Generic Constructions of Master-Key KDM Secure Attribute-based Encryption*

Jiaxin Pan†[1] Chen Qian‡[2,3] Benedikt Wagner [4,5]

November 29, 2023

[1] NTNU – Norwegian University of Science and Technology, Trondheim, Norway
jiaxin.pan@ntnu.no
[2] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Qingdao, Shandong, China
chen.qian@sdu.edu.cn
[3] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China

[4] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
benedikt.wagner@cispa.de
[5] Saarland University, Saarbrücken, Germany

Abstract

Master-key key-dependent message (mKDM) security is a strong security notion for attribute-based encryption (ABE) schemes, which has been investigated in recent years. This line of research was started with identity-based encryption (IBE; Garg, Gay, and Hajiabadi, PKC 2020) and then was extended to (more general) ABE (Feng, Gong, and Chen, PKC 2021). Both these constructions are based on dual system techniques which crucially rely on pairings. How to construct mKDM secure ABEs without pairings or even generically was an open problem.

In this paper, we propose two generic constructions of mKDM secure ABE from an ABE secure against chosen-plaintext attacks in the random oracle model (ROM) and standard model. In the ROM, our construction is very efficient, and it gives rise to the *first* mKDM secure ABE from lattices. Our construction in the standard model requires indistinguishability obfuscation, but it shows that, even in the standard model, mKDM security can be achieved generically, and it is not limited to dual-system-based techniques.

Keywords: Master-key KDM, attribute-based encryption, identity-based encryption, generic construction, indistinguishability obfuscation.

1 Introduction

Indistinguishability against chosen-plaintext attacks (IND-CPA, or semantic security) is the most basic security notion for encryption schemes, which guarantees that an adversary can hardly learn any information of the plaintext encrypted in a ciphertext. Key-dependent message (KDM) security [BRS03] is a stronger form of semantic security where the encrypted plaintext may depend on the secret key. This stronger notion is desirable due to the use of encryption in practice, and also as a building block for more advanced cryptosystems, such as fully homomorphic encryption [Gen09].

MKDM SECURITY FOR IBE AND ABE. Identity-based encryption (IBE) allows to encrypt with respect to identities instead of public keys. Its classical security requirement, IND-CPA, is a natural extension of that of public-key encryption, but additionally allows adversaries to ask many user secret keys adaptively.

*Corresponding author: Chen Qian

†Supported by the Research Council of Norway under Project No. 324235

‡Supported by the National Key Research and Development Program of China (Grant No. 2018YFA0704702), the Major Basic Research Project of Natural Science Foundation of Shandong Province, China (Grant No. ZR202010220025)

As for public key encryption, one can consider the stronger notion of KDM security for IBE. Most works that investigate KDM security for IBE consider the notion of user-key KDM (uKDM) security [AP12, CZDC16, KT18], where the messages encrypted can arbitrarily depend on a set of user secret keys. Another notion is master-key KDM security (mKDM) [GHV12, GGH20, FGC21], which is technically more challenging to achieve. Here, the messages can arbitrarily depend on the master secret key.

As it has been discussed in [GGH20], the notion of mKDM is interesting in its own right and more natural than uKDM security. For instance, an mKDM secure IBE scheme implies a KDM-CCA secure public-key encryption scheme [GHV12], which is not the case for a uKDM secure one. We refer [GGH20] for detailed discussions.

Using predicate encoding schemes Feng, Gong, and Chen [FGC21] have constructed the first mKDM secure attribute-based encryption (ABE) schemes. ABE is a generalization of IBE, and offers more fine-grained access control. Concretely, in an ABE scheme for boolean predicate $\mathcal{P}$, messages are encrypted under descriptive values x, user-secret keys are associated with values y, and a user-secret key decrypts the ciphertext if and only if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$. Here, the predicate $\mathcal{P}$ may express arbitrary access policy. In this paper, we construct mKDM secure ABE.

Prior Work on mKDM security. We note that mKDM security is difficult to achieve, and most prior works on mKDM security focus on IBE schemes. The study of mKDM security was initiated by Galindo et al. [GHV12] in 2012, and their IBE construction is restricted in the sense that it is only selectively secure and the number of its KDM queries must be bounded beforehand.

The first adaptively mKDM secure IBE scheme was proposed by Garg et al. [GGH20] in 2020 using techniques from tightly secure IBEs [HKS15, AHY15, GDCC16]. The Garg et al. scheme contains $\Theta(\lambda^2)$ many group elements[1] in a master public key due to the tight security technique. The first mKDM secure ABE scheme was only constructed recently by Feng et al. [FGC21]. Furthermore, their ABE implies a mKDM secure IBE with constant-size master public keys (cf. Table 1).

Both works on adaptive mKDM security require dual system techniques [Wat09, LW10] which heavily rely on pairing-based assumptions. Especially, mKDM secure IBE or ABE from other assumptions, e.g. lattice-based assumptions, was not known before our work. Motivated by this state of affairs, we raise the natural question whether there is a generic construction of mKDM secure ABE schemes, for instance, from any IND-CPA secure ABE. We note that there are generic constructions of uKDM secure IBE [CZDC16, KT18], while this is the missing piece for mKDM security.

1.1 Our Contributions

We propose the first two generic constructions of mKDM secure ABE with and without random oracles. They are the first constructions that do not rely on dual system techniques.

Our Efficient Construction in the Random Oracle Model. Our construction with random oracles is a generic transformation that turns an IND-CPA secure ABE to an mKDM secure one by computing one additional hash. In fact, our transformation only requires one-wayness against chosen-plaintext attacks (OW-CPA) of the underlying ABE scheme, which is a security notion weaker than and implied by IND-CPA. If we assume OW-CPA security with multiple challenge ciphertexts, our security proof is tight, namely, its security loss is constant. This approach is an extension of hybrid encryption approach for PKEs in the ABE setting, and we borrow techniques from Kitagawa et al. [KMHT16] which carefully program the random oracle.

We stress that using the schemes in [GPV08, KYY18] our generic construction gives us the first (tightly) mKDM secure IBE schemes based on lattices. Unfortunately, there is no tightly OW-CPA secure ABE, and hence we do not have suitable scheme to implement our mKDM secure ABE tightly. But with an adaptively secure lattice-based ABE (such as the Tsabary scheme [Tsa19]), our generic construction yields the first mKDM secure ABE scheme from lattices.

Our generic construction in the random oracle model is very efficient, and we view it as obtaining mKDM security almost for free. Moreover, we provide another transformation from OW-CPA to mKDM security against chosen-ciphertext attacks (mKDM-CCA) using the Fujisaki-Okamoto transformation [FO99] for ABEs. This transformation is as efficient as the first one. One should always aim for the strongest security notion while keeping schemes efficient, and our result provides one way of achieving this.

[1] Here, λ is the security paramter.

To support the claim of efficiency, we instantiate our generic construction with pairings for IBE and compare it with the previous known mKDM secure IBE schemes, which are all in the pairing setting, in Table 1. In this table, our schemes are instantiated with the Boneh-Franklin IBE [BF01] and its tight variant using the Katz-Wang random-bit technique [KW03], see also Section 3.3. We only focus on IBE, since most works on mKDM security are about IBE and we want to take tightness into account (while there is no tightly secure ABE).

Scheme	Assumption	ROM	CCA	tight	$\vert\mathsf{mpk}\vert$	$\vert\mathsf{sk_{id}}\vert$	$\vert\mathsf{ct}\vert$
GGH [GGH20]	SXDH	✗	✗	✓	$\Theta(\lambda^2)$	$\Theta(\lambda)$	$\Theta(\lambda)$
FGC [FGC21]	SXDH	✗	✓	✗	15	10	4
BF [BF01] + Sec. 3.2	BDH	✓	✓	✗	1	1	2
BF-KW [BF01] + Sec. 3.2	BDH	✓	✓	✓	1	1	3

Table 1: Comparison of existing adaptively mKDM secure identity-based encryption schemes (above the line) and our schemes in the ROM (below the line) in the pairing setting. Sizes of keys and ciphertexts are given as the number of group elements. For $|\mathsf{ct}|$, we only count the overhead, i.e. we subtract the encoding size of a message. λ is the security parameter. Note that our CPA and CCA constructions in Sections 3.1 and 3.2, respectively, have the same efficiency in terms of $|\mathsf{mpk}|$, $|\mathsf{sk_{id}}|$ and $|\mathsf{ct}|$.

Our Construction in the Standard Model. Although our construction in the ROM is practical, it crucially relies on the random oracle model. To remove the need for such an idealized model, we propose another generic construction of mKDM secure ABEs in the standard model. It transforms an IND-CPA secure ABE to an mKDM secure one using indistinguishability obfuscation (iO) [BGI$^+$01] and non-interactive proof systems. iO has been constructed recently from circular security of the GSW FHE scheme [GP21] and well-formed (sub-exponential) assumptions [JLS21].

Due to the use of iO, our construction of mKDM ABE is only a feasibility result, and it is far from being practical. However, it is theoretically interesting, and introduces new ways of achieving mKDM security. Our techniques can serve as a starting point for further study of generic constructions of mKDM ABE in the standard model.

Similar to our practical constructions in the ROM, we have constructions with mKDM-CPA and mKDM-CCA security in the standard model. In particular, we use a Naor-Yung-like [NY90, CCS09] transformation to lift mKDM-CPA security to mKDM-CCA security.

Open Problems. To obtain efficiency of our constructions in the standard model, we leave generically constructing mKDM secure ABE without iO as our main open problem. Further, our standard model construction relies on a perfectly complete and sound NIZK proof system. This motivates the study of such a proof system.

1.2 Technical Overview

We give an high-level overview of our techniques. Due to space limitations, we only discuss our construction in the standard model. For simplicity of exposition, we consider the special case of IBE in this overview. The detailed construction (for ABE) can be found in Section 4.

In general, the technical tension of proving KDM security is that an adversary will submit a function and the reduction needs to apply this function on the secret key and encrypts its result. However, the reduction itself usually does not know this secret key and therefore it seems rather challenging to achieve this security. Our starting point is trying to shift the burden of constructing KDM ciphertexts of master secret keys to the adversary. This has been proposed in the PKE setting by Marcedone, Pass, and shelat [MPs16]. However, it is difficult to “translate” this to the mKDM IBE setting. In the following, we first recall their idea, demonstrate the difficulty in achieving mKDM IBE, and explain how we resolve it.

Warmup: KDM Security for PKE. We first demonstrate our idea using the simpler case of public key encryption. This idea is from the work of Marcedone, Pass, and shelat [MPs16]. Let $\mathcal{R}$ be an **NP** relation and $\mathcal{L}_{\mathcal{R}} \subseteq \mathcal{X}$ is the corresponding language such that $\mathcal{L}_{\mathcal{R}}$ and $\mathcal{X} \setminus \mathcal{L}_{\mathcal{R}}$ are computationally indistinguishable. Our KDM secure public key encryption scheme is as follows. Our public key is a statement $x^* \in \mathcal{L}_\mathcal{R}$ and its secret key is a witness w^* such that $(x^*, w^*) \in \mathcal{R}$. A ciphertext for message $\mathsf{m} = f(w^*)$ is an obfuscation of the circuit $\mathsf{C}_{x^*,\mathsf{m}}$:

$$w \longmapsto \begin{cases} f(w^*) & \text{if } (x^*, w) \in \mathcal{R} \\ \perp & \text{otherwise} \end{cases}.$$

Here f is given by the adversary for KDM queries. With the correct secret key w^*, one can decrypt and get back m by the functionality of $\mathsf{C}_{x^*,\mathsf{m}}$.

At the first glance, this does not solve our problem, since $\mathsf{C}_{x^*,\mathsf{m}}$ depends on $\mathsf{m} = f(w^*)$ which still depends on w^*. Namely, m is hardcoded in $\mathsf{C}_{x^*,\mathsf{m}}$. But if w^* is unique, we can use the security of obfuscation to switch this to an obfuscation of the circuit $\mathsf{C}_{x^*,f}$:

$$w \longmapsto \begin{cases} f(w) & \text{if } (x^*, w) \in \mathcal{R} \\ \perp & \text{otherwise} \end{cases}.$$

Now the secret key w^* is not hardcoded in C_{x^*} anymore. Instead, it is provided by the decryptor. Using the computational indistinguishability between $\mathcal{L}_\mathcal{R}$ and $\mathcal{X} \setminus \mathcal{L}_\mathcal{R}$ and the obfuscation security again, we can switch this circuit C_{x^*} to a circuit that always returns $\perp$.

MKDM SECURITY FOR IBE. In our generic construction we transfer this idea to the IBE setting. However, we encounter another dilemma: On the one hand, it is natural to let w^* of the above construction be the master secret key, in order to achieve mKDM security. On the other hand, in an IBE's decryption, no master secret key, but only the user secret key, is given. In fact, the above approach can only give us a user-key KDM secure IBE, but we need some novel insights for mKDM security.

Our approach is to embed the master secret key msk into every user secret key in some encrypted form, and the ciphertext is an obfuscated circuit that outputs m ($\mathsf{m} = f(\mathsf{msk})$ for KDM queries) if a user secret key is a valid one, namely, includes an encrypted msk. The validity of a user secret key is guaranteed by a NIZK proof.

In our security proof, we switch this obfuscated circuit to a circuit that first checks the validity of the NIZK proof and then decrypts and gets the msk to simulate KDM queries for challenge identities id^*. To conclude the security, we also need to remove the information about msk in user secret key queries for identities different to id^*. Therefore, the aforementioned encrypted form of msk needs to be implemented carefully together with the NIZK proof, otherwise, we may encounter a problem where we need to extract msk and simulate the proof simultaneously.

Our strategy to solve the above problem can be viewed as an identity-based extractable NIZK proof system. More precisely, a user secret key of an identity id contains the user secret key of the underlying IND-CPA secure IBE and a NIZK proof showing that this user secret key is generated under msk which is the witness. This NIZK proof is an identity-based extractable NIZK. Such a proof system has the following "all-but-many" property: For all identities except the many challenge ones, the proofs can be perfectly simulated without witness; and for the many challenge identities, the proofs are extractable. We implement this proof system using another IND-CPA secure IBE and a dual mode NIZK system.

As an extension, we also show how to obtain mKDM-CCA security generically from mKDM-CPA security in the standard model using a variant of Naor-Yung transformation [NY90, CCS09] for public key encryption. To do so, we encrypt the message under the mKDM-CPA secure scheme and under a public key encryption scheme, and show the consistency of both ciphertexts using a simulation-sound NIZK.

2 Preliminaries

By $\mathbb{N}, \mathbb{P}, \mathbb{R}, \mathbb{Z}, \mathbb{Z}_q, \{0,1\}^*$ we denote sets of natural numbers, primes, real numbers, integers, integers modulo $q \in \mathbb{N}$ and bit strings, respectively. By $[n] := \{1, \ldots, n\}$ we denote the set of the first n natural numbers. The security parameter is denoted by $\lambda \in \mathbb{N}$ and all algorithms will get 1^λ implicitly as input. We say that a probabilistic algorithm A is PPT (probabilistic polynomial time) if its running time $\mathbf{T}(\mathsf{A})$ is bounded by a polynomial in its input size. We use asymptotic notation for positive functions such as ω and O. A function $\nu : \mathbb{N} \to \mathbb{R}$ is negligible in its input λ if $\nu \in \lambda^{-\omega(1)}$ and $\mathsf{negl}(\lambda)$ denotes a negligible function. Conversely, a function ν with $\nu \geq 1 - \mathsf{negl}(\lambda)$ is said to be overwhelming. We write $x \leftarrow \mathcal{D}$ to state that x is sampled from a distribution $\mathcal{D}$. For a finite set S the expression $x \xleftarrow{\$} S$ states that x is sampled from the uniform distribution over S. If the statistical distance between two distributions is negligible in λ, we say they are statistically close. We treat probabilistic algorithms A on an input x as a distribution and write $y \leftarrow \mathsf{A}(x)$ accordingly. If we make the randomness used by an algorithm explicit we will write $y = \mathsf{A}(x; r)$ for randomness $r \in \{0,1\}^*$. The notation $y \in \mathsf{A}(x)$ means that y is a possible output of A on input x. In all security games, numerical values are assumed to be implicitly initialized as 0, sets, lists and associative array as $\emptyset$. The symbol $\perp$ indicates an uninitialized value or the output of an algorithm if it aborts. For a game $\mathbf{G}$, we write $\mathbf{G}^{\mathcal{A}}(\lambda) \Rightarrow b$ to state that the game $\mathbf{G}$ outputs $b \in \{0,1\}$ considering adversary $\mathcal{A}$ and security parameter λ. We will now introduce cryptographic primitives that are relevant for this work.

