdot \widehat{\mathbf{u}}_2] \cdot [\mathbf{c}_0^\top \mathbf{a}^\perp]_T^{\tilde{\alpha}_1 + h^\star \cdot \tilde{\alpha}_2}.$$

If the answer to $\mathcal{A}$'s decryption query on $\mathsf{ct}_x = ((\mathsf{ct}_{0,i} = [\mathbf{c}'_i]_1)_{i \in [0,w_1]}, (\mathsf{ct}_{1,t})_{t \in [w_3]}, \mathsf{ct}_T, \pi)$ is not $\perp$, $\pi = [\mathbf{c}'^\top_0 (\mathbf{u}_1 + h \cdot \mathbf{u}_2)]_1$ holds due to the condition (14). If $\mathbf{c}'_0$ does not live in the span of $\mathbf{A}$, $\mathcal{A}$ learns the value of $\tilde{\alpha}_1 + h \cdot \tilde{\alpha}_2$, where the change in Game_1 ensures that $h \neq h^\star$ holds. Then, a computationally unbounded $\mathcal{A}$'s ability to make a critical decryption query is equivalent to the knowledge of $(\tilde{\alpha}_1, \tilde{\alpha}_2) \in \mathbb{Z}_p^2$. $\mathcal{A}$ cannot learn $\tilde{\alpha}_1 + h \cdot \tilde{\alpha}_2$ for any h from answers to dependent evaluation queries since the change in Game_2 ensures that the discrete logarithm of $\mathsf{ct}^{[d]}_{0,0}$ lives in the span of $\mathbf{A}$. (If $d \in [2, D]$, the change in $\mathsf{Game}_{5,d-1}$ is also required to ensure the fact.) Although $\mathcal{A}$ of Type-2 can learn $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ when it makes the first critical homomorphic evaluation key reveal query in Phase 2, it is not allowed to make decryption queries after the query. The only way for $\mathcal{A}$ to learn $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ is making decryption queries and evaluation queries such that $\mathbf{c}'_0$ does not live in the span of $\mathbf{A}$. Although $\mathcal{A}$ can eliminate a candidate of $\tilde{\alpha}_1 + h \cdot \tilde{\alpha}_2$ for some h by making a decryption query or an evaluation query and the answer is $\perp$, there are exponentially many candidates and $\mathcal{A}$ is allowed to make only polynomial number of queries. Thus, $\mathsf{Game}_{5,d} \approx \mathsf{Game}_{6,d}$ holds with probability $1 - (Q_{\mathsf{Dec}} + Q_{\mathsf{Eval}})/q$, where Q_{Dec} (resp. Q_{Eval}) denotes the number of $\mathbf{A}$'s decryption (resp. evaluation) queries. □

□

8.3.3 Ciphertext Indistinguishability

We prove Lemmata 19 and 25.

Proof of Lemma 19. We show that for any PPT adversary $\mathcal{A}$ that breaks the adaptive $\mathsf{KH\text{-}CCA}$ security of Π_{ABKHE}, there exists a reduction algorithm $\mathcal{B}_1$ that solves the $(w_1 + w_2)$-fold matrix DDH assumption over $\mathbb{G}_1$, where

$$|\Pr[E_{9,d-1}] - \Pr[E_{3,d}]| \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{mDDH}_{\mathbb{G}_1}}(\lambda). \tag{17}$$

We prove only for $d = 1$ since proofs for the other cases are essentially the same. $\mathcal{B}_1$ receives $(\mathcal{G}(1^\lambda), [\mathbf{A}]_1, [\mathbf{V}]_1)$ which is an instance of the $(w_1 + w_2)$-fold matrix DDH problem over $\mathbb{G}_1$, where $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$, $\mathbf{V} = \mathbf{AS}$ for $\mathbf{S} \leftarrow_R \mathbb{Z}_p^{k \times (w_1 + w_2)}$ or $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1) \times (w_1 + w_2)}$. $\mathcal{B}_1$ chooses a collision-resistant hash function $H \leftarrow_R \mathcal{H}$, samples $(\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, random matrices $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_R \mathbb{Z}_p^{(k+1) \times (k+1)}$, and random vectors $(\mathbf{u}_\iota)_{\iota \in [0,2]} \leftarrow_R \mathbb{Z}_p^{k+1}$, then sends $\mathsf{mpk} = \Big(\mathcal{G}(1^\lambda), [\mathbf{A}]_1, [\mathbf{B}]_2, ([\mathbf{W}_j^\top \mathbf{A}]_1, [\mathbf{W}_j \mathbf{B}]_2)_{j \in [n]}, ([\mathbf{A}^\top \mathbf{u}_\iota]_T)_{\iota \in [0,2]}, H\Big)$ to $\mathcal{A}$. Since $\mathcal{B}_1$ knows $(\mathbf{u}_\iota)_{\iota \in [0,2]}$, it can answer all $\mathcal{A}$'s decryption key reveal queries, homomorphic evaluation key reveal queries, decryption queries, and evaluation queries by creating normal $\mathsf{sk}_{y,0}, \mathsf{sk}_{y,1}$, and $\mathsf{sk}_{y,2}$.

Upon $\mathcal{A}$'s challenge query on $(x^\star, \mu_0^\star, \mu_1^\star)$, $\mathcal{B}_1$ samples $\mathsf{coin} \leftarrow_R \{0,1\}$ and creates $\mathsf{ct}^\star_{x^\star} = ((\mathsf{ct}^\star_{0,i})_{i \in [0,w_1]}, (\mathsf{ct}^\star_{1,t})_{t \in [w_3]}, \mathsf{ct}^\star_T, \pi^\star)$;

$$\begin{aligned} &\mathsf{ct}^\star_{0,i} = [\mathbf{v}_i]_1, \qquad \mathsf{ct}^\star_{1,t} = \prod_{i \in [w_2]} [\mathbf{v}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i \in [0,w_1], j \in [n]} [\mathbf{W}_j^\top \mathbf{v}_i]_1^{\eta_{t,i,j}}, \\ &\mathsf{ct}^\star_T = \mu \cdot [\mathbf{v}_0^\top \mathbf{u}_0]_T, \quad \pi = [\mathbf{v}_0^\top (\mathbf{u}_1 + h \cdot \mathbf{u}_2)]_T, \end{aligned} \tag{18}$$

where $h^\star = H((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, \mathsf{ct}^\star_T)$ and $\mathbf{v}_i$ is an i-th column vector of $\mathbf{V}$. The challenge ciphertext $\mathsf{ct}^\star_{x^\star}$ is distributed as in $\mathsf{Game}_{9,0}$ (resp. $\mathsf{Game}_{3,1}$) if $\mathbf{V} = \mathbf{AS}$ (resp. $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(w_1+w_2)}$). Thus, the inequality (17) holds. $\square$

Proof of Lemma 25. We can show that for any PPT adversary $\mathcal{A}$ that breaks the adaptive $\mathsf{KH\text{-}CCA}$ security of Π_{ABKHE}, there exists a reduction algorithm $\mathcal{B}_9$ that solves the $(w_1 + w_2)$-fold matrix DDH assumption over $\mathbb{G}_1$, where

$$|\Pr[E_{8,d}] - \Pr[E_{9,d}]| \leq \mathsf{Adv}^{\mathsf{mDDH}_{\mathbb{G}_1}}_{\mathcal{B}_9}(\lambda). \tag{19}$$

The proof is almost the same as the proof of Lemma 19. After $\mathcal{B}_9$ receives $(\mathcal{G}(1^\lambda), [\mathbf{A}]_1, [\mathbf{V}]_1)$, it sends mpk to $\mathcal{A}$ in the same way as $\mathcal{B}_1$. $\mathcal{B}_9$ answers all $\mathcal{A}$'s decryption key reveal queries, homomorphic evaluation key reveal queries, decryption queries, and evaluation queries in the same way as $\mathcal{B}_1$. Although $\mathcal{B}_9$ cannot create semi-functional $\mathsf{sk}_{y,0}, \mathsf{sk}_{y,1}$, and $\mathsf{sk}_{y,2}$ since it does not know $\mathbf{a}^\perp$, normal $\mathsf{sk}_{y,0}, \mathsf{sk}_{y,1}$, and $\mathsf{sk}_{y,2}$ are sufficient for answering the queries due to the changes in $\mathsf{Game}_{7,d}$ and $\mathsf{Game}_{8,d}$. If $d = 1$, $\mathcal{B}_9$ answers $\mathcal{A}$'s challenge query in the same way as (18) except $\mathsf{ct}^\star_T \leftarrow_R \mathbb{G}_T$. The challenge ciphertext $\mathsf{ct}^\star_{x^\star}$ is distributed as in $\mathsf{Game}_{9,1}$ (resp. $\mathsf{Game}_{8,1}$) if $\mathbf{V} = \mathbf{AS}$ (resp. $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(w_1+w_2)}$). Thus, the inequality (19) holds. $\square$

8.3.4 Key Indistinguishability

We prove Lemmata 20, 21, 23, and 24. For this purpose, we use the following auxiliary distributions for ABE secret keys $\mathsf{sk}_{y,\iota}$.