<u>Public Key Encryption.</u> We give the standard definition of public key encryption and its security.

Definition 2.1 (Public Key Encryption Scheme). A public key encryption scheme (PKE) is defined as a tuple of PPT algorithms $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where

- $\mathsf{Gen}(1^\lambda)$ takes as input the security parameter λ and outputs a public key pk and a secret key sk. We assume that pk implicitly defines a message space $\mathcal{M} = \mathcal{M}_{\mathsf{pk}}$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ takes as input a public key pk and a message $\mathsf{m} \in \mathcal{M}$ and outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ is deterministic, takes as input a secret key sk and ciphertext ct and outputs a message $\mathsf{m} \in \mathcal{M}$.

We say that PKE is ρ-complete, if for every $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Setup}(1^\lambda), \mathsf{m} \in \mathcal{M}$ we have

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{m} \mid \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})\right] \geq \rho.$$

If $\rho = 1$, we say that PKE is perfectly complete.

Definition 2.2 (IND Security of PKE). Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public key encryption scheme. Consider games $\mathbf{IND\text{-}CPA}_b$ for $b \in \{0,1\}$ given in Figure 1. We say that PKE is $\mathsf{IND\text{-}CPA}$ secure, if for every PPT adversary $\mathcal{A}$ the following advantage is negligible in λ:

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) := \left|\Pr\left[\mathbf{IND\text{-}CPA}^{\mathcal{A}}_{0,\mathsf{PKE}}(\lambda) \Rightarrow 1\right] - \Pr\left[\mathbf{IND\text{-}CPA}^{\mathcal{A}}_{1,\mathsf{PKE}}(\lambda) \Rightarrow 1\right]\right|.$$

Game $\mathbf{IND\text{-}CPA}^{\mathcal{A}}_{b,\mathsf{PKE}}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
$(St, \mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(\mathsf{pk})$
if $|\mathsf{m}_0| \neq |\mathsf{m}_1|$: **return** 0
$\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$
return $b' \leftarrow \mathcal{A}(St, \mathsf{ct}^*)$

Figure 1: The games $\mathbf{IND\text{-}CPA}_b$ for bit $b \in \{0,1\}$, public key encryption scheme PKE and adversary $\mathcal{A}$.

<u>Indistinguishability Obfuscation.</u> We introduce indistinguishability obfuscation [BGI+01, GGH+13, JLS21].

Definition 2.3 (Indistinguishability Obfuscator). We call a PPT algorithm iO an indistinguishability obfuscator for polynomial size circuit class $\mathcal{C} = \{\mathcal{C}_\lambda\}_\lambda$ if $\mathsf{iO}(\mathsf{C})$ takes as input a circuit $\mathsf{C} \in \mathcal{C}_\lambda$ and outputs a circuit $\hat{\mathsf{C}}$, such that

- **Preserved Functionality:** For every $\mathsf{C} \in \mathcal{C}_\lambda$ with input length z, all $x \in \{0,1\}^z$, all $\hat{\mathsf{C}} \in \mathsf{iO}(\mathsf{C})$ we have $\mathsf{C}(x) = \hat{\mathsf{C}}(x)$.

Game $\mathbf{IODIST}^{\mathcal{A}}_{b,\mathsf{iO}}(\lambda)$

$(St, \mathsf{C}_0, \mathsf{C}_1) \leftarrow \mathcal{A}(1^\lambda)$
if $\mathsf{C}_0 \notin \mathcal{C}_\lambda \vee \mathsf{C}_1 \notin \mathcal{C}_\lambda$: **return** 0
if $|\mathsf{C}_0| \neq |\mathsf{C}_1|$: **return** 0
if $\exists x \in \{0,1\}^* : \mathsf{C}_0(x) \neq \mathsf{C}_1(x)$: **return** 0
$\hat{\mathsf{C}}_0 \leftarrow \mathsf{iO}(\mathsf{C}_0), \hat{\mathsf{C}}_1 \leftarrow \mathsf{iO}(\mathsf{C}_1)$
return $b' \leftarrow \mathcal{A}(St, \hat{\mathsf{C}}_b)$

Figure 2: The games $\mathbf{IODIST}_b$ for bit $b \in \{0,1\}$, an obfuscator iO and an adversary $\mathcal{A}$.

- **Security:** For every PPT algorithm $\mathcal{A}$ the following advantage is negligible in λ:

$$\mathsf{Adv}^{\mathsf{iodist}}_{\mathcal{A},\mathsf{iO}}(\lambda) := \left|\Pr\left[\mathbf{IODIST}^{\mathcal{A}}_{0,\mathsf{iO}}(\lambda) \Rightarrow 1\right] - \Pr\left[\mathbf{IODIST}^{\mathcal{A}}_{1,\mathsf{iO}}(\lambda) \Rightarrow 1\right]\right|,$$

  where the games $\mathbf{IODIST}_b$ for $b \in \{0,1\}$ are given in Figure 2.

<u>Non-interactive Zero-Knowledge.</u> The definition of non-interactive zero-knowledge proof systems and their properties follows [Gro06, GS08, GHKP18]. For any binary relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$, we define the corresponding language $\mathcal{L}_{\mathcal{R}} \subseteq \{0,1\}^*$ via

$$x \in \mathcal{L}_{\mathcal{R}} \Longleftrightarrow \exists w \in \{0,1\}^* : (x,w) \in \mathcal{R},$$

for all $x \in \{0,1\}^*$. If membership in $\mathcal{R}$ is efficiently (i.e. polynomial time) decidable and there is a polynomial p such that for all $(x,w) \in \mathcal{R}_{\mathcal{L}}$ we have $|w| \leq p(|x|)$, we say that $\mathcal{R}$ is an **NP** relation. In this case, we clearly have $\mathcal{L}_{\mathcal{R}} \in \mathbf{NP}$, which motivates the terminology. We assume that languages and relations implicitly depend on the security parameter, with the restriction that there exists some polynomial poly, such that for any $x \in \mathcal{L}_{\mathcal{R}}$ we have $|x| \leq \mathsf{poly}(\lambda)$.

Definition 2.4 (Non-interactive Zero-Knowledge Proof System). Let $\mathcal{R} = \{\mathcal{R}_\lambda\}$ be an **NP** relation. A $(\rho, \varepsilon_{\mathsf{so}}, \varepsilon_{\mathsf{zk}})$-non-interactive zero-knowledge (NIZK) proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$ for $\mathcal{R}$ is a tuple of PPT algorithms, where

- $\mathsf{PGen}(1^\lambda)$ takes as input the security parameter λ and outputs a common reference string crs. We assume that crs implicitly defines a proof space $\mathcal{P} = \mathcal{P}_{\mathsf{crs}}$.
- $\mathsf{PTrapGen}(1^\lambda)$ has the same syntax as PGen, but additionally outputs a trapdoor td.
- $\mathsf{PProve}(\mathsf{crs}, x, w)$ takes as input the common reference string crs, a statement x and a witness w and outputs a proof π.
- $\mathsf{PVer}(\mathsf{crs}, x, \pi)$ takes as input the common reference string crs, a statement x and a proof π and outputs a bit $b \in \{0,1\}$.
- $\mathsf{PSim}(\mathsf{crs}, \mathsf{td}, x)$ takes as input the common reference string crs, a trapdoor td and a statement x and outputs a proof π.

We require the scheme to be perfectly complete and sound in the following sense:

- **Completeness:** For all $\mathsf{crs} \in \mathsf{PGen}(1^\lambda)$ and all $(x,w) \in \mathcal{R}$ it holds that

$$\Pr\left[\mathsf{PVer}(\mathsf{crs}, x, \pi) = 1 \mid \pi \leftarrow \mathsf{PProve}(\mathsf{crs}, x, w)\right] \geq \rho.$$

  If $\rho = 1$, we say that PS is perfectly complete and omit ρ.

- **Soundness:** For all (not necessarily efficient) adversaries $\mathcal{A}$ we have

$$\Pr\left[\mathsf{PVer}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin \mathcal{L}_{\mathcal{R}} \mid \mathsf{crs} \leftarrow \mathsf{PGen}(1^\lambda), (x,\pi) \leftarrow \mathcal{A}(\mathsf{crs})\right] \leq \varepsilon_{\mathsf{so}}.$$

  If $\varepsilon_{\mathsf{so}} = 0$, we say that PS is perfectly sound and omit $\varepsilon_{\mathsf{so}}$.

We also require the following zero-knowledge properties to hold:

- **CRS Indistinguishability:** For all PPT algorithms $\mathcal{A}$ the following advantage is negligible in λ:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{A},\mathsf{PS}}(\lambda) := |\Pr\left[\mathcal{A}(\mathsf{crs}) = 1 \mid \mathsf{crs} \leftarrow \mathsf{PGen}(1^\lambda)\right] \\ -\Pr\left[\mathcal{A}(\mathsf{crs}) = 1 \mid (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda)\right]|.\end{aligned}$$

- **Zero-Knowledge:** For all $(x, w) \in \mathcal{R}$ the following distributions have statistical distance at most $\varepsilon_{\mathsf{zk}}$:

$$\{(\pi, \mathsf{crs}, \mathsf{td}) \mid (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda),\ \pi \leftarrow \mathsf{PProve}(\mathsf{crs}, x, w)\}$$

  and

$$\{(\pi, \mathsf{crs}, \mathsf{td}) \mid (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda),\ \pi \leftarrow \mathsf{PSim}(\mathsf{crs}, \mathsf{td}, x)\}.$$

We also define the stronger notion of simulation soundness. However, we do not need it all the time, which is why we give it in a separate definition.

Definition 2.5 (Simulation Soundness). Let $\mathcal{R} = \{\mathcal{R}_\lambda\}$ be an **NP** relation and $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$ an $(\rho, \varepsilon_{\mathsf{so}}, \varepsilon_{\mathsf{zk}})$-NIZK proof system for $\mathcal{R}$. Consider the game **SIMSO** in Figure 3. We say that PS is $\varepsilon_{\mathsf{sso}}$-simulation-sound if for any (not necessarily efficient) adversary $\mathcal{A}$ we have

$$\Pr\left[\mathbf{SIMSO}^{\mathcal{A}}_{\mathsf{PS}} \Rightarrow 1\right] \le \varepsilon_{\mathsf{sso}}.$$

Game $\mathbf{SIMSO}^{\mathcal{A}}_{\mathsf{PS}}(\lambda)$

$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda),\ (x, \pi) \leftarrow \mathcal{A}^{\textsc{Sim}}(\mathsf{crs})$
if $(x, \pi) \notin \mathcal{L} \wedge\ x \notin \mathcal{L}_{\mathcal{R}} \wedge\ \mathsf{PVer}(\mathsf{crs}, x, \pi) = 1$:
  return 1
return 0

Oracle $\textsc{Sim}(x)$

$\pi \leftarrow \mathsf{PSim}(\mathsf{crs}, \mathsf{td}, x)$
$\mathcal{L} := \mathcal{L} \cup \{(x, \pi)\}$
return π

Figure 3: The game **SIMSO** for a NIZK proof system PS and an adversary $\mathcal{A}$.

Attribute-Based Encyption. We define attribute-based encryption (ABE) and different security notions for it. We remark that we define all security notions in the multi-challenge setting. For $\mathsf{IND\text{-}CPA}$ and $\mathsf{OW\text{-}CPA}$, this notion is implied by the single-challenge setting, using a standard hybrid argument.

Definition 2.6 (Attribute-Based Encryption Scheme). Let $\mathcal{X} = \mathcal{X}_\lambda$ and $\mathcal{Y} = \mathcal{Y}_\lambda$ be two (families of) sets, and $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ be an efficiently computable predicate on $\mathcal{X}, \mathcal{Y}$. An attribute-based encryption scheme (ABE) for $\mathcal{P}$ is a tuple of PPT algorithms $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$, where

- $\mathsf{Setup}(1^\lambda)$ takes as input the security parameter λ and outputs a master public key mpk and a master secret key msk. We assume that mpk implicitly defines a message space $\mathcal{M} = \mathcal{M}_{\mathsf{mpk}}$, and an user-secret key space $\mathcal{K} = \mathcal{K}_{\mathsf{mpk}}$.
- $\mathsf{KeyExt}(\mathsf{msk}, \mathsf{y})$ takes as input a master secret key msk and an attribute $\mathsf{y} \in \mathcal{Y}$ and outputs a secret key $\mathsf{sk_y} \in \mathcal{K}$. We assume that $\mathsf{sk_y}$ implicitly contains y.
- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})$ takes as input a master public key mpk, an attribute $\mathsf{x} \in \mathcal{X}$ and a message $\mathsf{m} \in \mathcal{M}$ and outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk_y}, \mathsf{ct})$ is deterministic, takes as input a user secret key $\mathsf{sk_y} \in \mathcal{K}$ and ciphertext ct and outputs a message $\mathsf{m} \in \mathcal{M}$.

We say that ABE is ρ-complete, if for every $(\mathsf{mpk}, \mathsf{msk}) \in \mathsf{Setup}(1^\lambda), \mathsf{m} \in \mathcal{M}, \mathsf{x} \in \mathcal{X}, \mathsf{y} \in \mathcal{Y}$ with $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$, we have

$$\Pr\left[\mathsf{Dec}(\mathsf{sk_y}, \mathsf{ct}) = \mathsf{m} \mid \mathsf{sk_y} \leftarrow \mathsf{KeyExt}(\mathsf{msk}, \mathsf{id}), \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})\right] \ge \rho.$$

If $\rho = 1$, we say that ABE is perfectly complete.

Game $\mathbf{mKDM\text{-}CPA}^{\mathcal{A}}_{b,\mathsf{ABE}}(\lambda)$,
Game $\mathbf{mKDM\text{-}CCA}^{\mathcal{A}}_{b,\mathsf{ABE}}(\lambda)$

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
$\mathrm{O} := (\textsc{Key}, \textsc{Kdm}_b)$
$\mathrm{O} := (\textsc{Key}, \textsc{Kdm}_b, \textsc{Dec})$
return $b' \leftarrow \mathcal{A}^{\mathrm{O}}(\mathsf{mpk})$

Oracle $\textsc{Dec}(\mathsf{y}, \mathsf{ct})$

if $\exists \mathsf{x} \in \mathcal{X}$ s.t. $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1 \wedge (\mathsf{x}, \mathsf{ct}) \in \mathcal{L}_{ct}$:
  return $\perp$
$\mathsf{sk_y} \leftarrow \mathsf{KeyExt}(\mathsf{msk}, \mathsf{y})$
return $\mathsf{Dec}(\mathsf{sk_y}, \mathsf{ct})$

Oracle $\textsc{Key}(\mathsf{y})$

if $\mathsf{hit}_{\mathcal{P}}(\mathcal{L}_{ch}, \{\mathsf{y}\})$: **return** $\perp$
$\mathcal{L}_{sk} := \mathcal{L}_{sk} \cup \{\mathsf{y}\}$
return $\mathsf{sk_y} \leftarrow \mathsf{KeyExt}(\mathsf{msk}, \mathsf{y})$

Oracle $\textsc{Kdm}_b(\mathsf{x}, f \in \mathcal{F})$

if $\mathsf{hit}_{\mathcal{P}}(\{\mathsf{x}\}, \mathcal{L}_{sk})$: **return** $\perp$
$\mathcal{L}_{ch} := \mathcal{L}_{ch} \cup \{\mathsf{x}\}$
$\mathsf{m}_0 := 0^{|f(\cdot)|}, \mathsf{m}_1 := f(\mathsf{msk})$
$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m}_b)$
$\mathcal{L}_{ct} := \mathcal{L}_{ct} \cup \{(\mathsf{x}, \mathsf{ct})\}$
return ct

Figure 4: The games $\mathbf{mKDM\text{-}CPA}_b, \mathbf{mKDM\text{-}CCA}_b$ for bit $b \in \{0,1\}$, an attribute-based encryption scheme ABE for predicate $\mathcal{P}$, and an adversary $\mathcal{A}$. The shaded statement is only executed in game $\mathbf{mKDM\text{-}CCA}_b$.

The above notion captures both ciphertext-policy (CP) and key-policy (KP) ABE. For example, KP-ABE for a class of policies $\{P\}$ is obtained by considering the universal predicate $\mathcal{P}(\mathsf{x}, P) = P(\mathsf{x})$.

Definition 2.7 (Smoothness of ABE). Consider an attribute-based encryption scheme $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$. We say that ABE is ε-smooth if we have

$$\mathbb{E}_{(\mathsf{mpk},\mathsf{msk})\leftarrow\mathsf{Setup}(1^\lambda)}\left[\max_{\mathsf{x},\mathsf{m},\mathsf{ct}'} \Pr[\mathsf{ct} = \mathsf{ct}' \mid \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})]\right] \leq \varepsilon.$$

To improve readability of our security games, we introduce a predicate hit. Informally, it extends the predicate $\mathcal{P}$ to lists and sets.

Definition 2.8 (List Predicate of ABE). Consider a predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. We define the predicate $\mathsf{hit}_{\mathcal{P}} : 2^{\mathcal{X}} \times 2^{\mathcal{Y}} \to \{0,1\}$ as follows:

$$\mathsf{hit}_{\mathcal{P}}(\mathcal{L}_{\mathsf{x}}, \mathcal{L}_{\mathsf{y}}) = 1 \iff \exists \mathsf{x} \in \mathcal{L}_{\mathsf{x}}, \mathsf{y} \in \mathcal{L}_{\mathsf{y}} : \mathcal{P}(\mathsf{x}, \mathsf{y}) = 1.$$

Definition 2.9 (mKDM Security of ABE). Let $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ be an attribute-based encryption scheme with master secret key space $\mathcal{K}_{\mathrm{m}}$ for a predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. Let $\mathcal{F}$ be a class of efficiently computable functions with domain $\mathcal{K}_{\mathrm{m}}$. We assume that the range of $\mathcal{F}$ is a subset of the message space $\mathcal{M}$ of ABE. Consider games $\mathbf{mKDM\text{-}CPA}_b, \mathbf{mKDM\text{-}CCA}_b$ for $b \in \{0,1\}$ given in Figure 4. We say that ABE is $\mathcal{F}$-$\mathsf{mKDM\text{-}CPA}$ secure, if for every PPT adversary $\mathcal{A}$ the following advantage is negligible in λ:

$$\mathsf{Adv}^{\mathsf{mKDM\text{-}CPA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) := \left|\Pr\left[\mathbf{mKDM\text{-}CPA}^{\mathcal{A}}_{0,\mathsf{ABE}}(\lambda) \Rightarrow 1\right] - \Pr\left[\mathbf{mKDM\text{-}CPA}^{\mathcal{A}}_{1,\mathsf{ABE}}(\lambda) \Rightarrow 1\right]\right|.$$

We say that ABE is $\mathcal{F}$-$\mathsf{mKDM\text{-}CCA}$ secure, if for every PPT adversary $\mathcal{A}$ the following advantage is negligible in λ:

$$\mathsf{Adv}^{\mathsf{mKDM\text{-}CCA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) := \left|\Pr\left[\mathbf{mKDM\text{-}CCA}^{\mathcal{A}}_{0,\mathsf{ABE}}(\lambda) \Rightarrow 1\right] - \Pr\left[\mathbf{mKDM\text{-}CCA}^{\mathcal{A}}_{1,\mathsf{ABE}}(\lambda) \Rightarrow 1\right]\right|.$$

Definition 2.10 (IND Security of ABE). Let $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ be an attribute-based encryption scheme for a predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. Consider games $\mathbf{IND\text{-}CPA}_b$ for $b \in \{0,1\}$ given in Figure 5. We say that ABE is $\mathsf{IND\text{-}CPA}$ secure, if for every PPT adversary $\mathcal{A}$ the following advantage is negligible in λ:

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) := \left|\Pr\left[\mathbf{IND\text{-}CPA}^{\mathcal{A}}_{0,\mathsf{ABE}}(\lambda) \Rightarrow 1\right] - \Pr\left[\mathbf{IND\text{-}CPA}^{\mathcal{A}}_{1,\mathsf{ABE}}(\lambda) \Rightarrow 1\right]\right|.$$

Game $\mathbf{IND\text{-}CPA}^{\mathcal{A}}_{b,\mathsf{ABE}}(\lambda)$	**Game** $\mathbf{OW\text{-}CPA}^{\mathcal{A}}_{\mathsf{ABE}}(\lambda)$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
return $b' \leftarrow \mathcal{A}^{\text{KEY},\text{CH}_b}(\mathsf{mpk})$	$\mathcal{L}_{ans} \leftarrow \mathcal{A}^{\text{KEY},\text{CH}}(\mathsf{mpk})$
	return $\mathcal{L}_{ans} \cap \mathcal{L}_{pt} \neq \emptyset$
Oracle $\text{CH}_b(\mathsf{x}, \mathsf{m}_0, \mathsf{m}_1)$	**Oracle** $\text{CH}(\mathsf{x})$
if $\mathsf{hit}_{\mathcal{P}}(\{\mathsf{x}\}, \mathcal{L}_{sk})$: **return** $\perp$	**if** $\mathsf{hit}_{\mathcal{P}}(\{\mathsf{x}\}, \mathcal{L}_{sk})$: **return** $\perp$
if $\lvert\mathsf{m}_0\rvert \neq \lvert\mathsf{m}_1\rvert$: **return** $\perp$	$\mathcal{L}_{ch} := \mathcal{L}_{ch} \cup \{\mathsf{x}\}$
$\mathcal{L}_{ch} := \mathcal{L}_{ch} \cup \{\mathsf{id}\}$	$\mathsf{m} \xleftarrow{\$} \mathcal{M}, \mathcal{L}_{pt} := \mathcal{L}_{pt} \cup \{(\mathsf{x}, \mathsf{m})\}$
return $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m}_b)$	**return** $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})$

Figure 5: The games $\mathbf{IND\text{-}CPA}_b$ (left) and **OW-CPA** (right) for bit $b \in \{0,1\}$, an attribute-based encryption scheme ABE for predicate $\mathcal{P}$, and an adversary $\mathcal{A}$. Oracle KEY is defined exactly as in Figure 4.

Definition 2.11 (OW Security of ABE). Let $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ be an attribute-based encryption scheme for a predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. Consider game **OW-CPA** given in Figure 5. We say that ABE is $\mathsf{OW\text{-}CPA}$ secure, if for every PPT adversary $\mathcal{A}$ the following advantage is negligible in λ:

$$\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) := \Pr\left[\mathbf{OW\text{-}CPA}^{\mathcal{A}}_{\mathsf{ABE}}(\lambda) \Rightarrow 1\right].$$

3 Generic Construction in the Random Oracle Model

In this section, we construct two attribute-based encryption schemes, which are $\mathsf{mKDM\text{-}CPA}$ and $\mathsf{mKDM\text{-}CCA}$ secure. We use hybrid encryption to transform any $\mathsf{OW\text{-}CPA}$ secure ABE into an $\mathsf{mKDM\text{-}CPA}$ secure one. Then we show that using an attribute-based variant of the Fujisaki-Okamoto transform [FO99], we can construct an $\mathsf{mKDM\text{-}CCA}$ secure ABE from any $\mathsf{OW\text{-}CPA}$ secure ABE. An overview of the results in this section is given in Figure 6.

$$\mathsf{ABE} : \mathsf{OW\text{-}CPA} \xrightarrow{\text{Sec. 3.1}} \mathsf{ABE_H} : \mathsf{mKDM\text{-}CPA}$$
$$\mathsf{ABE} : \mathsf{OW\text{-}CPA} \xrightarrow{\text{Sec. 3.2}} \mathsf{ABE_{FO}} : \mathsf{mKDM\text{-}CCA}$$

Figure 6: Overview of our construction of $\mathsf{mKDM\text{-}CPA}$ and $\mathsf{mKDM\text{-}CCA}$ secure attribute-based encryption in the random oracle model. We transform any $\mathsf{OW\text{-}CPA}$ secure attribute-based encryption scheme into $\mathsf{mKDM\text{-}CPA}$ and $\mathsf{mKDM\text{-}CCA}$ secure schemes.

3.1 mKDM-CPA Secure ABE via Hybrid Encryption

In this section, we construct an $\mathsf{mKDM\text{-}CPA}$ secure attribute-based encryption scheme $\mathsf{ABE_H}$ with message space $\mathcal{M} = \{0,1\}^\ell$ for predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ from a $\mathsf{OW\text{-}CPA}$ secure attribute-based encryption scheme ABE with message space $\mathcal{M}$ for predicate $\mathcal{P}$ using a random oracle $\mathsf{H} : \mathcal{X} \times \{0,1\}^\ell \to \{0,1\}^\ell$ via hybrid encryption. The construction is presented in Figure 7. Completeness of $\mathsf{ABE_H}$ immediately follows from the completeness of ABE.

Alg $\mathsf{Enc_H}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})$	**Alg** $\mathsf{Dec_H}(\mathsf{sk_y}, \mathsf{ct} = (c, d))$
$r \xleftarrow{\$} \{0,1\}^\ell,\ K := \mathsf{H}(r)$	$r := \mathsf{Dec}(\mathsf{sk_y}, c)$
$c \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, r)$	$K := \mathsf{H}(r)$
return $\mathsf{ct} := (c, d := K \oplus \mathsf{m})$	**return** $\mathsf{m} := K \oplus d$

Figure 7: The attribute-based encryption scheme $\mathsf{ABE_H} = (\mathsf{Setup_H} := \mathsf{Setup}, \mathsf{KeyExt_H} := \mathsf{KeyExt}, \mathsf{Enc_H}, \mathsf{Dec_H})$ for a given attribute-based encryption scheme $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ and a random oracle H.

Theorem 3.1 ($\mathsf{mKDM\text{-}CPA}$ **Security of** $\mathsf{ABE_H}$)**.** *Let $\mathcal{F}$ be a class of efficiently computable functions with oracle access to* H*. If* ABE *is a* $\mathsf{OW\text{-}CPA}$ *secure attribute-based encryption scheme, then* $\mathsf{ABE_H}$ *given in Figure 7 is* $\mathcal{F}$-$\mathsf{mKDM\text{-}CPA}$ *secure in the random oracle model. More precisely, for every PPT algorithm $\mathcal{A}$ making Q_C, Q_H queries to the oracles* $\textsc{Kdm}, \mathsf{H}$*, respectively, there exists a PPT algorithm $\mathcal{B}$ with $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and*

$$\mathsf{Adv}^{\mathsf{mKDM\text{-}CPA}}_{\mathcal{A},\mathsf{ABE_H}}(\lambda) \leq \frac{Q_C \cdot Q_H}{2^{\ell-1}} + 2 \cdot \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda).$$

Proof. First, as in [KMHT16], during the security game, random oracle H is called in one of the following three cases:

- publicly called by the adversary,
- privately called by the game in the $\textsc{Kdm}$ oracle,
- privately called by the game while computing the function f^{H} in the $\textsc{Kdm}$ oracle.

We will gradually separate these three different cases of the random oracle through the security games. We denote them by $\mathsf{H}, \mathsf{H}^\star, \hat{\mathsf{H}}$, respectively, see Figures 8 and 9. For each game $\mathbf{G}_i$, we denote the probability that it outputs 1 by pr_i, namely,

$$\mathsf{pr}_i := \Pr\left[\mathbf{G}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right].$$

Game $\mathbf{G}_0$: This game is the $\mathsf{mKDM\text{-}CPA}$ security game that always encrypts $f^{\hat{\mathsf{H}}}(\mathsf{msk})$. We make the conceptual modification that the answers of the random oracle queries are separated into $\mathsf{H}, \mathsf{H}^\star, \hat{\mathsf{H}}$. More specifically, the random oracles H and $\mathsf{H}^\star$ maintain two lists for the hash values (r, K). They are also sharing access of their lists in the sense that every time r' is queried to H (resp. $\mathsf{H}^\star$), if there is already a K' such that $(r', K') \in \mathcal{L}_{\mathsf{H}} \cup \mathcal{L}_{\mathsf{H}^\star}$, then it returns K'. The random oracle $\hat{\mathsf{H}}$ does not maintain any list, it behaves exactly as H. The detailed behavior of oracles $\textsc{Kdm}, \mathsf{H}, \mathsf{H}^\star, \hat{\mathsf{H}}$ is given in Figure 8. Note that the

Oracle $\textsc{Kdm}(\mathsf{x}, f)$
$\mathsf{m} := f^{\hat{\mathsf{H}}}(\mathsf{msk})$
$r \xleftarrow{\$} \{0,1\}^\ell$
$K := \mathsf{H}^\star(r)$
$c \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, r)$
return $\mathsf{ct} := (\mathsf{x}, c, K \oplus \mathsf{m})$

Oracle $\mathsf{H}^\star(r)$
if $\exists K : (r, K) \in \mathcal{L}_{\mathsf{H}} \cup \mathcal{L}_{\mathsf{H}^\star}$:
  return K
else :
  $K \xleftarrow{\$} \{0,1\}^\ell$
  $\mathcal{L}_{\mathsf{H}^\star} := \mathcal{L}_{\mathsf{H}^\star} \cup \{(r, K)\}$
  return K

Oracle $\mathsf{H}(r)$
if $\exists K : (r, K) \in \mathcal{L}_{\mathsf{H}} \cup \mathcal{L}_{\mathsf{H}^\star}$:
  return K
else :
  $K \xleftarrow{\$} \{0,1\}^\ell$
  $\mathcal{L}_{\mathsf{H}} := \mathcal{L}_{\mathsf{H}} \cup \{(r, K)\}$
  return K

Oracle $\hat{\mathsf{H}}(r)$
if $\exists K : (r, K) \in \mathcal{L}_{\mathsf{H}} \cup \mathcal{L}_{\mathsf{H}^\star}$:
  return K
else :
  $K \xleftarrow{\$} \{0,1\}^\ell$
  $\mathcal{L}_{\mathsf{H}} := \mathcal{L}_{\mathsf{H}} \cup \{(r, K)\}$
  return K

Figure 8: The description of the oracles $\textsc{Kdm}$, H, $\mathsf{H}^\star$ and $\hat{\mathsf{H}}$ in game $\mathbf{G}_0$ in the proof of Theorem 3.1.

only difference between $\mathbf{G}_0$ and the game $\mathbf{mKDM\text{-}CPA}_1$ is that we have conceptually differentiated the hash queries to $\mathsf{H}^\star$ from the ones to H and $\hat{\mathsf{H}}$. Therefore we have

$$\mathsf{pr}_0 = \Pr\Big[\mathbf{mKDM\text{-}CPA}^{\mathcal{A}}_{1,\mathsf{ABE_H}}(\lambda) \Rightarrow 1\Big].$$

Game $\mathbf{G}_1$: In game $\mathbf{G}_1$, we modify the behavior of the random oracle $\mathsf{H}^\star$ in the following way: Every query $\mathsf{H}^\star(r)$ is answered with a freshly generated K. Then, $\mathsf{H}^\star$ adds the pair (r, K) to the list $\mathcal{L}_{\mathsf{H}^\star}$. Note that H and $\hat{\mathsf{H}}$ still need $\mathcal{L}_{\mathsf{H}^\star}$ to answer queries. Given r, if there exist multiple values K such that $(r, K) \in \mathcal{L}_{\mathsf{H}^\star}$, H and $\hat{\mathsf{H}}$ take the first entry as the random oracle's output. The detailed behavior of $\mathsf{H}^\star$ is given in Figure 9.