Pseudo-normal Secret Key. An ι-th semi-functional secret key $\mathsf{sk}_{y,\iota}$ for y is defined as $\mathsf{sk}_{y,\iota} = ((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]})$;

$$\mathsf{sk}_{\iota,0,i'} = [\mathbf{d}_{\iota,i'}]_2, \qquad \mathsf{sk}_{\iota,1,t'} = [\mathbf{u}_\iota]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{d}_{\iota,m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2], j\in[n]} [\mathbf{W}_j \mathbf{d}_{\iota,i'}]_2^{\phi_{t',i',j}},$$

where $\mathbf{d}_{\iota,1}, \ldots, \mathbf{d}_{\iota,m_1+m_2} \leftarrow_R \mathbb{Z}_p^{k+1}$.

Pseudo-SF Secret Key. An ι-th semi-functional secret key $\mathsf{sk}_{y,\iota}$ for y is defined as $\mathsf{sk}_{y,\iota} = ((\mathsf{sk}_{\iota,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{\iota,1,t'})_{t'\in[m_3]})$;

$$\mathsf{sk}_{\iota,0,i'} = [\mathbf{d}_{\iota,i'}]_2, \qquad \mathsf{sk}_{\iota,1,t'} = [\mathbf{u}_\iota + \alpha_{\iota,y}\mathbf{a}^\perp]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{d}_{\iota,m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2], j\in[n]} [\mathbf{W}_j \mathbf{d}_{\iota,i'}]_2^{\phi_{t',i',j}},$$

where $\mathbf{d}_{\iota,1}, \ldots, \mathbf{d}_{\iota,m_1+m_2} \leftarrow_R \mathbb{Z}_p^{k+1}$ and $\alpha_{\iota,y} \leftarrow_R \mathbb{Z}_p$.

Proof of Lemma 20. We use the following games $\mathsf{Game}_{3,d,\zeta,1}$, $\mathsf{Game}_{3,d,\zeta,2}$, and $\mathsf{Game}_{3,d,\zeta,3}$ for $\zeta \in [Q_{\mathsf{dk}}]$, where Q_{dk} denotes the number of $\mathcal{A}$'s decryption key reveal queries, $\mathsf{Game}_{3,d,0,3} = \mathsf{Game}_{3,d}$, and $\mathsf{Game}_{3,d,Q_{\mathsf{dk}},3} = \mathsf{Game}_{4,d}$.

$\mathsf{Game}_{3,d,\zeta,1}$. This is the same as $\mathsf{Game}_{3,d,\zeta-1,3}$ except that $\mathcal{C}$ answers pseudo-normal $\mathsf{sk}_{y,0}$ upon $\mathcal{A}$'s ζ-th decryption key reveal queries on y.

$\mathsf{Game}_{3,d,\zeta,2}$. This is the same as $\mathsf{Game}_{3,d,\zeta,1}$ except that $\mathcal{C}$ answers pseudo-SF $\mathsf{sk}_{y,0}$ upon $\mathcal{A}$'s ζ-th decryption key reveal queries on y.

$\mathsf{Game}_{3,d,\zeta,3}$. This is the same as $\mathsf{Game}_{3,d,\zeta,2}$ except that $\mathcal{C}$ answers semi-functional $\mathsf{sk}_{y,0}$ upon $\mathcal{A}$'s ζ-th decryption key reveal queries on y.

Table 4: Distributions of ABE secret keys $\mathsf{sk}_{y,0}$ to answer $\mathcal{A}$'s decryption key reveal queries in $\mathsf{Game}_{3,d,\zeta,1}$, $\mathsf{Game}_{3,d,\zeta,2}$, and $\mathsf{Game}_{3,d,\zeta,3}$

	before ζ-th query	ζ-th query	after ζ-th query
$\mathsf{Game}_{3,d,\zeta,1}$	semi-functional	pseudo-normal	normal
$\mathsf{Game}_{3,d,\zeta,2}$	semi-functional	pseudo-SF	normal
$\mathsf{Game}_{3,d,\zeta,3}$	semi-functional	semi-functional	normal

Table 4 summarizes distributions of $\mathsf{sk}_{y,0}$ in each game. To prove $\mathsf{Game}_{3,d} \approx_c \mathsf{Game}_{4,d}$, we show that $\mathsf{Game}_{3,d,\zeta-1,3} \approx_c \mathsf{Game}_{3,d,\zeta,1} \equiv \mathsf{Game}_{3,d,\zeta,2} \approx_c \mathsf{Game}_{3,d,\zeta,3}$.

Lemma 26 ($\mathsf{Game}_{3,d,\zeta-1,3} \approx_c \mathsf{Game}_{3,d,\zeta,1}$)**.** *If the matrix DDH assumption over $\mathbb{G}_2$ holds, $\mathsf{Game}_{3,d,\zeta-1,3}$ and $\mathsf{Game}_{3,d,\zeta,1}$ are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 26. We prove only for $d = 1$ since proofs for the other cases are essentially the same. We show that for any PPT adversary $\mathcal{A}$ that breaks the adaptive $\mathsf{KH\text{-}CCA}$ security of Π_{ABKHE}, there exists a reduction algorithm $\mathcal{B}_{3,1}$ that solves the (m_1+m_2)-fold matrix DDH assumption over $\mathbb{G}_2$, where

$$|\Pr[E_{3,d,\zeta-1,3}] - \Pr[E_{3,d,\zeta,1}]| \leq \mathsf{Adv}_{\mathcal{B}_{3,1}}^{\mathsf{mDDH}_{\mathbb{G}_2}}(\lambda). \tag{20}$$

$\mathcal{B}_{3,1}$ receives $(\mathcal{G}(1^\lambda), [\mathbf{B}]_2, [\mathbf{V}]_2)$ which is an instance of the (m_1+m_2)-fold matrix DDH problem over $\mathbb{G}_2$, where $(\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, $\mathbf{V} = \mathbf{BR}$ for $\mathbf{R} \leftarrow_R \mathbb{Z}_p^{k\times(m_1+m_2)}$ or $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(m_1+m_2)}$. $\mathcal{B}_{3,1}$ chooses a collision-resistant hash function $H \leftarrow_R \mathcal{H}$, samples $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$, random matrices $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$, and random vectors $(\mathbf{u}_\iota)_{\iota\in[0,2]} \leftarrow_R \mathbb{Z}_p^{k+1}$, then sends $\mathsf{mpk} = (\mathcal{G}(1^\lambda), [\mathbf{A}]_1, [\mathbf{B}]_2, ([\mathbf{W}_j^\top \mathbf{A}]_1, [\mathbf{W}_j\mathbf{B}]_2)_{j\in[n]}, ([\mathbf{A}^\top \mathbf{u}_\iota]_T)_{\iota\in[0,2]}, H)$ to $\mathcal{A}$. Since $\mathcal{B}_{3,1}$ knows $(\mathbf{u}_\iota)_{\iota\in[2]}$, it can answer all $\mathcal{A}$'s homomorphic evaluation key reveal queries and evaluation queries by creating normal $\mathsf{sk}_{y,1}$, and $\mathsf{sk}_{y,2}$. Since $\mathcal{B}_{3,1}$ knows $(\mathbf{u}_\iota)_{\iota\in[0,2]}$, it can answer all $\mathcal{A}$'s decryption key reveal queries after the ζ-th query and decryption queries by creating normal $\mathsf{sk}_{y,0}$, $\mathsf{sk}_{y,1}$, and $\mathsf{sk}_{y,2}$. Since $\mathcal{B}_{3,1}$ knows $(\mathbf{u}_\iota)_{\iota\in[0,2]}$ and $\mathbf{a}^\perp$, it can answer all $\mathcal{A}$'s decryption key reveal queries before the ζ-th query by creating semi-functional $\mathsf{sk}_{y,0}$ and normal $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$. Since $\mathcal{B}_{3,1}$ knows $(\mathbf{u}_\iota)_{\iota\in[0,2]}$ and $\mathbf{W}_1, \ldots, \mathbf{W}_n$, it can answer $\mathcal{A}$'s challenge query by creating semi-functional $\mathsf{ct}^\star_{x^\star}$.