We define the event COL that when $\textsc{Kdm}(\mathsf{x}, f)$ generates $r \xleftarrow{\$} \{0,1\}^\ell$, there already exists an entry of the form $(r, \cdot)$ in the list $\mathcal{L}_\mathsf{H} \cup \mathcal{L}_{\mathsf{H}^\star}$. We can see that $\mathbf{G}_1$ is different from $\mathbf{G}_0$ only if COL happens. Further, there are at most Q_H many entries in the list $\mathcal{L}_\mathsf{H} \cup \mathcal{L}_{\mathsf{H}^\star}$. Moreover, since in each query $\textsc{Kdm}(\mathsf{x}, f)$ the value r is uniformly chosen at random $r \xleftarrow{\$} \{0,1\}^\ell$, the probability that COL happens in such a query is at most $Q_H/2^\ell$. By the union bound over all the $\textsc{Kdm}$ queries, we have

$$|\mathsf{pr}_0 - \mathsf{pr}_1| \le \Pr[\mathsf{COL}] \le \frac{Q_C \cdot Q_H}{2^\ell}.$$

Game $\mathbf{G}_2$: In game $\mathbf{G}_2$, the only difference compared to $\mathbf{G}_1$ is that the simulation of H does not refer to list $\mathcal{L}_{\mathsf{H}^\star}$ anymore. However, $\hat{\mathsf{H}}$ still refers to $\mathcal{L}_\mathsf{H}$ and $\mathcal{L}_{\mathsf{H}^\star}$ in this game. The behavior of H is given in Figure 9.

We define the event BHQ_i ("Bad Hash Query") that in $\mathbf{G}_i$ when $\mathcal{A}$ queries r to H, there already exists an entry of the form $(r, \cdot)$ in $\mathcal{L}_{\mathsf{H}^\star}$. Note that $\mathbf{G}_2$ differs from $\mathbf{G}_1$ only if BHQ_2 happens. Therefore, we have

$$|\mathsf{pr}_1 - \mathsf{pr}_2| \le \Pr[\mathsf{BHQ}_2].$$

Game $\mathbf{G}_3$: In game $\mathbf{G}_3$, we modify the behavior of the $\textsc{Kdm}$ queries $\textsc{Kdm}(\mathsf{x}, f)$. It returns the encryption of a uniformly random message of length $|f(\cdot)|$. See Figure 9 for details.

Note that in $\mathbf{G}_3$ every query to $\textsc{Kdm}$ is answered with a randomly generated ciphertext. Further, note that we can do a similar sequence of game transitions starting with the game $\mathbf{mKDM\text{-}CPA}_0$ and end up at the very same game $\mathbf{G}_3$. Thus, by the triangle inequality, it is sufficient to bound $|\mathsf{pr}_0 - \mathsf{pr}_3|$ to finish our proof. To do so, we use the following lemmas to bound $|\mathsf{pr}_2 - \mathsf{pr}_3|$ and $\Pr[\mathsf{BHQ}_2]$.

Oracle $\mathsf{H}^\star(r)$ // $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$

$K \xleftarrow{\$} \{0,1\}^\ell$
$\mathcal{L}_{\mathsf{H}^\star} := \mathcal{L}_{\mathsf{H}^\star} \cup \{(r, K)\}$
return K

Oracle $\textsc{Kdm}(\mathsf{x}, f)$ // $\mathbf{G}_3$

$r \xleftarrow{\$} \{0,1\}^\ell$; $\mathsf{m} \xleftarrow{\$} \{0,1\}^{|f(\cdot)|}$
$K := \mathsf{H}^\star(r)$
$c \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, r)$
return $\mathsf{ct} := (c, K \oplus \mathsf{m})$

Oracle $\mathsf{H}(r)$ // $\mathbf{G}_2, \mathbf{G}_3$

if $\exists K : (r, K) \in \mathcal{L}_\mathsf{H}$:
  return K
else :
  $K \xleftarrow{\$} \{0,1\}^\ell$
  $\mathcal{L}_\mathsf{H} := \mathcal{L}_\mathsf{H} \cup \{(r, K)\}$
  return K

Figure 9: The changes of the oracles $\textsc{Kdm}$, H, $\mathsf{H}^\star$ and $\hat{\mathsf{H}}$ from $\mathbf{G}_1$ to $\mathbf{G}_3$ in the proof of Theorem 3.1.

Lemma 3.2 $|\mathsf{pr}_2 - \mathsf{pr}_3| = 0$.

Proof. Note that the only difference between $\mathbf{G}_3$ and $\mathbf{G}_2$ is the output of the oracle $\textsc{Kdm}$. Suppose that there are Q_C $\textsc{Kdm}$ queries. We will introduce $Q_C + 1$ intermediate hybrids $\mathbf{G}_{2.0}, \ldots, \mathbf{G}_{2.Q_C}$ that gradually change the $\textsc{Kdm}$ oracle answers. Note that, $\mathbf{G}_{2.0}$ will be identical to $\mathbf{G}_2$ and $\mathbf{G}_{2.Q_C}$ will be identical to $\mathbf{G}_3$. The game $\mathbf{G}_{2.i}$ for $i \in [Q_C]$ is defined as follows:

Game $\mathbf{G}_{2.i}$: In this game, the challenger answers the first $Q_C - i$ queries to $\textsc{Kdm}$ as in $\mathbf{G}_2$. From the $(Q_C - i + 1)$-th to the Q_C-th $\textsc{Kdm}$ queries, the challenger answers as in $\mathbf{G}_3$ (described in Figure 9). Namely, the remaining $\textsc{Kdm}$ queries are answered with random ciphertexts.

To bound the success probability of distinguishing $\mathbf{G}_{2.i-1}$ and $\mathbf{G}_{2.i}$, note that the only difference is the behavior of the $(Q_C - i)$-th $\textsc{Kdm}$ query. Due to the change of $\mathsf{H}^\star$ in $\mathbf{G}_1$, $K := \mathsf{H}^\star(r)$ is a freshly generated randomness in $\mathbf{G}_{2.i}$ regardless of whether r has been queried to H or $\mathsf{H}^\star$ before. Moreover, the value K is only stored in $\mathcal{L}_{\mathsf{H}^\star}$. Due to change in $\mathbf{G}_2$, the values stored in $\mathcal{L}_{\mathsf{H}^\star}$ are only accessible by the adversary indirectly via the hash query $\hat{\mathsf{H}}$ as part of the $\textsc{Kdm}$ queries. However, in $\mathbf{G}_{2.i}$, from the $(Q_C - i + 1)$-th to the Q_C-th $\textsc{Kdm}$ queries are all answered without quering $\hat{\mathsf{H}}$. In summary, K is an uniformly generated randomness in the $(Q_C - i)$-th query, and it is not reused afterwards. This implies that K acts as a one-time pad and $K \oplus \mathsf{m}$ returned by the $(Q_C - i)$-th query is statistically identical to a uniformly random element in $\{0,1\}^\ell$. Therefore we have $|\mathsf{pr}_{2.i-1} - \mathsf{pr}_{2.i}| = 0$. By using the triangle inequality over all $Q_C + 1$ hybrids, we have $|\mathsf{pr}_2 - \mathsf{pr}_3| = 0$. □

Lemma 3.3 *There exists an algorithm $\mathcal{B}$ with $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and*

$$\Pr[\mathsf{BHQ}_3] \leq \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda).$$

Proof. Given an adversary $\mathcal{A}$, we construct an adversary $\mathcal{B}$ that wins the OW-CPA security game of the underlying scheme ABE whenever BHQ_3 happens. The construction of $\mathcal{B}$ is given in Figure 10. Firstly,

$\underline{\mathcal{B}^{\text{Key},\text{Ch}}(\mathsf{mpk})}$

$b' \leftarrow \mathcal{A}^{\text{Key},\text{Kdm},\mathsf{H}}(\mathsf{mpk})$

return $\mathcal{L}_{ans}$

$\underline{\textbf{Oracle } \text{Kdm}(\mathsf{x}, f)}$

$\mathsf{m} \xleftarrow{\$} \{0,1\}^{|f(\cdot)|}$; $K \xleftarrow{\$} \{0,1\}^{|f(\cdot)|}$

$c \leftarrow \text{Ch}(\mathsf{x})$

return $\mathsf{ct} := (c, K \oplus \mathsf{m})$

$\underline{\textbf{Oracle } \mathsf{H}(r)}$

if $\exists K : (r, K) \in \mathcal{L}_{\mathsf{H}}$:

  return K

else :

  $\mathcal{L}_{ans} := \mathcal{L}_{ans} \cup \{r\}$

  $K \xleftarrow{\$} \{0,1\}^{\ell}$

  $\mathcal{L}_{\mathsf{H}} := \mathcal{L}_{\mathsf{H}} \cup \{(r, K)\}$

  return K

Figure 10: The reduction $\mathcal{B}$ in the proof of Theorem 3.1. It simulates $\mathbf{G}_3$ for adversary $\mathcal{A}$ to win the OW-CPA security game.

it is straightforward that $\mathcal{B}$ perfectly simulates game $\mathbf{G}_3$ to $\mathcal{A}$. Moreover, the OW-CPA security game maintains a list $\mathcal{L}_{pt}$ which corresponds to all the queries by $\mathsf{H}^\star$ in $\mathbf{G}_3$. If BHQ happens, then there is $(r, K) \in \mathcal{L}_{\mathsf{H}}$ with $r \in \mathcal{L}_{pt}$. Therefore $\mathcal{B}$ is successful. □

We can now summarize as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{mKDM\text{-}CPA}}_{\mathcal{A},\mathsf{ABE_H}}(\lambda) &\leq 2|\mathsf{pr}_0 - \mathsf{pr}_3| \leq 2\left(\frac{Q_C \cdot Q_H}{2^\ell} + \Pr[\mathsf{BHQ}_2]\right) \\
&\leq 2\left(\frac{Q_C \cdot Q_H}{2^\ell} + |\Pr[\mathsf{BHQ}_2] - \Pr[\mathsf{BHQ}_3]| + \Pr[\mathsf{BHQ}_3]\right) \\
&\leq 2\left(\frac{Q_C \cdot Q_H}{2^\ell} + |\Pr[\mathsf{BHQ}_2] - \Pr[\mathsf{BHQ}_3]| + \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda)\right) \\
&\leq \frac{Q_C \cdot Q_H}{2^{\ell-1}} + 2 \cdot \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda),
\end{aligned}$$

where the last inequality follows from $|\Pr[\mathsf{BHQ}_2] - \Pr[\mathsf{BHQ}]_3| \leq |\mathsf{pr}_2 - \mathsf{pr}_3| = 0$. This is because any difference between $\Pr[\mathsf{BHQ}_2]$ and $\Pr[\mathsf{BHQ}_3]$ can be used to distinguish $\mathbf{G}_2$ from $\mathbf{G}_3$. □

3.2 mKDM-CCA Secure ABE via the Fujisaki-Okamoto Transform

In this section, we turn any OW-CPA secure attribute-based encryption scheme ABE with message space $\mathcal{M} = \{0,1\}^\ell$ for predicate $\mathcal{P} \colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ into an mKDM-CCA secure scheme $\mathsf{ABE_{FO}}$ with message space $\mathcal{M}$ for predicate $\mathcal{P}$. We do so using random oracles $\mathsf{H} \colon \mathcal{X} \times \{0,1\}^\ell \to \{0,1\}^\ell, \mathsf{G} \colon \mathcal{X} \times \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^\ell$ following the Fujisaki-Okamoto transform. The scheme is presented in Figure 11. Completeness follows directly from the completeness of ABE. The proof of its mKDM-CCA security is an extension of the proof of Theorem 3.1.

$\underline{\textbf{Alg } \mathsf{Enc_{FO}}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})}$

$r \xleftarrow{\$} \{0,1\}^\ell, K := \mathsf{H}(r)$

$d := K \oplus \mathsf{m}, \rho := \mathsf{G}(r, d)$

$c := \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, r; \rho)$

return $\mathsf{ct} := (c, d)$

$\underline{\textbf{Alg } \mathsf{Dec_{FO}}(\mathsf{sk_y}, \mathsf{ct} = (c, d))}$

$r := \mathsf{Dec}(\mathsf{sk_y}, c), \rho \leftarrow \mathsf{G}(\mathsf{x}, r, d)$

$\mathsf{m} = \mathsf{H}(r) \oplus d$

if $c = \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, r; \rho)$:

  return m

return $\perp$

Figure 11: The attribute-based encryption scheme $\mathsf{ABE_{FO}} = (\mathsf{Setup_{FO}} := \mathsf{Setup}, \mathsf{KeyExt_{FO}} := \mathsf{KeyExt}, \mathsf{Enc_{FO}}, \mathsf{Dec_{FO}})$ for a given attribute-based encryption scheme $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ and random oracles H, G.

Theorem 3.4 ($\mathsf{mKDM\text{-}CCA}$ **Security of** $\mathsf{ABE_{FO}}$)**.** *Let $\mathcal{F}$ be a class of efficiently computable functions with oracle access to H, G. If ABE is a $\mathsf{OW\text{-}CPA}$ secure and ε-smooth identity-based encryption scheme, then $\mathsf{ABE_{FO}}$ given in Figure 11 is $\mathcal{F}$-$\mathsf{mKDM\text{-}CCA}$ secure in the random oracle model. More precisely, for every PPT algorithm $\mathcal{A}$ making Q_C, Q_D, Q_H, Q_G queries to the oracles* $\textsc{Kdm}, \textsc{Dec}, \mathsf{H}, \mathsf{G}$*, respectively, there exists a PPT algorithm $\mathcal{B}$ with $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and*

$$\mathsf{Adv}^{\mathsf{mKDM\text{-}CCA}}_{\mathcal{A},\mathsf{ABE_{FO}}}(\lambda) \leq \frac{Q_C \cdot (Q_G + Q_H)}{2^{\ell-1}} + 4Q_D \cdot \varepsilon + 8 \cdot \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda).$$

Proof. We give the proof via a sequence of hybrid games. For each game $\mathbf{G}_i$, we denote the probability that it outputs 1 by pr_i, namely,

$$\mathsf{pr}_i := \Pr\left[\mathbf{G}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right].$$

We note that the random oracle G and H can be called throughout the security game in three different cases:

- publicly called by the adversary,
- privately called by the game in the $\textsc{Kdm}$ oracle,
- privately called by the game while computing the function $f^{\mathsf{H},\mathsf{G}}$ in the $\textsc{Kdm}$ oracle.

We will gradually separate these three different cases of the random oracle through the security games by denoting them by $\mathsf{G}, \mathsf{G}^\star, \hat{\mathsf{G}}$ and $\mathsf{H}, \mathsf{H}^\star, \hat{\mathsf{H}}$, respectively.

Game $\mathbf{G}_0$: The game $\mathbf{G}_0$ is the $\mathsf{mKDM\text{-}CCA}$ security game that always encrypts the message $f^{\mathsf{H},\mathsf{G}}(\mathsf{msk})$, except that the answers of the random oracle G, H are conceptually separated into $\mathsf{G}, \mathsf{G}^\star, \hat{\mathsf{G}}$ and $\mathsf{H}, \mathsf{H}^\star, \hat{\mathsf{H}}$. More explicitly, the behavior of the random oracles is given as follows: The random oracles G and $\mathsf{G}^\star$ independently maintain two lists for the hash values $((r,d),\rho)$. They are also sharing the list of values in the sense that every time (r',d') is queried to G (resp. $\mathsf{G}^\star$), if there is already a ρ' such that $((r',d'),\rho') \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$, then the oracle returns ρ'. The random oracle $\hat{\mathsf{G}}$ does not maintain any list, it behaves exactly as G. The random oracles $\mathsf{H}, \mathsf{H}^\star, \hat{\mathsf{H}}$ behave similarly. The details of how oracles

Oracle $\textsc{Kdm}(\mathsf{x}, f)$
$\mathsf{m} := f^{\hat{\mathsf{H}},\hat{\mathsf{G}}}(\mathsf{msk})$
$r \xleftarrow{\$} \{0,1\}^\ell$
$K := \mathsf{H}^\star(r), d := K \oplus \mathsf{m}$
$\rho := \mathsf{G}^\star(r,d)$
$c := \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, r; \rho)$
return $\mathsf{ct} := (c,d)$

Oracle $\mathsf{G}^\star(r,d)$
if $\exists \rho : ((r,d),\rho) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$:
  return ρ
else :
  $\rho \xleftarrow{\$} \{0,1\}^\ell$
  $\mathcal{L}_{\mathsf{G}^\star} := \mathcal{L}_{\mathsf{G}^\star} \cup \{((r,d),\rho)\}$
  return ρ

Oracle $\mathsf{G}(r,d)$
if $\exists \rho : ((r,d),\rho) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$:
  return ρ
else :
  $K \xleftarrow{\$} \{0,1\}^\ell$
  $\mathcal{L}_{\mathsf{G}} := \mathcal{L}_{\mathsf{G}} \cup \{(r,\rho)\}$
  return ρ

Oracle $\hat{\mathsf{G}}(r,d)$
if $\exists \rho : ((r,d),\rho) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$:
  return ρ
else :
  $\rho \xleftarrow{\$} \{0,1\}^\ell$
  $\mathcal{L}_{\mathsf{G}} := \mathcal{L}_{\mathsf{G}} \cup \{((r,d),\rho)\}$
  return ρ

Figure 12: The description of the oracles $\textsc{Kdm}$, G, $\mathsf{G}^\star$ and $\hat{\mathsf{G}}$ in $\mathbf{G}_0$ in the proof of Theorem 3.4.