Upon $\mathcal{A}$'s ζ-th decryption key reveal query on y, $\mathcal{B}_{3,1}$ creates normal $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$, and $\mathsf{sk}_{y,0} = ((\mathsf{sk}_{0,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{0,1,t'})_{t'\in[m_3]})$;

$$\mathsf{sk}_{0,0,i'} = [\mathbf{v}_{i'}]_2, \qquad \mathsf{sk}_{0,1,t'} = [\mathbf{u}_0]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{v}_{m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\mathbf{W}_j\mathbf{v}_{i'}]_2^{\phi_{t',i',j}}, \tag{21}$$

where $\mathbf{v}_i$ is an i-th column vector of $\mathbf{V}$. The ζ-th $\mathsf{sk}_{y,0}$ is distributed as in $\mathsf{Game}_{3,d,\zeta-1,3}$ (resp. $\mathsf{Game}_{3,d,\zeta,1}$) if $\mathbf{V} = \mathbf{BR}$ (resp. $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(m_1+m_2)}$). Thus, the inequality (20) holds. □

Lemma 27 ($\mathsf{Game}_{3,d,\zeta,1} \equiv \mathsf{Game}_{3,d,\zeta,2}$)**.** *If the* PES *satisfies the perfect security,* $\mathsf{Game}_{3,d,\zeta,1}$ *and* $\mathsf{Game}_{3,d,\zeta,2}$ *follow the same distribution from $\mathcal{A}$'s view.*

Proof of Lemma 27. We prove only for $d = 1$ since proofs for the other cases are essentially the same. In $\mathsf{Game}_{3,1,\zeta,1}$, the challenge ciphertext $\mathsf{ct}^\star_{x^\star} = ((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, (\mathsf{ct}^\star_{1,t})_{t\in[w_3]}, \mathsf{ct}^\star_T, \pi^\star)$ is semi-functional;

$$\mathsf{ct}^\star_{0,i} = [\mathbf{c}_i]_1, \qquad \mathsf{ct}^\star_{1,t} = \prod_{i\in[w_2]} [\mathbf{c}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1],j\in[n]} [\mathbf{W}_j^\top \mathbf{c}_i]_1^{\eta_{t,i,j}},$$

$$\mathsf{ct}^\star_T = \mu \cdot [\mathbf{c}_0^\top \mathbf{u}_0]_T, \qquad \pi^\star = [\mathbf{c}_0^\top(\mathbf{u}_1 + h\cdot \mathbf{u}_2)]_T,$$

where $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{w_1+w_2} \leftarrow_R \mathbb{Z}_p^{k+1}$ and $h = H((\mathsf{ct}^\star_{0,i})_{i\in[0,w_1]}, \mathsf{ct}^\star_T)$. Due to the basis lemma (Lemma 18), the distribution is identical to

$$\mathsf{ct}^\star_{0,i} = [\mathbf{A}\mathbf{s}_i + s_i\mathbf{b}^\perp]_1, \qquad \mathsf{ct}^\star_{1,t} = \prod_{i\in[w_2]} [\mathbf{A}\mathbf{s}_{w_1+i} + \hat{s}_i\mathbf{b}^\perp]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1],j\in[n]} [\mathbf{W}_j^\top(\mathbf{A}\mathbf{s}_i + s_i\mathbf{b}^\perp)]_1^{\eta_{t,i,j}},$$
$$\mathsf{ct}^\star_T = \mu\cdot[(\mathbf{A}\mathbf{s}_0 + s_0\mathbf{b}^\perp)^\top\mathbf{u}_0]_T, \qquad \pi^\star = [(\mathbf{A}\mathbf{s}_0 + s_0\mathbf{b}^\perp)^\top(\mathbf{u}_1 + h\cdot\mathbf{u}_2)]_T,$$

with overwhelming probability, where $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{w_1+w_2} \leftarrow_R \mathbb{Z}_p^k$ and $s_0, s_1, \ldots, s_{w_1}, \hat{s}_1, \ldots, \hat{s}_{w_2} \leftarrow_R \mathbb{Z}_p$. Similarly, the distribution of ζ-th pseudo-normal $\mathsf{sk}_{y,0} = ((\mathsf{sk}_{0,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{0,1,t'})_{t'\in[m_3]})$ in $\mathsf{Game}_{3,1,\zeta,1}$ is identical to

$$\begin{aligned}\mathsf{sk}_{0,0,i'} &= [\mathbf{B}\mathbf{r}_{i'} + r_{i'}\mathbf{a}^\perp]_2,\\ \mathsf{sk}_{0,1,t'} &= [\mathbf{u}_0]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{m_1+i'} + \hat{r}_{i'}\mathbf{a}^\perp]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\mathbf{W}_j(\mathbf{B}\mathbf{r}_{i'} + r_{i'}\mathbf{a}^\perp)]_2^{\phi_{t',i',j}},\end{aligned}$$

with overwhelming probability, where $\mathbf{r}_1, \ldots, \mathbf{r}_{m_1+m_2} \leftarrow_R \mathbb{Z}_p^k$ and $r_1, \ldots, r_{m_1}, \hat{r}_1, \ldots, \hat{r}_{m_2} \leftarrow_R \mathbb{Z}_p$. In $\mathsf{Game}_{3,1,\zeta,1}$, each $\mathbf{W}_1, \ldots, \mathbf{W}_n$ is sampled according to $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$. The distribution is identical to

$$\mathbf{W}_1 = \widetilde{\mathbf{W}}_1 + b_1({\mathbf{a}^\perp}^\top\mathbf{b}^\perp)^{-1}\mathbf{a}{\mathbf{b}^\perp}^\top, \ldots, \mathbf{W}_n = \widetilde{\mathbf{W}}_n + b_n({\mathbf{a}^\perp}^\top\mathbf{b}^\perp)^{-1}\mathbf{a}{\mathbf{b}^\perp}^\top$$

where $\widetilde{\mathbf{W}}_1, \ldots, \widetilde{\mathbf{W}}_n \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$ and $b_1, \ldots, b_n \leftarrow_R \mathbb{Z}_p$. Since it holds that

$$\begin{aligned}\mathbf{W}_j^\top\mathbf{A} &= \widetilde{\mathbf{W}}_j^\top\mathbf{A} + b_j({\mathbf{a}^\perp}^\top\mathbf{b}^\perp)^{-1}\mathbf{b}^\perp{\mathbf{a}^\perp}^\top\mathbf{A} = \widetilde{\mathbf{W}}_j^\top\mathbf{A},\\ \mathbf{W}_j\mathbf{B} &= \widetilde{\mathbf{W}}_j\mathbf{B} + b_j({\mathbf{a}^\perp}^\top\mathbf{b}^\perp)^{-1}\mathbf{a}^\perp{\mathbf{b}^\perp}^\top\mathbf{B} = \widetilde{\mathbf{W}}_j\mathbf{B},\end{aligned}$$

mpk that contains $[\mathbf{W}_1^\top\mathbf{A}]_1, \ldots, [\mathbf{W}_n^\top\mathbf{A}]_1, [\mathbf{W}_1\mathbf{B}]_2, \ldots, [\mathbf{W}_n\mathbf{B}]_2$ does not depend on $b_1, \ldots, b_n$. Thus, the only elements that depend on $b_1, \ldots, b_n$ are $\mathsf{ct}^\star_{x^\star}$ and ζ-th pseudo-normal $\mathsf{sk}_{y,0}$. Since it holds that

$$\begin{aligned}\mathbf{W}_j^\top\mathbf{b}^\perp &= \widetilde{\mathbf{W}}_j^\top\mathbf{b}^\perp + b_j({\mathbf{a}^\perp}^\top\mathbf{b}^\perp)^{-1}\mathbf{b}^\perp({\mathbf{a}^\perp}^\top\mathbf{b}^\perp) = \widetilde{\mathbf{W}}_j^\top\mathbf{b}^\perp + b_j\mathbf{b}^\perp,\\ \mathbf{W}_j\mathbf{a}^\perp &= \widetilde{\mathbf{W}}_j\mathbf{a}^\perp + b_j({\mathbf{a}^\perp}^\top\mathbf{b}^\perp)^{-1}\mathbf{a}^\perp({\mathbf{b}^\perp}^\top\mathbf{a}^\perp) = \widetilde{\mathbf{W}}_j\mathbf{a}^\perp + b_j\mathbf{a}^\perp,\end{aligned}$$

we have

$$\begin{aligned}\mathsf{ct}^\star_{1,t} =& \prod_{i\in[w_2]}[\mathbf{A}\mathbf{s}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1],j\in[n]}[\widetilde{\mathbf{W}}_j^\top(\mathbf{A}\mathbf{s}_i + s_i\mathbf{b}^\perp)]_1^{\eta_{t,i,j}}\\ &\cdot [\mathbf{b}^\perp]_1^{\sum_{i\in[w_2]}\eta_{t,i}\hat{s}_i + \sum_{i\in[0,w_1],j\in[n]}\eta_{t,i,j}s_ib_j},\\ \mathsf{sk}_{0,1,t'} =& [\mathbf{u}_0]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]}[\mathbf{B}\mathbf{r}_{m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]}[\widetilde{\mathbf{W}}_j(\mathbf{B}\mathbf{r}_{i'} + r_{i'}\mathbf{a}^\perp)]_2^{\phi_{t',i',j}}\\ &\cdot [\mathbf{a}^\perp]_2^{\sum_{i'\in[m_2]}\phi_{t',i'}\hat{r}_{i'} + \sum_{i'\in[m_2],j\in[n]}\phi_{t',i',j}r_ib_j}.\end{aligned}$$

Observe that even when we do not know all of $(s_0, s_1, \ldots, s_{w_1}, \hat{s}_1, \ldots, \hat{s}_{w_2}, r_1, \ldots, r_{m_1}, \hat{r}_1, \ldots, \hat{r}_{m_2})$, $(s_0, s_1, \ldots, s_{w_1}, r_1, \ldots, r_{m_1}, (\sum_{i\in[w_2]}\eta_{t,i}\hat{s}_i + \sum_{i\in[0,w_1],j\in[n]}\eta_{t,i,j}s_ib_j)_{t\in[w_3]}, (\sum_{i'\in[m_2]}\phi_{t',i'}\hat{r}_{i'} +$

$\sum_{i'\in[m_1],j\in[n]} \phi_{t',i',j} r_{i'} b_j)_{t'\in[m_3]})$ are sufficient for simulating the semi-functional challenge ciphertext $\mathsf{ct}^\star_{x^\star}$ and ζ-th pseudo-normal $\mathsf{sk}_{y,0}$. Since it holds that $f(x^\star, y) = 0$ and all $s_0, s_1, \ldots, s_{w_1}, \hat{s}_1, \ldots, \hat{s}_{w_2}, r_1, \ldots, r_{m_1}, \hat{r}_1, \ldots, \hat{r}_{m_2}$ are sampled according to the uniform distribution over $\mathbb{Z}_p$, the perfect security of PES ensures that $\mathsf{ct}^\star_{x^\star}$ and ζ-th $\mathsf{sk}_{y,0}$ are identically distributed by simulating with $(s_0, \mathbf{s}, \mathbf{r}, (\sum_{i\in[w_2]} \eta_{t,i}\hat{s}_i + \sum_{i\in[0,w_1],j\in[n]} \eta_{t,i,j} s_i b_j)_{t\in[w_3]}, (\phi_{t'}\alpha_{0,y} + \sum_{i'\in[m_2]} \phi_{t',i'}\hat{r}_{i'} + \sum_{i'\in[m_1],j\in[n]} \phi_{t',i',j} r_{i'} b_j)_{t'\in[m_3]})$, where $\alpha_{0,y} \leftarrow_R \mathbb{Z}_p$. Then, we have

$$\begin{aligned}
\mathsf{sk}_{0,1,t'} &= [\mathbf{u}_0]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\widetilde{\mathbf{W}}_j(\mathbf{B}\mathbf{r}_{i'} + r_{i'}\mathbf{a}^\perp)]_2^{\phi_{t',i',j}} \\
&\quad \cdot [\mathbf{a}^\perp]_2^{\phi_{t'}\alpha_{0,y} + \sum_{i'\in[m_2]} \phi_{t',i'}\hat{r}_{i'} + \sum_{i'\in[m_2],j\in[n]} \phi_{t',i',j} r_i b_j} \\
&= [\mathbf{u}_0 + \alpha_{0,y}\mathbf{a}^\perp]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{m_1+i'} + \hat{r}_{i'}\mathbf{a}^\perp]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\mathbf{W}_j(\mathbf{B}\mathbf{r}_{i'} + r_{i'}\mathbf{a}^\perp)]_2^{\phi_{t',i',j}}.
\end{aligned}$$

Due to the basis lemma (Lemma 18), the distribution of ζ-th $\mathsf{sk}_{y,0}$ is identically distributed to pseudo-SF secret key. Thus, we complete the proof. $\square$

Lemma 28 ($\mathsf{Game}_{3,d,\zeta,2} \approx_c \mathsf{Game}_{3,d,\zeta,3}$). *If the matrix DDH assumption over $\mathbb{G}_2$ holds, $\mathsf{Game}_{3,d,\zeta,2}$ and $\mathsf{Game}_{3,d,\zeta,3}$ are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 28. We prove only for $d = 1$ since proofs for the other cases are essentially the same. We show that for any PPT adversary $\mathcal{A}$ that breaks the adaptive $\mathsf{KH\text{-}CCA}$ security of Π_{ABKHE}, there exists a reduction algorithm $\mathcal{B}_{3,3}$ that solves the $(m_1 + m_2)$-fold matrix DDH assumption over $\mathbb{G}_2$, where

$$|\Pr[E_{3,d,\zeta,2}] - \Pr[E_{3,d,\zeta,3}]| \le \mathsf{Adv}^{\mathsf{mDDH}_{\mathbb{G}_2}}_{\mathcal{B}_{3,3}}(\lambda). \tag{22}$$

The proof is almost the same as the proof of Lemma 26. After $\mathcal{B}_{3,3}$ receives $(\mathcal{G}(1^\lambda), [\mathbf{B}]_2, [\mathbf{V}]_2)$, it sends mpk to $\mathcal{A}$ in the same way as $\mathcal{B}_{3,1}$. $\mathcal{B}_{3,3}$ answers all $\mathcal{A}$'s decryption key reveal queries, homomorphic evaluation key reveal queries, decryption queries, evaluation queries, and challenge query in the same way as $\mathcal{B}_{3,1}$ except ζ-th $\mathsf{sk}_{y,0}$. $\mathcal{B}_{3,3}$ creates ζ-th $\mathsf{sk}_{y,0}$ in the same way as (21) except

$$\mathsf{sk}_{0,1,t'} = [\mathbf{u}_0 + \alpha_{0,y}\mathbf{a}^\perp]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{v}_{m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\mathbf{W}_j\mathbf{v}_{i'}]_2^{\phi_{t',i',j}},$$

where $\mathbf{v}_i$ is an i-th column vector of $\mathbf{V}$ and $\alpha_{0,y} \leftarrow_R \mathbb{Z}_p$. The ζ-th $\mathsf{sk}_{y,0}$ is distributed as in $\mathsf{Game}_{3,d,\zeta,3}$ (resp. $\mathsf{Game}_{3,d,\zeta,2}$) if $\mathbf{V} = \mathbf{B}\mathbf{R}$ (resp. $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(m_1+m_2)}$). Thus, the inequality (22) holds. $\square$

Based on Lemmata 26, 27, and 28, we have

$$|\Pr[E_3] - \Pr[E_4]| \le Q_{\mathsf{dk}}\Big(\mathsf{Adv}^{\mathsf{mDDH}_{\mathbb{G}_2}}_{\mathcal{B}_{3,1}}(\lambda) + \mathsf{Adv}^{\mathsf{mDDH}_{\mathbb{G}_2}}_{\mathcal{B}_{3,3}}(\lambda)\Big).$$

$\square$

Proof of Lemma 21. We use the following games $\mathsf{Game}_{4,d,\iota,\zeta,1}$, $\mathsf{Game}_{4,d,\iota,\zeta,2}$, and $\mathsf{Game}_{4,d,\iota,\zeta,3}$ for $\iota \in [2]$ and $\zeta \in [Q_{\mathsf{dk}} + Q_{\mathsf{hk}}]$, where Q_{dk} (resp. Q_{hk}) denotes the number of $\mathcal{A}$'s decryption key reveal queries (resp. homomorphic evaluation key reveal queries), $\mathsf{Game}_{4,d,1,0,3} = \mathsf{Game}_{4,d}$, $\mathsf{Game}_{4,d,2,0,3} = \mathsf{Game}_{4,d,1,Q_{\mathsf{dk}}+Q_{\mathsf{hk}},3}$, and $\mathsf{Game}_{4,d,2,Q_{\mathsf{dk}}+Q_{\mathsf{hk}},3} = \mathsf{Game}_{5,d}$.