$\textsc{Kdm}, \mathsf{G}, \mathsf{G}^\star, \hat{\mathsf{G}}$ behave are given in Figure 12. Note that the only difference between $\mathbf{G}_0$ and the game $\mathbf{mKDM\text{-}CCA}_1$ is that we have conceptually differentiated the hash queries to G and H in three different cases. Therefore we have

$$\mathsf{pr}_0 = \Pr\left[\mathbf{mKDM\text{-}CCA}^{\mathcal{A}}_{1,\mathsf{ABE_{FO}}}(\lambda) \Rightarrow 1\right].$$

Game $\mathbf{G}_1$: In game $\mathbf{G}_1$, we modify the behavior of the random oracles $\mathsf{G}^\star$ and $\mathsf{H}^\star$. Every query $\mathsf{G}^\star(r,d)$ and $\mathsf{H}^\star(r)$ is answered with a freshly generated ρ and K. Then, we add the pairs $((r,d),\rho)$ and (r,K)

to the lists $\mathcal{L}_{\mathsf{G}^\star}$ and $\mathcal{L}_{\mathsf{H}}$ respectively. Note that $\mathsf{G}, \hat{\mathsf{G}}$ and $\mathsf{H}, \hat{\mathsf{H}}$ still refer to $\mathcal{L}_{\mathsf{G}^\star}$. Given (r, d), if there exits multiple values ρ or K such that $((r, d), \rho) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$ or $(r, K) \in \mathcal{L}_{\mathsf{H}} \cup \mathcal{L}_{\mathsf{H}^\star}$, then $\mathsf{G}^\star, \hat{\mathsf{G}}$ and $\mathsf{H}^\star, \hat{\mathsf{H}}$ take the first entry as the random oracle's output. The detailed behavior of $\mathsf{G}^\star$ is given in Figure 13, the behavior of $\mathsf{H}^\star$ is similar to $\mathsf{G}^\star$ and we omit it here.

Similar to the proof of Theorem 3.1, we denote by COL the event that in a $\textsc{Kdm}(\mathsf{x}, f)$ query there already exists an entry of the form $((r, d), \cdot) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$ or an entry of the form $(r, \cdot) \in \mathcal{L}_{\mathsf{H}} \cup \mathcal{L}_{\mathsf{H}^\star}$. It is easy to see that $\mathbf{G}_1$ and $\mathbf{G}_2$ only differ when COL happens. We can use the union bound over all Q_C queries in an argumentation similar to the step from $\mathbf{G}_0$ to $\mathbf{G}_1$ in the proof of Theorem 3.1 to show

$$|\mathsf{pr}_0 - \mathsf{pr}_1| \leq \Pr[\mathsf{COL}] \leq \frac{Q_C \cdot Q_G}{2^\ell} + \frac{Q_C \cdot Q_H}{2^\ell}.$$

<u>**Game $\mathbf{G}_2$:**</u> In game $\mathbf{G}_2$, we further modify the behavior of the random oracle G and H, in the sense that G and H no longer refer to $\mathcal{L}_{\mathsf{G}^\star}$ and $\mathcal{L}_{\mathsf{H}^\star}$. We emphasize that $\hat{\mathsf{G}}$ and $\hat{\mathsf{H}}$ still refer to both $\mathcal{L}_{\mathsf{G}}, \mathcal{L}_{\mathsf{G}^\star}$ and $\mathcal{L}_{\mathsf{H}}, \mathcal{L}_{\mathsf{H}^\star}$ respectively in this game. The behavior of $\mathsf{G}^\star$ is presented in Figure 13 and $\mathsf{H}^\star$ behaves similarly, so we omit it here.

We denote by BHQ_i the event that when $\mathcal{A}$ queries (r, d) to G in $\mathbf{G}_i$, there already exists an entry of the form $((r, d), \cdot)$ in $\mathcal{L}_{\mathsf{G}^\star}$ or when $\mathcal{A}$ queries r to H in $\mathbf{G}_i$, there already exists an entry of the form $(r, \cdot)$ in $\mathcal{L}_{\mathsf{H}^\star}$. Note that the only difference between $\mathbf{G}_1$ and $\mathbf{G}_2$ is when BHQ_2 occurs. Therefore, we have

$$|\mathsf{pr}_1 - \mathsf{pr}_2| \leq \Pr[\mathsf{BHQ}_2].$$

<u>**Game $\mathbf{G}_3$:**</u> In game $\mathbf{G}_3$, we modify how decryption queries (i.e. queries of the form $\textsc{Dec}(\mathsf{x}, \mathsf{ct} = (c, d))$) are answered. That is, the game searches for an entry $((r, d), \rho) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$ such that for $\mathsf{m} := \hat{\mathsf{H}}(r) \oplus d$, c is an encryption of m under ABE for attribute x with randomness ρ. It then returns m. If such an entry does not exist, it returns $\perp$. We emphasize that, due to the change in this game, the challenger does not need the secret key to answer the decryption queries. The modified decryption oracle is given in Figure 13.

We define SMTH to be the event that $\mathcal{A}$ makes a decryption query $(\mathsf{x}, \mathsf{ct} = (c, d)) \notin \mathcal{L}_{ch}$ such that there exists m, r, ρ such that

$$c = \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, r; \rho) \wedge ((r, d), \rho) \notin \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}.$$

Note that the only difference between $\mathbf{G}_2$ and $\mathbf{G}_3$ is when SMTH occurs. Further, for each fixed query, if SMTH occurs in this query, then ρ is freshly sampled at random and the probability that the given c coincides with $\mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m}; \rho)$ can be bounded by ε due to the smoothness of ABE. Using a hybrid over all Q_D queries, we obtain

$$|\mathsf{pr}_2 - \mathsf{pr}_3| \leq \Pr[\mathsf{SMTH}] \leq Q_D \cdot \varepsilon.$$

<u>**Game $\mathbf{G}_4$:**</u> We further modify the decryption oracle $\textsc{Dec}$. In this game, instead of searching in the list $\mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$, $\textsc{Dec}$ only searches in $\mathcal{L}_{\mathsf{G}}$, and the decryption oracle uses H instead of $\hat{\mathsf{H}}$ to compute K, see Figure 13.

We also define the event BDQ ("Bad Decryption Query") that $\mathcal{A}$ makes a decryption query $(\mathsf{x}, \mathsf{ct} = (c, d)) \notin \mathcal{L}_{ch}$ which satisfies that there exists $((r, d), \rho) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$ such that

$$c = \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, r; \rho) \wedge \exists K : (r, K) \in \mathcal{L}_{\mathsf{H}^\star}.$$

Note that any difference between $\mathbf{G}_3$ and $\mathbf{G}_4$ implies that BDQ occurs. Therefore we have

$$|\mathsf{pr}_3 - \mathsf{pr}_4| \leq \Pr[\mathsf{BDQ}].$$

<u>**Game $\mathbf{G}_5$:**</u> In $\mathbf{G}_5$, we modify the behavior of the $\textsc{Kdm}$ queries $\textsc{Kdm}(\mathsf{x}, f)$. It returns the encryption of a uniformly random message of length $|f(\cdot)|$.

Similar to the transitions we presented from $\mathbf{mKDM\text{-}CCA}_1$ to $\mathbf{G}_5$, we can follow similar transitions from $\mathbf{mKDM\text{-}CCA}_0$ to $\mathbf{G}_5$. That is, we have

$$\mathsf{Adv}^{\mathsf{mKDM\text{-}CCA}}_{\mathcal{A}, \mathsf{ABE}_{\mathsf{FO}}}(\lambda) \leq 2|\mathsf{pr}_0 - \mathsf{pr}_5|.$$

Thus, it is sufficient to bound $|\mathsf{pr}_0 - \mathsf{pr}_5|$ to finish the proof. To do so, it remains to bound $\Pr[\mathsf{BHQ}_2], \Pr[\mathsf{BDQ}]$ and $|\mathsf{pr}_4 - \mathsf{pr}_5|$, which we do using the following lemmas. Note that the proof of Lemma 3.5 is similar to the proof of Lemma 3.3 in the proof Section 3.1 and the proof of Lemma 3.8 is similar to the proof of Lemma 3.2. Therefore we omit it here.

Oracle $\mathsf{G}^\star(r,d)$ // $\mathbf{G}_1$-$\mathbf{G}_5$	**Oracle** $\textsc{Dec}(\mathsf{x},\mathsf{ct})$ // $\mathbf{G}_3$
$\rho \xleftarrow{\$} \{0,1\}^\ell$	**let** $\mathsf{ct} = (c,d)$
$\mathcal{L}_{\mathsf{G}^\star} := \mathcal{L}_{\mathsf{G}^\star} \cup \{((r,d),\rho)\}$	**for** $((r,d),\rho) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$:
return ρ	$\quad K := \hat{\mathsf{H}}(r), \mathsf{m} := K \oplus d$
	$\quad$ **if** $c = \mathsf{Enc}(\mathsf{mpk},\mathsf{x},r;\rho)$:
Oracle $\mathsf{G}(r,d)$ // $\mathbf{G}_2$-$\mathbf{G}_5$	$\quad\quad$ **return** m
if $\exists \rho : ((r,d),\rho) \in \mathcal{L}_{\mathsf{G}}$:	**return** $\perp$
$\quad$ **return** ρ	
else :	**Oracle** $\textsc{Dec}(\mathsf{x},\mathsf{ct})$ // $\mathbf{G}_4$-$\mathbf{G}_5$
$\quad \rho \xleftarrow{\$} \{0,1\}^\ell$	**let** $\mathsf{ct} = (c,d)$
$\quad \mathcal{L}_{\mathsf{G}} := \mathcal{L}_{\mathsf{G}} \cup \{((r,d),\rho)\}$	**for** $((r,d),\rho) \in \mathcal{L}_{\mathsf{G}}$:
$\quad$ **return** ρ	$\quad K := \mathsf{H}(r), \mathsf{m} := K \oplus d$
	$\quad$ **if** $c = \mathsf{Enc}(\mathsf{mpk},\mathsf{x},r;\rho)$:
	$\quad\quad$ **return** m
	return $\perp$

Figure 13: The changes of the oracles $\textsc{Dec}$, G, $\mathsf{G}^\star$ and $\hat{\mathsf{G}}$ from $\mathbf{G}_1$ to $\mathbf{G}_6$ in the proof of Theorem 3.4. The random oracles $(\mathsf{H}, \mathsf{H}^\star, \hat{\mathsf{H}})$ change similarly.

Lemma 3.5 *There exists an algorithm $\mathcal{B}$ with $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and*

$$\Pr[\mathsf{BHQ}_5] \leq 2 \cdot \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda).$$

Lemma 3.6 $\Pr[\mathsf{BHQ}_2] \leq \Pr[\mathsf{BHQ}_5] + |\mathsf{pr}_4 - \mathsf{pr}_5| + \Pr[\mathsf{SMTH}] + \Pr[\mathsf{BDQ}]$.

Proof. By the triangle inequality, we have:

$$\begin{aligned}\Pr[\mathsf{BHQ}_2] \leq & \Pr[\mathsf{BHQ}_5] + |\Pr[\mathsf{BHQ}_4] - \Pr[\mathsf{BHQ}_5]| \\ & + |\Pr[\mathsf{BHQ}_3] - \Pr[\mathsf{BHQ}_4]| + |\Pr[\mathsf{BHQ}_2] - \Pr[\mathsf{BHQ}_3]|.\end{aligned}$$

Since BHQ_i is a detectable event by the adversary, we can upper bound the probability $|\Pr[\mathsf{BHQ}_{i+1}] - \mathsf{BHQ}_i|$ by $|\mathsf{pr}_{i+1} - \mathsf{pr}_i|$ and the claim follows. $\square$

Lemma 3.7 *There exists an algorithm $\mathcal{B}$ with $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and*

$$\Pr[\mathsf{BDQ}] \leq \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda).$$

Proof. Suppose that BDQ occurs, i.e. the adversary $\mathcal{A}$ queries $(\mathsf{x}, \mathsf{ct} = (c,d)) \notin \mathcal{L}_{ch}$, which satisfies that there exists $((r,d),\rho) \in \mathcal{L}_{\mathsf{G}} \cup \mathcal{L}_{\mathsf{G}^\star}$ such that

$$c = \mathsf{Enc}(\mathsf{mpk},\mathsf{x},r;\rho) \wedge \exists K : (r,K) \in \mathcal{L}_{\mathsf{H}^\star}.$$

We can define the following events as subcases of BDQ:

$$\begin{aligned}\mathsf{BDQ}_1 &:= \exists((r,d),\rho) \in \mathcal{L}_{\mathsf{G}} : (c = \mathsf{Enc}(\mathsf{mpk},\mathsf{x},r;\rho) \wedge \exists K : (r,K) \in \mathcal{L}_{\mathsf{H}^\star})\,, \\ \mathsf{BDQ}_2 &:= \exists((r,d),\rho) \in \mathcal{L}_{\mathsf{G}^\star} : (c = \mathsf{Enc}(\mathsf{mpk},\mathsf{x},r;\rho) \wedge \exists K : (r,K) \in \mathcal{L}_{\mathsf{H}^\star})\,.\end{aligned}$$

It is straightforward that we have $\Pr[\mathsf{BDQ}] = \Pr[\mathsf{BDQ}_1] + \Pr[\mathsf{BDQ}_2]$. For BDQ_1, we can construct an adversary $\mathcal{B}$ against the $\mathsf{OW\text{-}CPA}$ security of ABE. The description of $\mathcal{B}$ is given in Figure 14. Note that $\mathcal{B}$ perfectly simulates $\mathbf{G}_4$ for $\mathcal{A}$ and if BDQ_1 happens, we have $\mathcal{L}_{ans} \cap \mathcal{L}_{pt} \neq \emptyset$, where $\mathcal{L}_{pt}$ is the list hold by the $\mathsf{OW\text{-}CPA}$ security game. Therefore, we have

$$\Pr[\mathsf{BDQ}_1] \leq \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda).$$

For BDQ_2, note that $((r,d),\rho) \in \mathcal{L}_{\mathsf{G}^\star}$ implies that there exists $\mathsf{ct}' = (c',d)$ such that

$$(\mathsf{x},\mathsf{ct}') \in \mathcal{L}_{ch} \wedge c' = \mathsf{Enc}(\mathsf{mpk},\mathsf{x},r;\rho) = c.$$

Therefore, we have $\mathsf{ct} = \mathsf{ct}'$, which contradicts the condition $(\mathsf{x},\mathsf{ct}) \notin \mathcal{L}_{ch}$ and the claim follows. $\square$

$\mathcal{B}^{\textsc{Key},\textsc{Ch}}(\mathsf{mpk})$
$b' \leftarrow \mathcal{A}^{\textsc{Key},\textsc{Kdm},\mathsf{H},\mathsf{G}}(\mathsf{mpk})$
return $\mathcal{L}_{ans}$

Oracle $\mathsf{H}(r)$
if $\exists K : (r, K) \in \mathcal{L}_{\mathsf{H}}$:
  return K
else :
  $\mathcal{L}_{ans} := \mathcal{L}_{ans} \cup \{r\}$
  $K \xleftarrow{\$} \{0,1\}^{\ell}$
  $\mathcal{L}_{\mathsf{H}} := \mathcal{L}_{\mathsf{H}} \cup \{(r, K)\}$
  return K

Oracle $\textsc{Kdm}(\mathsf{x}, f)$
$\mathsf{m}, K \xleftarrow{\$} \{0,1\}^{|f(\cdot)|}, c \leftarrow \textsc{Ch}(\mathsf{x})$
return $\mathsf{ct} := (c, K \oplus \mathsf{m})$

Oracle $\mathsf{G}(r, d)$
if $\exists \rho : ((r, d), \rho) \in \mathcal{L}_{\mathsf{G}}$:
  return ρ
else :
  $\mathcal{L}_{ans} := \mathcal{L}_{ans} \cup \{r\}$
  $\rho \xleftarrow{\$} \{0,1\}^{\ell}$
  $\mathcal{L}_{\mathsf{G}} := \mathcal{L}_{\mathsf{G}} \cup \{((r, d), \rho)\}$
  return ρ

Figure 14: The reduction $\mathcal{B}$ in the proof of Lemma 3.7, which simulates $\mathbf{G}_4$ for adversary $\mathcal{A}$ to win the OW-CPA security game.