Table 5: Distributions of ABE secret keys $\mathsf{sk}_{y,\iota}$ to answer $\mathcal{A}$'s decryption key reveal queries and homomorphic evaluation key reveal queries in $\mathsf{Game}_{4,d,\iota,\zeta,1}$, $\mathsf{Game}_{4,d,\iota,\zeta,2}$, and $\mathsf{Game}_{4,d,\iota,\zeta,3}$

	before ζ-th query	ζ-th query	after ζ-th query
$\mathsf{Game}_{4,d,\iota,\zeta,1}$	semi-functional	pseudo-normal	normal
$\mathsf{Game}_{4,d,\iota,\zeta,2}$	semi-functional	pseudo-SF	normal
$\mathsf{Game}_{4,d,\iota,\zeta,3}$	semi-functional	semi-functional	normal

$\mathsf{Game}_{4,d,\iota,\zeta,1}$. This is the same as $\mathsf{Game}_{4,d,\iota,\zeta-1,3}$ except that $\mathcal{C}$ answers pseudo-normal $\mathsf{sk}_{y,\iota}$ upon $\mathcal{A}$'s ζ-th decryption key reveal queries or homomorphic evaluation key reveal queries on y.

$\mathsf{Game}_{4,d,\iota,\zeta,2}$. This is the same as $\mathsf{Game}_{4,d,\iota,\zeta,1}$ except that $\mathcal{C}$ answers pseudo-SF $\mathsf{sk}_{y,\iota}$ upon $\mathcal{A}$'s ζ-th decryption key reveal queries or homomorphic evaluation key reveal queries on y.

$\mathsf{Game}_{4,d,\iota,\zeta,3}$. This is the same as $\mathsf{Game}_{4,d,\iota,\zeta,2}$ except that $\mathcal{C}$ answers semi-functional $\mathsf{sk}_{y,\iota}$ upon $\mathcal{A}$'s ζ-th decryption key reveal queries or homomorphic evaluation key reveal queries on y.

Table 5 summarizes distributions of $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$ in each game $\mathsf{Game}_{4,d,\iota,\zeta,1}$, $\mathsf{Game}_{4,d,\iota,\zeta,2}$, and $\mathsf{Game}_{4,d,\iota,\zeta,3}$, where all $\mathsf{sk}_{y,2}$ are always normal if $\iota = 1$ and all $\mathsf{sk}_{y,1}$ are always semi-functional if $\iota = 2$. To prove $\mathsf{Game}_{4,d} \approx_c \mathsf{Game}_{5,d}$, we show that $\mathsf{Game}_{4,d,\iota,\zeta-1,3} \approx_c \mathsf{Game}_{4,d,\iota,\zeta,1} \equiv \mathsf{Game}_{4,d,\iota,\zeta,2} \approx_c \mathsf{Game}_{4,d,\iota,\zeta,3}$ as the proof of Lemma 21.

Lemma 29 ($\mathsf{Game}_{4,d,\iota,\zeta-1,3} \approx_c \mathsf{Game}_{4,d,\iota,\zeta,1}$)**.** *If the matrix DDH assumption over $\mathbb{G}_2$ holds, $\mathsf{Game}_{4,d,\iota,\zeta-1,3}$ and $\mathsf{Game}_{4,d,\iota,\zeta,1}$ are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 29. We prove only for $d = 1$ and $\iota = 1$ since proofs for the other cases are essentially the same. We show that for any PPT adversary $\mathcal{A}$ that breaks the adaptive $\mathsf{KH\text{-}CCA}$ security of Π_{ABKHE}, there exists a reduction algorithm $\mathcal{B}_{4,1}$ that solves the $(m_1 + m_2)$-fold matrix DDH assumption over $\mathbb{G}_2$, where

$$|\Pr[E_{4,d,\iota,\zeta-1,3}] - \Pr[E_{4,d,\iota,\zeta,1}]| \leq \mathsf{Adv}^{\mathsf{mDDH}_{\mathbb{G}_2}}_{\mathcal{B}_{4,1}}(\lambda). \tag{23}$$

The proof is almost the same as the proof of Lemma 26. After $\mathcal{B}_{4,1}$ receives $(\mathcal{G}(1^\lambda), [\mathbf{B}]_2, [\mathbf{V}]_2)$, it sends mpk to $\mathcal{A}$ in the same way as $\mathcal{B}_{3,1}$. Since $\mathcal{B}_{4,1}$ knows $(\mathbf{u}_\iota)_{\iota\in[2]}$, it can answer all $\mathcal{A}$'s evaluation queries by creating normal $\mathsf{sk}_{y,1}$, and $\mathsf{sk}_{y,2}$. Since $\mathcal{B}_{4,1}$ knows $(\mathbf{u}_\iota)_{\iota\in[0,2]}$, it can answer all $\mathcal{A}$'s decryption queries by creating normal $\mathsf{sk}_{y,0}$, $\mathsf{sk}_{y,1}$, and $\mathsf{sk}_{y,2}$. Since $\mathcal{B}_{4,1}$ knows $(\mathbf{u}_\iota)_{\iota\in[0,2]}$ and $\mathbf{a}^\perp$, it can answer all $\mathcal{A}$'s homomorphic evaluation key reveal queries and decryption key reveal queries before the ζ-th query by creating semi-functional $\mathsf{sk}_{y,0}$ and $\mathsf{sk}_{y,1}$ and normal $\mathsf{sk}_{y,2}$. Similarly, $\mathcal{B}_{4,1}$ can answer all $\mathcal{A}$'s homomorphic evaluation key reveal queries and decryption key reveal queries after the ζ-th query by creating semi-functional $\mathsf{sk}_{y,0}$ and normal $\mathsf{sk}_{y,1}$ and $\mathsf{sk}_{y,2}$. Since $\mathcal{B}_{4,1}$ knows $(\mathbf{u}_\iota)_{\iota\in[0,2]}$ and $\mathbf{W}_1, \ldots, \mathbf{W}_n$, it can answer $\mathcal{A}$'s challenge query by creating semi-functional $\mathsf{ct}^\star_{x^\star}$.

Upon $\mathcal{A}$'s ζ-th homomorphic evaluation key reveal query or decryption key reveal query on y, $\mathcal{B}_{4,1}$ creates semi-functional $\mathsf{sk}_{y,0}$, normal $\mathsf{sk}_{y,2}$, and $\mathsf{sk}_{y,1} = ((\mathsf{sk}_{0,0,i'})_{i'\in[m_1]}, (\mathsf{sk}_{0,1,t'})_{t'\in[m_3]})$ in the same way as (21) except

$$\mathsf{sk}_{1,1,t'} = [\mathbf{u}_1]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{v}_{m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\mathbf{W}_j\mathbf{v}_{i'}]_2^{\phi_{t',i',j}}, \tag{24}$$

where $\mathbf{v}_i$ is an i-th column vector of $\mathbf{V}$. The ζ-th $\mathsf{sk}_{y,1}$ is distributed as in $\mathsf{Game}_{4,d,1,\zeta-1,3}$ (resp. $\mathsf{Game}_{4,d,1,\zeta,1}$) if $\mathbf{V} = \mathbf{BR}$ (resp. $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(m_1+m_2)}$). Thus, the inequality (23) holds. $\square$

Lemma 30 ($\mathsf{Game}_{4,d,\iota,\zeta,1} \equiv \mathsf{Game}_{4,d,\iota,\zeta,2}$)**.** *If the* PES *satisfies the perfect security,* $\mathsf{Game}_{4,d,\iota,\zeta,1}$ *and* $\mathsf{Game}_{4,d,\iota,\zeta,2}$ *follow the same distribution from $\mathcal{A}$'s view.*

Proof of Lemma 30. We prove only for $d = 1$ and $\iota = 1$ since proofs for the other cases are essentially the same. As the proof of Lemma 27, we set

$$\mathbf{W}_1 = \widetilde{\mathbf{W}}_1 + b_1(\mathbf{a}^{\perp\top}\mathbf{b}^{\perp})^{-1}\mathbf{a}\mathbf{b}^{\perp\top}, \ldots, \mathbf{W}_n = \widetilde{\mathbf{W}}_n + b_n(\mathbf{a}^{\perp\top}\mathbf{b}^{\perp})^{-1}\mathbf{a}\mathbf{b}^{\perp\top}$$

where $\widetilde{\mathbf{W}}_1, \ldots, \widetilde{\mathbf{W}}_n \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$ and $b_1, \ldots, b_n \leftarrow_R \mathbb{Z}_p$. Then, only elements that depend on $b_1, \ldots, b_n$ are the semi-functional challenge ciphertext $\mathsf{ct}^\star_{x^\star}$ and ζ-th pseudo-normal $\mathsf{sk}_{y,1}$. In particular, we have

$$\begin{aligned}
\mathsf{ct}^\star_{1,t} = & \prod_{i\in[w_2]} [\mathbf{A}\mathbf{s}_{w_1+i}]_1^{\eta_{t,i}} \cdot \prod_{i\in[0,w_1],j\in[n]} [\widetilde{\mathbf{W}}_j^\top(\mathbf{A}\mathbf{s}_i + s_i\mathbf{b}^\perp)]_1^{\eta_{t,i,j}} \\
& \cdot [\mathbf{b}^\perp]_1^{\sum_{i\in[w_2]}\eta_{t,i}\hat{s}_i + \sum_{i\in[0,w_1],j\in[n]}\eta_{t,i,j}s_ib_j}, \\
\mathsf{sk}_{1,1,t'} = & [\mathbf{u}_1]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\widetilde{\mathbf{W}}_j(\mathbf{B}\mathbf{r}_{i'} + r_{i'}\mathbf{a}^\perp)]_2^{\phi_{t',i',j}} \\
& \cdot [\mathbf{a}^\perp]_2^{\sum_{i'\in[m_2]}\phi_{t',i'}\hat{r}_{i'} + \sum_{i'\in[m_2],j\in[n]}\phi_{t',i',j}r_ib_j}.
\end{aligned}$$

Observe that even when we do not know all of $(s_0, s_1, \ldots, s_{w_1}, \hat{s}_1, \ldots, \hat{s}_{w_2}, r_1, \ldots, r_{m_1}, \hat{r}_1, \ldots, \hat{r}_{m_2})$, $(s_0, s_1, \ldots, s_{w_1}, r_1, \ldots, r_{m_1}, (\sum_{i\in[w_2]}\eta_{t,i}\hat{s}_i + \sum_{i\in[0,w_1],j\in[n]}\eta_{t,i,j}s_ib_j)_{t\in[w_3]}, (\sum_{i'\in[m_2]}\phi_{t',i'}\hat{r}_{i'} + \sum_{i'\in[m_1],j\in[n]}\phi_{t',i',j}r_{i'}b_j)_{t'\in[m_3]})$ are sufficient for simulating the semi-functional challenge ciphertext $\mathsf{ct}^\star_{x^\star}$ and ζ-th pseudo-normal $\mathsf{sk}_{y,1}$. Since it holds that $f(x^\star, y) = 0$ and all $s_0, s_1, \ldots, s_{w_1}, \hat{s}_1, \ldots, \hat{s}_{w_2}, r_1, \ldots, r_{m_1}, \hat{r}_1, \ldots, \hat{r}_{m_2}$ are sampled according to the uniform distribution over $\mathbb{Z}_p$, the perfect security of PES ensures that $\mathsf{ct}^\star_{x^\star}$ and ζ-th $\mathsf{sk}_{y,1}$ are identically distributed by simulating with $(s_0, \mathbf{s}, \mathbf{r}, (\sum_{i\in[w_2]}\eta_{t,i}\hat{s}_i + \sum_{i\in[0,w_1],j\in[n]}\eta_{t,i,j}s_ib_j)_{t\in[w_3]}, (\phi_{t'}\alpha_{1,y} + \sum_{i'\in[m_2]}\phi_{t',i'}\hat{r}_{i'} + \sum_{i'\in[m_1],j\in[n]}\phi_{t',i',j}r_{i'}b_j)_{t'\in[m_3]})$, where $\alpha_{1,y} \leftarrow_R \mathbb{Z}_p$. Then, we have

$$\begin{aligned}
\mathsf{sk}_{1,1,t'} = & [\mathbf{u}_1]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\widetilde{\mathbf{W}}_j(\mathbf{B}\mathbf{r}_{i'} + r_{i'}\mathbf{a}^\perp)]_2^{\phi_{t',i',j}} \\
& \cdot [\mathbf{a}^\perp]_2^{\phi_{t'}\alpha_{1,y} + \sum_{i'\in[m_2]}\phi_{t',i'}\hat{r}_{i'} + \sum_{i'\in[m_2],j\in[n]}\phi_{t',i',j}r_ib_j} \\
= & [\mathbf{u}_1 + \alpha_{1,y}\mathbf{a}^\perp]_2^{\phi_{t'}} \cdot \prod_{i'\in[m_2]} [\mathbf{B}\mathbf{r}_{m_1+i'} + \hat{r}_{i'}\mathbf{a}^\perp]_2^{\phi_{t',i'}} \cdot \prod_{i'\in[m_2],j\in[n]} [\mathbf{W}_j(\mathbf{B}\mathbf{r}_{i'} + r_{i'}\mathbf{a}^\perp)]_2^{\phi_{t',i',j}}.
\end{aligned}$$

As the proof of Lemma 27, the distribution of ζ-th $\mathsf{sk}_{y,1}$ is identically distributed to pseudo-SF secret key. Thus, we complete the proof. $\square$

Lemma 31 ($\mathsf{Game}_{4,d,\iota,\zeta,2} \approx_c \mathsf{Game}_{4,d,\iota,\zeta,3}$)**.** *If the matrix DDH assumption over $\mathbb{G}_2$ holds,* $\mathsf{Game}_{4,d,\iota,\zeta,2}$ *and* $\mathsf{Game}_{4,d,\iota,\zeta,3}$ *are computationally indistinguishable for any PPT $\mathcal{A}$.*

Proof of Lemma 31. We prove only for $d = 1$ and $\iota = 1$ since proofs for the other cases are essentially the same. We show that for any PPT adversary $\mathcal{A}$ that breaks the adaptive KH-CCA security of Π_{ABKHE}, there exists a reduction algorithm $\mathcal{B}_{4,3}$ that solves the $(m_1 + m_2)$-fold matrix DDH assumption over $\mathbb{G}_2$, where

$$|\Pr[E_{4,d,\iota,\zeta,2}] - \Pr[E_{4,d,\iota,\zeta,3}]| \leq \mathsf{Adv}^{\mathsf{mDDH}_{\mathbb{G}_2}}_{\mathcal{B}_{4,3}}(\lambda). \tag{25}$$

The proof is almost the same as the proof of Lemmata 28 and 29. After $\mathcal{B}_{4,3}$ receives $(\mathcal{G}(1^\lambda), [\mathbf{B}]_2, [\mathbf{V}]_2)$, it sends mpk to $\mathcal{A}$ in the same way as $\mathcal{B}_{4,1}$. $\mathcal{B}_{4,3}$ answers all $\mathcal{A}$'s decryption key reveal queries, homomorphic evaluation key reveal queries, decryption queries, evaluation queries, and challenge query in the same way as $\mathcal{B}_{4,1}$ except ζ-th $\mathsf{sk}_{y,1}$. $\mathcal{B}_{4,3}$ creates ζ-th $\mathsf{sk}_{y,1}$ in the same way as (24) except

$$\mathsf{sk}_{1,1,t'} = [\mathbf{u}_1 + \alpha_{1,y}\mathbf{a}^\perp]_2^{\phi_{t'}} \cdot \prod_{i' \in [m_2]} [\mathbf{v}_{m_1+i'}]_2^{\phi_{t',i'}} \cdot \prod_{i' \in [m_2], j \in [n]} [\mathbf{W}_j \mathbf{v}_{i'}]_2^{\phi_{t',i',j}},$$

where $\mathbf{v}_i$ is an i-th column vector of $\mathbf{V}$ and $\alpha_{1,y} \leftarrow_R \mathbb{Z}_p$. The ζ-th $\mathsf{sk}_{y,1}$ is distributed as in $\mathsf{Game}_{4,d,1,\zeta,3}$ (resp. $\mathsf{Game}_{4,d,1,\zeta,2}$) if $\mathbf{V} = \mathbf{BR}$ (resp. $\mathbf{V} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(m_1+m_2)}$). Thus, the inequality (25) holds. □