Lemma 3.8 $|\mathsf{pr}_4 - \mathsf{pr}_5| = 0$.

In summary, we can now bound $\mathsf{Adv}^{\mathsf{mKDM\text{-}CCA}}_{\mathcal{A},\mathsf{ABE}_{\mathsf{FO}}}(\lambda)$ by

$$\begin{aligned} & 2\left(\Pr[\mathsf{COL}] + \Pr[\mathsf{BHQ}_2] + \Pr[\mathsf{SMTH}] + \Pr[\mathsf{BDQ}] + |\mathsf{pr}_4 - \mathsf{pr}_5|\right) \\ \leq\ & 2\left(\Pr[\mathsf{COL}] + \Pr[\mathsf{BHQ}_5] + 2\left(\Pr[\mathsf{SMTH}] + \Pr[\mathsf{BDQ}] + |\mathsf{pr}_4 - \mathsf{pr}_5|\right)\right) \\ \leq\ & \frac{Q_C \cdot (Q_G + Q_H)}{2^{\ell-1}} + 4Q_D \cdot \varepsilon + 8 \cdot \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{B},\mathsf{ABE}}(\lambda). \end{aligned}$$

□

3.3 Instantiation

In this section, we discuss instantiations of our transformation in Section 3.2. Note that we defined OW-CPA security in the multi-challenge setting, and our transformation is tight. This means that as long as the underlying scheme ABE is tightly OW-CPA secure in the multi-challenge setting, the resulting scheme $\mathsf{ABE}_{\mathsf{FO}}$ is tightly mKDM-CCA secure. The same holds for the transformation in Section 3.1 and mKDM-CPA security. As mentioned in the introduction, we are not aware of any tightly secure ABE.

LATTICE SETTING. In the lattice setting, the modification of the GPV IBE [GPV08] presented in [KYY18] is tightly OW-CPA secure in the multi-challenge setting in the random oracle model. In particular, using our transformation, we obtain the first mKDM-CPA (resp. mKDM-CCA) secure identity-based encryption scheme from lattices and the scheme is tightly secure. Instantiating our transformation with the Tsabary ABE scheme [Tsa19], we get the first mKDM-CPA (resp. mKDM-CCA) secure attribute-based encryption scheme from lattices.

PAIRING SETTING. We can instantiate our construction in the pairing setting using the Boneh-Franklin (BF) IBE scheme [BF01]. This scheme is not tightly IND-CPA secure in the multi-challenge setting. However, it is folklore knowledge that applying a Katz-Wang technique [KW03] yields a tightly secure scheme BF-KW and only increases the size of the ciphertext by one group element. For completeness, we describe both schemes in Figure 15.

4 Generic Construction in the Standard Model

Here, we generically construct an mKDM-CCA secure attribute-based encryption scheme in the standard model, starting from an underlying attribute-based encryption scheme with IND-CPA security. An overview of our construction is given in Figure 16. Before we do so, we define some properties of the underlying attribute-based encryption scheme that will be useful for our construction. In Sections 4.3 and 4.4, we explain how to achieve these properties.

Alg $\mathsf{Setup}(1^\lambda)$

$(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_\mathsf{T}, \mathsf{e}, p) \leftarrow \mathsf{GGen}(\lambda)$
$s \xleftarrow{\$} \mathbb{Z}_p$
$\mathsf{mpk} := \hat{g}^s \in \hat{\mathbb{G}},\ \mathsf{msk} := s$
return $(\mathsf{mpk}, \mathsf{msk})$

Alg $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mathsf{m})$

$r \xleftarrow{\$} \{0,1\}^\ell,\ K := \mathsf{H}(r)$
$w := K \oplus \mathsf{m}$
$t := \mathsf{G}(r, w)$
$g_\mathsf{id} := e(\mathsf{H}(\mathsf{id}), \mathsf{mpk})$
$\mathsf{ct}_\mathsf{id} := (\hat{g}^t, r \oplus \mathsf{H}_\mathsf{T}(g_\mathsf{id}^t), w)$
$g_{\mathsf{id},0} := e(\mathsf{H}(\mathsf{id}||0), \mathsf{mpk})$
$g_{\mathsf{id},1} := e(\mathsf{H}(\mathsf{id}||1), \mathsf{mpk})$
$\mathsf{ct}_\mathsf{id} := (\hat{g}^t, r \oplus \mathsf{H}_\mathsf{T}(g_{\mathsf{id},0}^t), r \oplus \mathsf{H}_\mathsf{T}(g_{\mathsf{id},1}^t), w)$
return ct_id

Alg $\mathsf{KeyExt}(\mathsf{msk}, \mathsf{id})$

$\mathsf{sk}_\mathsf{id} := \mathsf{H}(\mathsf{id})^s$
if $b_\mathsf{id} = \perp$: $b_\mathsf{id} \xleftarrow{\$} \{0,1\}$
$\mathsf{sk}_\mathsf{id} := (\mathsf{H}(\mathsf{id}||b_\mathsf{id})^s, b_\mathsf{id})$
return sk_id

Alg $\mathsf{Dec}(\mathsf{sk}_\mathsf{id}, \mathsf{ct}_\mathsf{id})$

let $\mathsf{sk}_\mathsf{id} = (u, b_\mathsf{id})$
let $\mathsf{ct}_\mathsf{id} = (\hat{u}, v, w)$
let $\mathsf{ct}_\mathsf{id} = (\hat{u}, v_0, v_1, w)$
// For BF, b_id is an empty string
$r := v_{b_\mathsf{id}} \oplus \mathsf{H}_\mathsf{T}(\mathsf{e}(u, \hat{u}))$
$t := \mathsf{G}(r, w)$
$\mathsf{m} := \mathsf{H}(r) \oplus w$
$h_{\mathsf{id},b_\mathsf{id}} \leftarrow \mathsf{H}(\mathsf{id}||b_\mathsf{id})$
$g_{\mathsf{id},b_\mathsf{id}} := e(h_{\mathsf{id},b_\mathsf{id}}, \mathsf{mpk})$
if $(\hat{u}, v_{b_\mathsf{id}}) = (\hat{g}^t, r \oplus \mathsf{H}_\mathsf{T}(g_{\mathsf{id},b_\mathsf{id}}^t))$:
  return m
return $\perp$

Figure 15: Our instantiations of IBE_FO in Section 3.2 using BF and its tight variant BF-KW. Codes in grey are only executed in the instantiation from BF-KW. $\mathsf{H} : \{0,1\}^* \to \mathbb{G}, \mathsf{G} : \{0,1\}^* \to \mathbb{Z}_p, \mathsf{H}_\mathsf{T} : \mathbb{G}_\mathsf{T} \to \{0,1\}^\ell$ are hash functions modeled as random oracles. GGen generates a pairing group with $\mathsf{e} : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_\mathsf{T}$, where $\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_\mathsf{T}$ are three groups of prime order p with generators $g, \hat{g}, g_\mathsf{T}$, respectively.

4.1 Definitions of Key Verifiability

For the following definitions, we consider a perfectly complete attribute-based encryption scheme $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ for predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$.

Definition 4.1 (Verifiable User Secret Keys). We say that ABE has verifiable user secret keys, if there exists a deterministic polynomial time algorithm VerK satisfying the following properties:

- $\mathsf{VerK}(\mathsf{mpk}, \mathsf{x}, \mathsf{sk}_\mathsf{y})$ takes as input a master public key mpk, an attribute $\mathsf{x} \in \mathcal{X}$, and a user secret key sk_y and outputs a bit $b \in \{0,1\}$.
- For all $(\mathsf{mpk}, \mathsf{msk}) \in \mathsf{Setup}(1^\lambda), \mathsf{x} \in \mathcal{X}$ and all sk_y with $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$, we have:

$$\mathsf{sk}_\mathsf{y} \in \mathsf{KeyExt}(\mathsf{msk}, \mathsf{y}) \Longrightarrow \mathsf{VerK}(\mathsf{mpk}, \mathsf{x}, \mathsf{sk}_\mathsf{y}) = 1 \text{ and}$$
$$\mathsf{VerK}(\mathsf{mpk}, \mathsf{x}, \mathsf{sk}_\mathsf{y}) = 1 \Longrightarrow \forall \mathsf{m} \in \mathcal{M} : \mathsf{Dec}(\mathsf{sk}_\mathsf{y}, \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})) = \mathsf{m}.$$

Definition 4.2 (Verifiable Master Secret Keys). We say that ABE has verifiable master secret keys, if there exists a deterministic polynomial time algorithm VerMK satisfying the following properties:

- $\mathsf{VerMK}(\mathsf{mpk}, \mathsf{msk})$ takes as input a master public key mpk and a master secret key msk and outputs a bit $b \in \{0,1\}$.
- For all $(\mathsf{mpk}, \mathsf{msk}) \in \mathsf{Setup}(1^\lambda)$ and all msk' we have:

$$\mathsf{VerMK}(\mathsf{mpk}, \mathsf{msk}') = 1 \Longleftrightarrow (\mathsf{mpk}, \mathsf{msk}') \in \mathsf{Setup}(1^\lambda).$$

Definition 4.3 (Uniquely Verifiable Master Secret Keys). We say that ABE has uniquely verifiable master secret keys if ABE has verifiable master secret keys with algorithm VerMK and for every $(\mathsf{mpk}, \mathsf{msk}) \in \mathsf{Setup}(1^\lambda)$ there does not exist a $\mathsf{msk}' \neq \mathsf{msk}$ such that $\mathsf{VerMK}(\mathsf{mpk}, \mathsf{msk}') = 1$.

We highlight that while we defined functional verifiability of user secret keys, we defined syntactical verifiability of master secret keys. That is, for user secret keys it should be verifiable if they can decrypt correctly, while for master secret keys it should be verifiable if they are really honestly generated. Note that this may be different conditions. We also want to remark that the properties only have to hold for honestly generated master public keys.

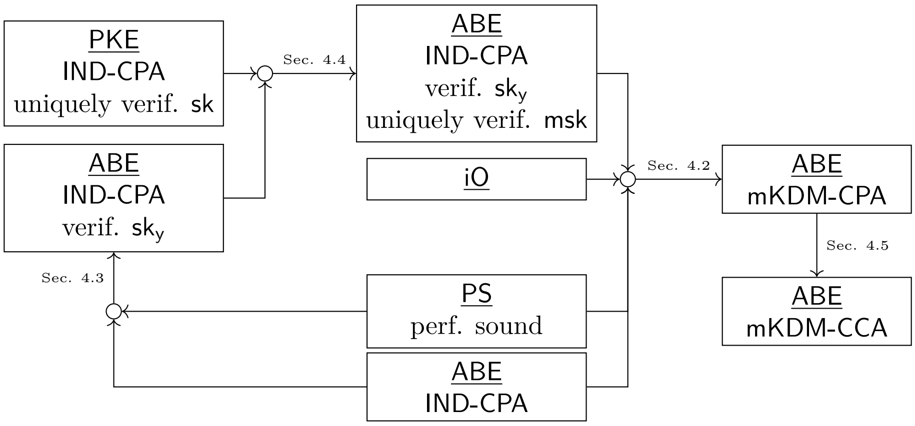

Figure 16: Overview of our construction of $\mathsf{mKDM\text{-}CPA}/\mathsf{mKDM\text{-}CCA}$ secure attribute-based encryption in the standard model. We transform an $\mathsf{IND\text{-}CPA}$ secure attribute-based encryption scheme into an $\mathsf{mKDM\text{-}CPA}$ secure one, using an indistinguishability obfuscator iO and a NIZK PS.

4.2 Main Construction

We first define when two predicates are compatible.

Definition 4.4 (Compatible Predicates). We say that two predicates $\mathcal{P}' \colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and $\mathcal{P}' \colon \mathcal{Y} \times \mathcal{X} \to \{0,1\}$ are compatible, if for all attributes $\mathsf{x} \in \mathcal{X}, \mathsf{y} \in \mathcal{Y}$, it holds that $\mathcal{P}'(\mathsf{x},\mathsf{y}) = \mathcal{P}''(\mathsf{y},\mathsf{x})$.

For our construction, let $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$ and $\mathsf{ABE}'' = (\mathsf{Setup}'', \mathsf{KeyExt}'', \mathsf{Enc}'', \mathsf{Dec}'')$ be two $\mathsf{IND\text{-}CPA}$ secure attribute-based encryption schemes for compatible predicates $\mathcal{P}' \colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and $\mathcal{P}'' \colon \mathcal{Y} \times \mathcal{X} \to \{0,1\}$, respectively. Further, we assume that ABE' and ABE'' are perfectly complete, and that ABE' has uniquely verifiable master secret keys and verifiable user secret keys with algorithms VerMK and VerK, respectively. We assume that the encryption randomness of ABE'' has length $z = z(\lambda)$ and that ABE'' can encrypt the master secret key of ABE'. In addition to that, we need a perfectly sound and perfectly complete non-interactive zero-knowledge proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$ for the relation

$$\mathcal{R} := \left\{ ((\mathsf{ct}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}''), (\mathsf{msk}', \rho)) \;\middle|\; \begin{array}{rrl} & \mathsf{Enc}''(\mathsf{mpk}'', \mathsf{y}, \mathsf{msk}'; \rho) & = \mathsf{ct} \\ \wedge & \mathsf{VerMK}(\mathsf{mpk}', \mathsf{msk}') & = 1 \end{array} \right\}.$$

That is, PS can be used to prove that a given ciphertext is an encryption (with respect to ABE'') of the valid master secret key under a given attribute. Finally, we assume an indistinguishability obfuscator iO[2]. Let $L = L(\lambda)$ be an upper bound on the size of all circuits presented in this section. We denote the execution of iO with a padding of a circuit C to size L as input by iO_p. Equipped with these primitives, we construct a new attribute-based encryption scheme $\mathsf{ABE}[\mathsf{ABE}', \mathsf{ABE}'', \mathsf{iO}, \mathsf{PS}]$ for predicate $\mathcal{P} = \mathcal{P}'$ with message space $\mathcal{M}$. The scheme is given in Figure 17. At a high level, the idea is to construct a circuit that outputs the message if the input is a valid user secret key and use this circuit as the ciphertext. Also, the master secret key is embedded into user secret keys in a hidden way.

Remark 4.5 (Message Space). In our construction, the message space $\mathcal{M}$ can be arbitrary. However, $\mathsf{mKDM\text{-}CPA}$ security will hold only with respect to efficiently computable functions with range $\mathcal{M}$ that have descriptions which can be encrypted by ABE'. Thus, there is a relation between security and the message spaces of ABE' and $\mathsf{ABE}[\mathsf{ABE}', \mathsf{ABE}'', \mathsf{iO}, \mathsf{PS}]$.