Based on Lemmata 29, 30, and 31, we have

$$|\Pr[E_4] - \Pr[E_5]| \leq 2(Q_{\mathsf{hk}} + Q_{\mathsf{dk}})\Big(\mathsf{Adv}_{\mathcal{B}_{4,1}}^{\mathsf{mDDH}_{\mathbb{G}_2}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{4,3}}^{\mathsf{mDDH}_{\mathbb{G}_2}}(\lambda)\Big).$$

□

9 Conclusion

In this paper, we proposed generic constructions of ABKFHE and ABKHE. In advance of ABKFHE, we modified Canetti et al.'s CCA1-secure FHE scheme [CRRV17] and proposed a generic construction of KFHE based on MFHE, IBE, OTS, and MAC, where the resulting scheme is the first KFHE scheme secure solely under the LWE assumption in the standard model. Then, we replaced several building blocks of KFHE with attribute-based ones and provided a generic construction of ABKFHE based on MFHE, DABE, and OTS, where the resulting scheme implies the first IBKFHE scheme. For this purpose, we constructed a DABE scheme by combining with Yamada's adaptively secure IBE scheme [Yam17] and Boneh et al.'s selectively secure ABE scheme [BGG+14]. Next, in advance of ABKHE, we provided a simpler proof of Emura et al.'s KHPKE scheme [EHN+18] if it is instantiated under the matrix DDH assumption. Then, we proposed a generic construction of ABKHE from PES by combining with ABE schemes over dual system groups [AC16, AC17, CGW15] and Emura et al.'s KHPKE scheme [EHN+18], where the resulting scheme implies the first IBKHE scheme under the standard k-linear assumption.

Due to the inefficiency of Canetti et al.'s CCA1-secure FHE scheme [CRRV17], our proposed ABKFHE scheme is also inefficient. To obtain more efficient ABKFHE schemes, a design of a more efficient CCA1-secure FHE scheme has to be an interesting open problem. Since there are several expressive ABE schemes which are not covered by PES, constructions of keyed homomorphic variants of the schemes should be an interesting open problem. A construction of attribute-based two-level keyed homomorphic encryption is also an interesting open problem.

Acknowledgement. We would like to thank anonymous reviewers of PKC 2024.

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*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, Heidelberg, May / June 2010. doi:10.1007/978-3-642-13190-5_28.

[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*, volume 6223 of *Lecture Notes in Computer Science*, pages 98–115. Springer, Heidelberg, August 2010. doi:10.1007/978-3-642-14623-7_6.

[ABS17] Miguel Ambrona, Gilles Barthe, and Benedikt Schmidt. Generic transformations of predicate encodings: Constructions and applications. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 36–66. Springer, Heidelberg, August 2017. doi:10.1007/978-3-319-63688-7_2.

[AC16] Shashank Agrawal and Melissa Chase. A study of pair encodings: Predicate encryption in prime order groups. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 259–288. Springer, Heidelberg, January 2016. doi:10.1007/978-3-662-49099-0_10.

[AC17] Shashank Agrawal and Melissa Chase. Simplifying design and analysis of complex predicate encryption schemes. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 627–656. Springer, Heidelberg, April / May 2017. doi:10.1007/978-3-319-56620-7_22.

[AJJM20] Prabhanjan Ananth, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Multi-key fully-homomorphic encryption in the plain model. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020: 18th Theory of Cryptography Conference, Part I*, volume 12550 of *Lecture Notes in Computer Science*, pages 28–57. Springer, Heidelberg, November 2020. doi:10.1007/978-3-030-64375-1_2.

[Amb21] Miguel Ambrona. Generic negation of pair encodings. In Juan Garay, editor, *PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12711 of *Lecture Notes in Computer Science*, pages 120–146. Springer, Heidelberg, May 2021. doi:10.1007/978-3-030-75248-4_5.

[Att14] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 557–577. Springer, Heidelberg, May 2014. doi:10.1007/978-3-642-55220-5_31.

[Att16] Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 591–623. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53890-6_20.

[Att19] Nuttapong Attrapadung. Unbounded dynamic predicate compositions in attribute-based encryption. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 34–67. Springer, Heidelberg, May 2019. doi:10.1007/978-3-030-17653-2_2.

[AY15] Nuttapong Attrapadung and Shota Yamada. Duality in ABE: Converting attribute based encryption for dual predicate and dual policy via computational encodings. In Kaisa Nyberg, editor, *Topics in Cryptology – CT-RSA 2015*, volume 9048 of *Lecture Notes in Computer Science*, pages 87–105. Springer, Heidelberg, April 2015. doi:10.1007/978-3-319-16715-2_5.

[BB04] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, Heidelberg, August 2004. doi:10.1007/978-3-540-28628-8_27.

[BBC+18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 669–699. Springer, Heidelberg, August 2018. doi:10.1007/978-3-319-96881-0_23.

[BCC+17] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. The hunting of the SNARK. *Journal of Cryptology*, 30(4):989–1066, October 2017. doi:10.1007/s00145-016-9241-9.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 111–120. ACM Press, June 2013. doi:10.1145/2488608.2488623.

[BCTW16] Zvika Brakerski, David Cash, Rotem Tsabary, and Hoeteck Wee. Targeted homomorphic attribute-based encryption. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 330–360. Springer, Heidelberg, October / November 2016. doi:10.1007/978-3-662-53644-5_13.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556. Springer, Heidelberg, May 2014. doi:10.1007/978-3-642-55220-5_30.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_1.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012: 3rd Innovations in Theoretical Computer Science*, pages 309–325. Association for Computing Machinery, January 2012. doi:10.1145/2090236.2090262.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12. Springer, Heidelberg, August 1998. doi:10.1007/BFb0055716.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584. ACM Press, June 2013. doi:10.1145/2488608.2488680.

[BN08] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4):469–491, October 2008. doi:10.1007/s00145-008-9026-x.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, Heidelberg, August 2012. doi:10.1007/978-3-642-32009-5_50.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 97–106. IEEE Computer Society Press, October 2011. doi:10.1109/FOCS.2011.12.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, Heidelberg, August 2011. doi:10.1007/978-3-642-22792-9_29.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In Moni Naor, editor, *ITCS 2014: 5th Conference on Innovations in Theoretical Computer Science*, pages 1–12. Association for Computing Machinery, January 2014. doi:10.1145/2554797.2554799.

[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, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 35–65. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70697-9_2.

[CGW15] Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 595–624. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_20.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, Heidelberg, May 2004. doi:10.1007/978-3-540-24676-3_13.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of Cryptology*, 25(4):601–639, October 2012. doi:10.1007/s00145-011-9105-2.

[CLL+14] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter identity-based encryption via asymmetric pairings. *Des. Codes Cryptogr.*, 73(3):911–947, 2014. doi:10.1007/S10623-013-9834-3.

[CM15] Michael Clear and Ciaran McGoldrick. Multi-identity and multi-key leveled FHE from learning with errors. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 630–656. Springer, Heidelberg, August 2015. doi:10.1007/978-3-662-48000-7_31.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 1–29. Springer, Heidelberg, December 2019. doi:10.1007/978-3-030-36033-7_1.

[CRRV17] Ran Canetti, Srinivasan Raghuraman, Silas Richelson, and Vinod Vaikuntanathan. Chosen-ciphertext secure fully homomorphic encryption. In Serge Fehr, editor, *PKC 2017: 20th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 10175 of *Lecture Notes in Computer Science*, pages 213–240. Springer, Heidelberg, March 2017. doi:10.1007/978-3-662-54388-7_8.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, Heidelberg, August 1998. doi:10.1007/BFb0055717.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, Heidelberg, April / May 2002. doi:10.1007/3-540-46035-7_4.

[CW14] Jie Chen and Hoeteck Wee. Dual system groups and its applications — compact HIBE and more. Cryptology ePrint Archive, Report 2014/265, 2014. https://eprint.iacr.org/2014/265.

[DGM15] Ricardo Dahab, Steven Galbraith, and Eduardo Morais. Adaptive key recovery attacks on NTRU-based somewhat homomorphic encryption schemes. In Anja Lehmann and Stefan Wolf, editors, *ICITS 15: 8th International Conference on Information Theoretic Security*, volume 9063 of *Lecture Notes in Computer Science*, pages 283–296. Springer, Heidelberg, May 2015. doi:10.1007/978-3-319-17470-9_17.

[EHK+17] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Luis Villar. An algebraic framework for Diffie-Hellman assumptions. *Journal of Cryptology*, 30(1):242–288, January 2017. doi:10.1007/s00145-015-9220-6.

[EHN$^+$13] Keita Emura, Goichiro Hanaoka, Koji Nuida, Go Ohtake, Takahiro Matsuda, and Shota Yamada. Chosen ciphertext secure keyed-homomorphic public-key encryption. Cryptology ePrint Archive, Report 2013/390, 2013. https://eprint.iacr.org/2013/390.

[EHN$^+$18] Keita Emura, Goichiro Hanaoka, Koji Nuida, Go Ohtake, Takahiro Matsuda, and Shota Yamada. Chosen ciphertext secure keyed-homomorphic public-key cryptosystems. *Des. Codes Cryptogr.*, 86(8):1623–1683, 2018. doi:10.1007/S10623-017-0417-6.

[EHO$^+$13] Keita Emura, Goichiro Hanaoka, Go Ohtake, Takahiro Matsuda, and Shota Yamada. Chosen ciphertext secure keyed-homomorphic public-key encryption. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 32–50. Springer, Heidelberg, February / March 2013. doi:10.1007/978-3-642-36362-7_3.

[Emu21] Keita Emura. On the security of keyed-homomorphic PKE: preventing key recovery attacks and ciphertext validity attacks. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 104-A(1):310–314, 2021. doi:10.1587/TRANSFUN.2020EAL2039.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178. ACM Press, May / June 2009. doi:10.1145/1536414.1536440.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 626–645. Springer, Heidelberg, May 2013. doi:10.1007/978-3-642-38348-9_37.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM Press, May 2008. doi:10.1145/1374376.1374407.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, Heidelberg, August 2013. doi:10.1007/978-3-642-40041-4_5.

[HK17] Ryo Hiromasa and Yutaka Kawai. Dynamic multi target homomorphic attribute-based encryption. In Máire O'Neill, editor, *16th IMA International Conference on Cryptography and Coding*, volume 10655 of *Lecture Notes in Computer Science*, pages 25–43. Springer, Heidelberg, December 2017.

[HWZ07] Qiong Huang, Duncan S. Wong, and Yiming Zhao. Generic transformation to strongly unforgeable signatures. In Jonathan Katz and Moti Yung, editors, *ACNS 07: 5th International Conference on Applied Cryptography and Network Security*, volume 4521 of *Lecture Notes in Computer Science*, pages 1–17. Springer, Heidelberg, June 2007. doi:10.1007/978-3-540-72738-5_1.

[JR15] Charanjit S. Jutla and Arnab Roy. Dual-system simulation-soundness with applications to UC-PAKE and more. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 630–655. Springer, Heidelberg, November / December 2015. doi:10.1007/978-3-662-48797-6_26.

[KY16] Shuichi Katsumata and Shota Yamada. Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 682–712. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53890-6_23.

[LDM+16] Junzuo Lai, Robert H. Deng, Changshe Ma, Kouichi Sakurai, and Jian Weng. CCA-secure keyed-fully homomorphic encryption. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 9614 of *Lecture Notes in Computer Science*, pages 70–98. Springer, Heidelberg, March 2016. doi:10.1007/978-3-662-49384-7_4.

[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*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_20.

[LMSV12] Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren. On CCA-secure somewhat homomorphic encryption. In Ali Miri and Serge Vaudenay, editors, *SAC 2011: 18th Annual International Workshop on Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 55–72. Springer, Heidelberg, August 2012. doi:10.1007/978-3-642-28496-0_4.

[LPJY14] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 514–532. Springer, Heidelberg, May 2014. doi:10.1007/978-3-642-55220-5_29.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *44th Annual ACM Symposium on Theory of Computing*, pages 1219–1234. ACM Press, May 2012. doi:10.1145/2213977.2214086.

[MBKM19] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 2111–2128. ACM Press, November 2019. doi:10.1145/3319535.3339817.

[ML19] Xuecheng Ma and Dongdai Lin. Multi-identity IBFHE and multi-attribute ABFHE in the standard model. In Kwangsu Lee, editor, *ICISC 18: 21st International*

Conference on Information Security and Cryptology, volume 11396 of *Lecture Notes in Computer Science*, pages 69–84. Springer, Heidelberg, November 2019. doi:10.1007/978-3-030-12146-4_5.

[MN22] Yusaku Maeda and Koji Nuida. Chosen ciphertext secure keyed two-level homomorphic encryption. In Khoa Nguyen, Guomin Yang, Fuchun Guo, and Willy Susilo, editors, *ACISP 22: 27th Australasian Conference on Information Security and Privacy*, volume 13494 of *Lecture Notes in Computer Science*, pages 209–228. Springer, Heidelberg, November 2022. doi:10.1007/978-3-031-22301-3_11.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_41.

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 735–763. Springer, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_26.

[PD20] Tapas Pal and Ratna Dutta. Chosen-ciphertext secure multi-identity and multi-attribute pure FHE. In Stephan Krenn, Haya Shulman, and Serge Vaudenay, editors, *CANS 20: 19th International Conference on Cryptology and Network Security*, volume 12579 of *Lecture Notes in Computer Science*, pages 387–408. Springer, Heidelberg, December 2020. doi:10.1007/978-3-030-65411-5_19.

[PS16] Chris Peikert and Sina Shiehian. Multi-key FHE from LWE, revisited. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 217–238. Springer, Heidelberg, October / November 2016. doi:10.1007/978-3-662-53644-5_9.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM Press, May 2005. doi:10.1145/1060590.1060603.

[SET22] Shingo Sato, Keita Emura, and Atsushi Takayasu. Keyed-fully homomorphic encryption without indistinguishability obfuscation. In Giuseppe Ateniese and Daniele Venturi, editors, *ACNS 22: 20th International Conference on Applied Cryptography and Network Security*, volume 13269 of *Lecture Notes in Computer Science*, pages 3–23. Springer, Heidelberg, June 2022. doi:10.1007/978-3-031-09234-3_1.

[Tak21] Atsushi Takayasu. Tag-based ABE in prime-order groups via pair encoding. *Des. Codes Cryptogr.*, 89(8):1927–1963, 2021. doi:10.1007/S10623-021-00894-4.

[vGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, Heidelberg, May / June 2010. doi:10.1007/978-3-642-13190-5_2.

[Wat05] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, Heidelberg, May 2005. doi:10.1007/11426639_7.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, Heidelberg, August 2009. doi:10.1007/978-3-642-03356-8_36.

[Wee14] Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 616–637. Springer, Heidelberg, February 2014. doi:10.1007/978-3-642-54242-8_26.

[Yam17] Shota Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 161–193. Springer, Heidelberg, August 2017. doi:10.1007/978-3-319-63697-9_6.

[ZPS12] Zhenfei Zhang, Thomas Plantard, and Willy Susilo. On the CCA-1 security of somewhat homomorphic encryption over the integers. In Mark Dermot Ryan, Ben Smyth, and Guilin Wang, editors, *Information Security Practice and Experience - 8th International Conference, ISPEC 2012, Hangzhou, China, April 9-12, 2012. Proceedings*, volume 7232 of *Lecture Notes in Computer Science*, pages 353–368. Springer, 2012. doi:10.1007/978-3-642-29101-2_24.

[ZSZ+22] Yuncong Zhang, Alan Szepieniec, Ren Zhang, Shi-Feng Sun, Geng Wang, and Dawu Gu. VOProof: Efficient zkSNARKs from vector oracle compilers. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022: 29th Conference on Computer and Communications Security*, pages 3195–3208. ACM Press, November 2022. doi:10.1145/3548606.3559387.