Lemma 4.6 (Completeness). *Let ABE' be a perfectly complete attribute-based encryption scheme for predicate $\mathcal{P}'$ with verifiable master secret keys and verifiable user secret keys. Let ABE'' be a perfectly complete attribute-based encryption scheme. Assume that $\mathcal{P}'$ and $\mathcal{P}''$ are compatible. Let PS be a perfectly complete $(\varepsilon_{\mathsf{so}}, \varepsilon_{\mathsf{zk}})$-NIZK proof system for the relation $\mathcal{R}$. Let iO be an indistinguishability obfuscator. Then $\mathsf{ABE}[\mathsf{ABE}', \mathsf{ABE}'', \mathsf{iO}, \mathsf{PS}]$ is perfectly complete for predicate $\mathcal{P} := \mathcal{P}'$.*

[2] We do not explicitly define the circuit class for which iO works. It is implicitly given in the construction and proof, see circuits $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{m}}$ in Figure 17 and $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_{\mathsf{x}}}$ in Figure 18.

Alg $\mathsf{Setup}(1^\lambda)$

01 $(\mathsf{mpk}', \mathsf{msk}') \leftarrow \mathsf{Setup}'(1^\lambda)$
02 $(\mathsf{mpk}'', \mathsf{msk}'') \leftarrow \mathsf{Setup}''(1^\lambda)$
03 $\mathsf{crs} \leftarrow \mathsf{PGen}(1^\lambda)$
04 $\mathsf{mpk} := (\mathsf{mpk}', \mathsf{mpk}'', \mathsf{crs}), \mathsf{msk} := \mathsf{msk}'$
05 **return** $(\mathsf{mpk}, \mathsf{msk})$

Alg $\mathsf{KeyExt}(\mathsf{msk}, \mathsf{y})$

06 $\mathsf{sk}'_\mathsf{y} \leftarrow \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y})$
07 $\rho \xleftarrow{\$} \{0,1\}^z$
08 $\mathsf{ct}_\mathsf{msk} := \mathsf{Enc}''(\mathsf{mpk}'', \mathsf{y}, \mathsf{msk}'; \rho)$
09 $\mathsf{stmt} := (\mathsf{ct}_\mathsf{msk}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}'')$
10 $\mathsf{witn} := (\mathsf{msk}', \rho)$
11 $\pi \leftarrow \mathsf{PProve}(\mathsf{crs}, \mathsf{stmt}, \mathsf{witn})$
12 **return** $\mathsf{sk}_\mathsf{y} := (\mathsf{sk}'_\mathsf{y}, \mathsf{ct}_\mathsf{msk}, \pi)$

Alg $\mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})$

13 $\hat{\mathsf{C}} := \mathsf{iO}_p(\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{m}})$
14 **return** $\mathsf{ct} := \hat{\mathsf{C}}$

Alg $\mathsf{Dec}(\mathsf{sk}_\mathsf{y}, \mathsf{ct} = \hat{\mathsf{C}})$

15 **return** $\hat{\mathsf{C}}(\mathsf{sk}_\mathsf{y})$

Circuit $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{m}}(\mathsf{sk}_\mathsf{y})$

16 **let** $\mathsf{sk}_\mathsf{y} = (\mathsf{sk}'_\mathsf{y}, \mathsf{ct}_\mathsf{msk}, \pi)$
17 **if** $\mathsf{VerK}(\mathsf{mpk}', \mathsf{x}, \mathsf{sk}'_\mathsf{y}) = 0$:
18   **return** $\perp$
19 $\mathsf{stmt} := (\mathsf{ct}_\mathsf{msk}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}'')$
20 **if** $\mathsf{PVer}(\mathsf{crs}, \mathsf{stmt}, \pi) = 0$:
21   **return** $\perp$
22 **if** $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 0$: **return** $\perp$
23 **return** m

Figure 17: The attribute-based encryption scheme $\mathsf{ABE}[\mathsf{ABE}', \mathsf{ABE}'', \mathsf{iO}, \mathsf{PS}] = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ for given attribute-based encryption schemes $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$ and $\mathsf{ABE}'' = (\mathsf{Setup}'', \mathsf{KeyExt}'', \mathsf{Enc}'', \mathsf{Dec}'')$, an indistinguishability obfuscator iO and a proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$.

Proof. Let $(\mathsf{mpk}, \mathsf{msk}) \in \mathsf{Setup}(1^\lambda), \mathsf{x} \in \mathcal{X}, \mathsf{y} \in \mathcal{Y}$ such that $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$, and $\mathsf{sk}_\mathsf{y} \in \mathsf{KeyExt}(\mathsf{msk}, \mathsf{y}), \mathsf{m} \in \mathcal{M}$. Recall that $\mathsf{mpk} = (\mathsf{mpk}', \mathsf{mpk}'', \mathsf{crs})$ and $\mathsf{sk}_\mathsf{y} = (\mathsf{sk}'_\mathsf{y}, \mathsf{ct}_\mathsf{msk}, \pi)$ for $\mathsf{sk}'_\mathsf{y}, \mathsf{ct}_\mathsf{msk}, \mathsf{mpk}', \mathsf{mpk}'', \mathsf{crs}, \pi$ as in Figure 17. Consider a ciphertext $\mathsf{ct} = \mathsf{iO}_p(\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{m}})$. Also, recall that decryption of $\mathsf{ABE}[\mathsf{ABE}', \mathsf{ABE}'', \mathsf{iO}, \mathsf{PS}]$ works by executing the circuit $\mathsf{iO}_p(\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{m}})$ on input sk_y. We have to show that this execution returns m. As iO and padding preserves functionality of circuits, it is sufficient to show that $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{m}}$ as defined in Figure 17 returns m. To see that, note that $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{m}}$ returns m unless the condition in Line 17, the condition in Line 20, or the condition in Line 22 is satisfied. Here, the first condition is never satisfied due to the definition of verifiable user secret keys, completeness of ABE', $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$, and $\mathsf{sk}'_\mathsf{y} \in \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y})$. The second condition is never satisfied due to the definition of verifiable master secret keys, perfect completeness of PS and $(\mathsf{mpk}', \mathsf{msk}') \in \mathsf{Setup}'(1^\lambda)$. The third condition is never satisfied by assumption. The claim follows. $\square$

Theorem 4.7 *Let* ABE' *be a perfectly complete* $\mathsf{IND\text{-}CPA}$ *secure attribute-based encryption scheme for predicate* $\mathcal{P}'$ *with master secret key space* $\mathcal{K}_m$*, uniquely verifiable master secret keys and verifiable user secret keys. Let* ABE'' *be a perfectly complete* $\mathsf{IND\text{-}CPA}$ *secure attribute-based encryption scheme for predicate* $\mathcal{P}''$*. Assume that* $\mathcal{P}'$ *and* $\mathcal{P}''$ *are compatible. Let* PS *be a perfectly complete and perfectly sound* ε_zk*-NIZK proof system for the relation* $\mathcal{R}$*. Let* iO *be an indistinguishability obfuscator. Finally, let* $\mathcal{F}$ *be the class of all efficiently computable functions with domain* $\mathcal{K}_m$ *and descriptions that can be encrypted by* ABE'*.*

Then $\mathsf{ABE} := \mathsf{ABE}[\mathsf{ABE}', \mathsf{ABE}'', \mathsf{iO}, \mathsf{PS}]$ *is* $\mathcal{F}$*-*$\mathsf{mKDM\text{-}CPA}$ *secure. In particular, for every PPT algorithm* $\mathcal{A}$ *making* Q_C, Q_K *queries to the oracles* $\textsc{Kdm}, \textsc{Key}$*, respectively, there are PPT algorithms* $\mathcal{B}_1^*, \mathcal{B}_2^*, \mathcal{B}_3^*, \mathcal{B}'$ *with* $\mathbf{T}(\mathcal{B}_i^*) \approx \mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}')$ *for* $i \in \{1,2,3\}$ *and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{mKDM\text{-}CPA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) \leq 2Q_C \cdot \mathsf{Adv}^{\mathsf{iodist}}_{\mathcal{B}_1^*,\mathsf{iO}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}_2^*,\mathsf{PS}}(\lambda) + 2Q_K \cdot \varepsilon_\mathsf{zk} \\ + 2 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_3^*,\mathsf{ABE}''}(\lambda) + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}',\mathsf{ABE}'}(\lambda).\end{aligned}$$

Proof. Let $\mathcal{A}$ be a PPT algorithm and $\mathsf{ABE} := \mathsf{ABE}[\mathsf{ABE}', \mathsf{ABE}'', \mathsf{iO}, \mathsf{PS}]$. We have to show that

$$\left|\Pr\left[\mathbf{mKDM\text{-}CPA}^{\mathcal{A}}_{0,\mathsf{ABE}}(\lambda) \Rightarrow 1\right] - \Pr\left[\mathbf{mKDM\text{-}CPA}^{\mathcal{A}}_{1,\mathsf{ABE}}(\lambda) \Rightarrow 1\right]\right|$$

is negligible. To do so, we interpolate between both games via intermediate games $\mathbf{G}_i$ for $0 \leq i \leq 9$. For each game $\mathbf{G}_i$, we denote the probability that it outputs 1 by pr_i, namely,

$$\mathsf{pr}_i := \Pr\left[\mathbf{G}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right].$$

First, let us introduce the structure of the proof on a high level. Starting with the game $\mathbf{mKDM\text{-}CPA}_{1,\mathsf{ABE}}$,

Game $\mathbf{G}_0$-$\mathbf{G}_9$

$(\mathsf{mpk}', \mathsf{msk}') \leftarrow \mathsf{Setup}'(1^\lambda)$
$(\mathsf{mpk}'', \mathsf{msk}'') \leftarrow \mathsf{Setup}''(1^\lambda)$
$\mathsf{crs} \leftarrow \mathsf{PGen}(1^\lambda)$ // $\mathbf{G}_0, \mathbf{G}_1, \mathbf{G}_8, \mathbf{G}_9$
$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda)$ // $\mathbf{G}_2$-$\mathbf{G}_7$
$\mathsf{mpk} := (\mathsf{mpk}', \mathsf{mpk}'', \mathsf{crs})$
$b' \leftarrow \mathcal{A}^{\textsc{Key},\textsc{Kdm}}(\mathsf{mpk})$
return b'

Oracle $\textsc{Kdm}(\mathsf{x}, f \in \mathcal{F})$

if $\mathsf{hit}_{\mathcal{P}}(\{\mathsf{x}\}, \mathcal{L}_{sk})$: **return** $\perp$
$\mathcal{L}_{ch} := \mathcal{L}_{ch} \cup \{\mathsf{x}\}$
$\mathsf{m} := f(\mathsf{msk}')$ // $\mathbf{G}_0$-$\mathbf{G}_4$
$\mathsf{m} := 0^{|f(\mathsf{msk}')|}$ // $\mathbf{G}_5$-$\mathbf{G}_9$
$\hat{\mathsf{C}} := \mathsf{iO}_p(\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{m}})$ // $\mathbf{G}_0, \mathbf{G}_9$
$\mathsf{ct}_f := \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, f)$ // $\mathbf{G}_1$-$\mathbf{G}_4$
$\mathsf{ct}_f := \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, Z)$ // $\mathbf{G}_5$-$\mathbf{G}_8$
$\mathsf{sk}''_{\mathsf{x}} \leftarrow \mathsf{KeyExt}''(\mathsf{msk}'', \mathsf{x})$ // $\mathbf{G}_1$-$\mathbf{G}_8$
$\hat{\mathsf{C}} := \mathsf{iO}_p(\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_{\mathsf{x}}})$ // $\mathbf{G}_1$-$\mathbf{G}_8$
return $\mathsf{ct} := \hat{\mathsf{C}}$

Oracle $\textsc{Key}(\mathsf{y})$

if $\mathsf{hit}_{\mathcal{P}}(\mathcal{L}_{ch}, \{\mathsf{y}\})$: **return** $\perp$
$\mathcal{L}_{sk} := \mathcal{L}_{sk} \cup \{\mathsf{y}\}$
$\mathsf{sk}'_{\mathsf{y}} \leftarrow \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y})$
$\rho \xleftarrow{\$} \{0,1\}^z$
$\mathsf{ct}_{\mathsf{msk}} := \mathsf{Enc}''(\mathsf{mpk}'', \mathsf{y}, \mathsf{msk}'; \rho)$ // $\mathbf{G}_0$-$\mathbf{G}_3, \mathbf{G}_6$-$\mathbf{G}_9$
$\mathsf{ct}_{\mathsf{msk}} := \mathsf{Enc}''(\mathsf{mpk}'', \mathsf{y}, 0^{|\mathsf{msk}'|}; \rho)$ // $\mathbf{G}_4$-$\mathbf{G}_5$
$\mathsf{stmt} := (\mathsf{ct}_{\mathsf{msk}}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}'')$
$\mathsf{witn} := (\mathsf{msk}', \rho)$ // $\mathbf{G}_0$-$\mathbf{G}_2, \mathbf{G}_7$-$\mathbf{G}_9$
$\pi \leftarrow \mathsf{PProve}(\mathsf{crs}, \mathsf{stmt}, \mathsf{witn})$ // $\mathbf{G}_0$-$\mathbf{G}_2, \mathbf{G}_7$-$\mathbf{G}_9$
$\pi \leftarrow \mathsf{PSim}(\mathsf{crs}, \mathsf{td}, \mathsf{stmt})$ // $\mathbf{G}_3$-$\mathbf{G}_6$
return $\mathsf{sk}_{\mathsf{y}} := (\mathsf{sk}'_{\mathsf{y}}, \mathsf{ct}_{\mathsf{msk}}, \pi)$

Circuit $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_{\mathsf{x}}}(\mathsf{sk}_{\mathsf{y}})$

let $\mathsf{sk}_{\mathsf{y}} = (\mathsf{sk}'_{\mathsf{y}}, \mathsf{ct}_{\mathsf{msk}}, \pi)$
if $\mathsf{VerK}(\mathsf{mpk}', \mathsf{x}, \mathsf{sk}'_{\mathsf{y}}) = 0$:
  return $\perp$
$\mathsf{stmt} := (\mathsf{ct}_{\mathsf{msk}}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}'')$
if $\mathsf{PVer}(\mathsf{crs}, \mathsf{stmt}, \pi) = 0$:
  return $\perp$
if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 0$: **return** 0
$\widehat{\mathsf{msk}} := \mathsf{Dec}''(\mathsf{sk}''_{\mathsf{x}}, \mathsf{ct}_{\mathsf{msk}})$
$\hat{f} := \mathsf{Dec}'(\mathsf{sk}'_{\mathsf{y}}, \mathsf{ct}_f)$
if $\mathsf{VerMK}(\mathsf{mpk}', \widehat{\mathsf{msk}}) = 0$:
  return $\perp$
return $\hat{f}(\widehat{\mathsf{msk}})$

Figure 18: The games $\mathbf{G}_0$-$\mathbf{G}_9$ in the proof of Theorem 4.7. Lines with highlighted comments are only executed in the corresponding games. Here, Z is the all-zero function.

which encrypts $\mathsf{m} = f(\mathsf{msk})$ in every query $\textsc{Kdm}(\mathsf{x}, f)$, we first use perfect soundness of PS and the security of iO to change the ciphertext to a circuit that can be constructed without knowing $\mathsf{msk} = \mathsf{msk}'$. Instead, we use msk'' to enable the circuit to extract msk from its input. Next, we use zero-knowledge and the security of ABE'' with respect to mpk'' to remove $\mathsf{msk} = \mathsf{msk}'$ from all key queries $\textsc{Key}(\mathsf{y})$. Finally, we apply the security of ABE' with respect to mpk' and undo all our previous changes, resulting in the game $\mathbf{mKDM\text{-}CPA}_{0,\mathsf{ABE}}$. Let us now go into the details of the proof.

Game $\mathbf{G}_0$: We set $\mathbf{G}_0 = \mathbf{mKDM\text{-}CPA}_{1,\mathsf{ABE}}$. That is, the adversary has access to oracles $\textsc{Kdm}$ and $\textsc{Key}$ that return ciphertexts and user secret keys, respectively. Recall that in this game, every query $\textsc{Kdm}(\mathsf{x}, f)$ returns a ciphertext ct encrypting $f(\mathsf{msk})$ with respect to x. This ciphertext is the obfuscation of a circuit $\mathsf{C}_{\mathsf{mpk},\mathsf{x},f(\mathsf{msk}')}$ computing the function

$$(\mathsf{sk}'_{\mathsf{y}}, \mathsf{ct}_{\mathsf{msk}}, \pi) \mapsto \begin{cases} f(\mathsf{msk}'), & \text{if} \quad \begin{array}{c} \mathsf{VerK}(\mathsf{mpk}', \mathsf{x}, \mathsf{sk}'_{\mathsf{y}}) = 1 \wedge \mathcal{P}(\mathsf{x}, \mathsf{y}) = 1 \\ \wedge \quad \mathsf{PVer}(\mathsf{crs}, (\mathsf{ct}_{\mathsf{msk}}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}''), \pi) = 1 \end{array} \\ \perp, & \text{otherwise} \end{cases}.$$

Game $\mathbf{G}_1$: In $\mathbf{G}_1$ we change the ciphertexts constructed in queries $\textsc{Kdm}(\mathsf{x}, f)$. The game computes an ABE' ciphertext $\mathsf{ct}_f := \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, f)$ and a user secret key $\mathsf{sk}''_{\mathsf{x}} \leftarrow \mathsf{KeyExt}''(\mathsf{msk}'', \mathsf{x})$ and returns

$\mathsf{ct} = \mathsf{iO}_p(\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}})$, where $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}}$ is a circuit that takes $(\mathsf{sk}'_\mathsf{y}, \mathsf{ct}_{\mathsf{msk}}, \pi)$ as input, decrypts $\widehat{\mathsf{msk}} := \mathsf{Dec}''(\mathsf{sk}''_\mathsf{x}, \mathsf{ct}_{\mathsf{msk}})$ and $\hat{f} := \mathsf{Dec}'(\mathsf{sk}'_\mathsf{y}, \mathsf{ct}_f)$, and the returns

$$\begin{cases} \hat{f}(\widehat{\mathsf{msk}}), & \text{if} \quad \begin{array}{c} \mathsf{VerK}(\mathsf{mpk}', \mathsf{x}, \mathsf{sk}'_\mathsf{y}) = 1 \wedge \mathsf{VerMK}(\mathsf{mpk}', \widehat{\mathsf{msk}}) = 1 \wedge \mathcal{P}(\mathsf{x}, \mathsf{y}) = 1 \\ \wedge \quad \mathsf{PVer}(\mathsf{crs}, (\mathsf{ct}_{\mathsf{msk}}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}''), \pi) = 1 \end{array} \\ \bot, & \text{otherwise} \end{cases}.$$

In the following, we argue that the circuits $\mathsf{C}_{\mathsf{mpk},\mathsf{x},f(\mathsf{msk}')}$ and $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}}$ are functionally equivalent. First, assume that both circuits do not output $\bot$. In this case, $\mathsf{C}_{\mathsf{mpk},\mathsf{x},f(\mathsf{msk}')}$ outputs the hardcoded $f(\mathsf{msk}')$, and $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}}$ outputs $\hat{f}(\widehat{\mathsf{msk}})$. As $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}}$ did not output $\bot$, it must hold that (1) $\mathsf{VerMK}(\mathsf{mpk}', \widehat{\mathsf{msk}}) = 1$ and (2) $\mathsf{VerK}(\mathsf{mpk}', \mathsf{x}, \mathsf{sk}'_\mathsf{y}) = 1$ and $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$. As ABE' has uniquely verifiable master secret keys, (1) implies that $\widehat{\mathsf{msk}} = \mathsf{msk}'$. By definition of verifiable user secret keys, (2) implies that

$$\hat{f} = \mathsf{Dec}'(\mathsf{sk}'_\mathsf{y}, \mathsf{ct}_f) = \mathsf{Dec}'(\mathsf{sk}'_\mathsf{y}, \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, f)) = f,$$

and therefore $\hat{f}(\widehat{\mathsf{msk}}) = f(\mathsf{msk}')$.

Second, we claim that the set of inputs for which circuit $\mathsf{C}_{\mathsf{mpk},\mathsf{x},f(\mathsf{msk}')}$ outputs $\bot$ and the set of inputs for which $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}}$ outputs $\bot$ are identical. Note that both circuits output $\bot$ if (1) $\mathsf{VerK}(\mathsf{mpk}', \mathsf{x}, \mathsf{sk}'_\mathsf{y}) = 0$, or (2) $\mathsf{PVer}(\mathsf{crs}, (\mathsf{ct}_{\mathsf{msk}}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}''), \pi) = 0$, or (3) $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 0$. Additionally, circuit $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}}$ outputs $\bot$, if (4) $\mathsf{VerMK}(\mathsf{mpk}', \widehat{\mathsf{msk}}) = 0$. We argue that if conditions (1),(2) and (3) do not hold, then (4) does not hold either. To see this, observe that the perfect soundness of PS and the definition of $\mathcal{R}$ imply that $\mathsf{ct}_{\mathsf{msk}}$ is an encryption with respect to $\mathsf{msk}'', \mathsf{y}$ of some $\overline{\mathsf{msk}}$ such that $\mathsf{VerMK}(\mathsf{mpk}', \overline{\mathsf{msk}}) = 1$. As condition (3) does not hold, we have $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$ and therefore, by completeness of ABE'',

$$\overline{\mathsf{msk}} = \mathsf{Dec}(\mathsf{sk}''_\mathsf{x}, \mathsf{ct}_{\mathsf{msk}}) = \widehat{\mathsf{msk}}.$$

Thus, it holds that $\mathsf{VerMK}(\mathsf{mpk}', \widehat{\mathsf{msk}}) = 1$, showing that condidion (4) does not hold. Functional equivalence follows. Now, the security of iO implies that for one query, the change is unnoticed by $\mathcal{A}$. Using a hybrid argument over all such queries, we obtain a reduction $\mathcal{B}_1$ with

$$|\mathsf{pr}_0 - \mathsf{pr}_1| \leq Q_C \cdot \mathsf{Adv}^{\mathsf{iodist}}_{\mathcal{B}_1,\mathsf{iO}}(\lambda).$$

Game $\mathbf{G}_2$: In $\mathbf{G}_2$, we change the way crs is generated. That is, we generate it in combination with a trapdoor td via $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda)$. A straight-forward reduction $\mathcal{B}_2$ shows that

$$|\mathsf{pr}_1 - \mathsf{pr}_2| \leq \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}_2,\mathsf{PS}}(\lambda).$$

Game $\mathbf{G}_3$: In $\mathbf{G}_3$, we change the way the proofs π in queries of the form $\textsc{Key}(\mathsf{y})$ are generated. Recall that in $\mathbf{G}_2$, the proofs are generated via $\pi \leftarrow \mathsf{PProve}(\mathsf{crs}, \mathsf{stmt}, \mathsf{witn})$, where $\mathsf{stmt} = (\mathsf{ct}_{\mathsf{msk}}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}'')$ and $\mathsf{witn} = (\mathsf{msk}', \rho)$. In $\mathbf{G}_3$, proofs are generated using the trapdoor td and the simulation algorithm via $\pi \leftarrow \mathsf{PSim}(\mathsf{crs}, \mathsf{td}, \mathsf{stmt})$. The games are statistically close by the zero-knowledge property of PS, i.e.

$$|\mathsf{pr}_2 - \mathsf{pr}_3| \leq Q_K \cdot \varepsilon_{\mathsf{zk}}.$$

Game $\mathbf{G}_4$: In $\mathbf{G}_4$, we change the way the ciphertexts $\mathsf{ct}_{\mathsf{msk}}$ in queries of the form $\textsc{Key}(\mathsf{y})$ are generated. Recall that before, we generated these ciphertexts as $\mathsf{ct}_{\mathsf{msk}} := \mathsf{Enc}''(\mathsf{mpk}'', \mathsf{y}, \mathsf{msk}'; \rho)$. In $\mathbf{G}_4$, we generate them as $\mathsf{ct}_{\mathsf{msk}} := \mathsf{Enc}''(\mathsf{mpk}'', \mathsf{y}, 0^{|\mathsf{msk}'|}; \rho)$. We claim that we can show indistinguishability of both games using the $\mathsf{IND\text{-}CPA}$ security of ABE'' with regards to the public key mpk''. To see this, note that we do not need ρ and to generate the proof π anymore, due to the changes in $\mathbf{G}_3$. Furthermore, the only point where we need msk'' is during queries of the form $\textsc{Kdm}(\mathsf{x}, f)$ to extract user secret keys sk''_x. However, in a reduction, we can simulate these extractions using our own key oracle. Using this insight, we build a reduction $\mathcal{B}''$, formally presented in Figure 19. Reduction $\mathcal{B}''$ gets a public key mpk'' as input, samples public and secret key $(\mathsf{mpk}', \mathsf{msk}')$ using algorithm Setup' and simulates the rest of the game $\mathbf{G}_3, \mathbf{G}_4$ for adversary $\mathcal{A}$. For key queries $\textsc{Key}(\mathsf{y})$ the reduction uses its challenge oracle $\textsc{Ch}''(\mathsf{y}, \mathsf{msk}', 0^{|\mathsf{msk}'|})$ to interpolate between the games. As described, it uses its own key oracle $\textsc{Key}''$ to simulate queries of the form $\textsc{Kdm}(\mathsf{x}, f)$. To see the correctness of the reduction, define the set $\mathcal{L}_{sk}$ of attributes $\mathsf{y} \in \mathcal{Y}$ for which

$\mathcal{A}$ issues a query $\textsc{Key}(\mathsf{y})$ and the set $\mathcal{L}_{ch}$ of attributes $\mathsf{x} \in \mathcal{X}$ for which $\mathcal{A}$ issues a query $\textsc{Kdm}(\mathsf{x}, f)$. Note that the reduction $\mathcal{B}''$ issues challenge queries for exactly the attributes in $\mathcal{L}_{sk}$ and key queries exactly for the attributes in $\mathcal{L}_{ch}$. If $\mathcal{A}$ is a valid adversary, these sets satisfy $\mathsf{hit}_{\mathcal{P}''}(\mathcal{L}_{sk}, \mathcal{L}_{ch}) = \mathsf{hit}_{\mathcal{P}}(\mathcal{L}_{ch}, \mathcal{L}_{sk}) = 0$, and hence $\mathcal{B}''$ is valid. Hence,

$$|\mathsf{pr}_3 - \mathsf{pr}_4| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}'',\mathsf{ABE}''}(\lambda).$$

Alg $\mathcal{B}''^{\textsc{Key}'',\textsc{Ch}''}(\mathsf{mpk}'')$

$(\mathsf{mpk}', \mathsf{msk}') \leftarrow \mathsf{Setup}'(1^\lambda)$
$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda)$
$\mathsf{mpk} := (\mathsf{mpk}', \mathsf{mpk}'', \mathsf{crs})$
return $b' \leftarrow \mathcal{A}^{\textsc{Key},\textsc{Kdm}}(\mathsf{mpk})$

Oracle $\textsc{Kdm}(\mathsf{x}, f \in \mathcal{F})$

if $\mathsf{hit}_{\mathcal{P}}(\{\mathsf{x}\}, \mathcal{L}_{sk})$: **return** $\perp$
$\mathcal{L}_{ch} := \mathcal{L}_{ch} \cup \{\mathsf{x}\}$
$\mathsf{ct}_f := \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, f)$
$\mathsf{sk}''_{\mathsf{x}} \leftarrow \textsc{Key}''(\mathsf{x})$
$\hat{\mathsf{C}} := \mathsf{iO}_p(\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_{\mathsf{x}}})$
return $\mathsf{ct} := \hat{\mathsf{C}}$

Oracle $\textsc{Key}(\mathsf{y})$

if $\mathsf{hit}_{\mathcal{P}}(\mathcal{L}_{ch}, \{\mathsf{y}\})$: **return** $\perp$
$\mathcal{L}_{sk} := \mathcal{L}_{sk} \cup \{\mathsf{y}\}$
$\mathsf{sk}'_{\mathsf{y}} \leftarrow \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y})$
$\rho \xleftarrow{\$} \{0,1\}^z$
$\mathsf{m}_0 := \mathsf{msk}', \mathsf{m}_1 := 0^{|\mathsf{msk}'|}$
$\mathsf{ct}_{\mathsf{msk}} \leftarrow \textsc{Ch}''(\mathsf{y}, \mathsf{m}_0, \mathsf{m}_1)$
$\mathsf{stmt} := (\mathsf{ct}_{\mathsf{msk}}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}'')$
$\pi \leftarrow \mathsf{PSim}(\mathsf{crs}, \mathsf{td}, \mathsf{stmt})$
return $\mathsf{sk}_{\mathsf{y}} := (\mathsf{sk}'_{\mathsf{y}}, \mathsf{ct}_{\mathsf{msk}}, \pi)$

Figure 19: The reduction $\mathcal{B}''$, used to interpolate between games $\mathbf{G}_3$ and $\mathbf{G}_4$ in the proof of Theorem 4.7. Circuit $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_{\mathsf{x}}}$ is defined as in Figure 18. Here, Z is the all-zero function.

Game $\mathbf{G}_5$: In game $\mathbf{G}_5$ we change the way we answer queries of the form $\textsc{Kdm}(\mathsf{x}, f)$. First of all, we set $\mathsf{m} := 0^{|f(\mathsf{msk}')|}$ (which was $\mathsf{m} := f(\mathsf{msk}')$ before). This is only a conceptual change, as the variable m has no influence on the output of $\textsc{Kdm}(\mathsf{x}, f)$ since $\mathbf{G}_1$. Secondly, we change the generation of variable ct_f. Recall that in $\mathbf{G}_4$, it was defined as $\mathsf{ct}_f := \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, f)$. Instead, we will now generate it as $\mathsf{ct}_f := \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, Z)$, where Z is the description of a all-zero function. Without loss of generality we can assume that $|f| = |Z|$ by using an appropriate padding. We claim that similarly to the change from $\mathbf{G}_3$ to $\mathbf{G}_4$, the games $\mathbf{G}_4$ and $\mathbf{G}_5$ are indistinguishable. This time, we use a reduction $\mathcal{B}'$ in against the $\mathsf{IND\text{-}CPA}$ security of ABE' that gets the public key mpk' as input, samples keys $(\mathsf{mpk}'', \mathsf{msk}'')$ and simulates games $\mathbf{G}_4, \mathbf{G}_5$ for $\mathcal{A}$. The reduction is formally given in Figure 20. Note that due to the changes we introduced before, the reduction never needs msk' to simulate these games. To simulate queries of the form $\textsc{Key}(\mathsf{y})$, the reduction uses its own oracle $\textsc{Key}'$. To answer queries of the form $\textsc{Kdm}(\mathsf{x}, f)$, $\mathcal{B}'$ can interpolate between $\mathbf{G}_4$ and $\mathbf{G}_5$ using its oracle $\textsc{Ch}'$. The validity of $\mathcal{A}$ transfers directly to the validity of $\mathcal{B}'$, i.e. it never asks for challenge ciphertext and secret keys for attributes x, y with $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$. We obtain

$$|\mathsf{pr}_4 - \mathsf{pr}_5| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}',\mathsf{ABE}'}(\lambda).$$

Games $\mathbf{G}_6$-$\mathbf{G}_9$: From $\mathbf{G}_6$ to $\mathbf{G}_9$ we undo all changes we did from $\mathbf{G}_1$ to $\mathbf{G}_4$. That is, $\mathbf{G}_{5+i}$ is defined as $\mathbf{G}_{4-i}$ for $i \in [4]$, except for the changes we introduced between $\mathbf{G}_4$ and $\mathbf{G}_5$ (i.e the definition of m and ct_f in $\textsc{Kdm}(\mathsf{x}, f)$ queries). In particular, $\mathbf{G}_9$ is as $\mathbf{G}_0$ except that queries of the form $\textsc{Kdm}(\mathsf{x}, f)$ always return an encryption of $0^{|f(\mathsf{msk}')|}$. Thus we have $\mathbf{G}_9 = \mathbf{mKDM\text{-}CPA}_{0,\mathsf{ABE}}$. It is easy to see, that all the arguments used above work again on the path from $\mathbf{G}_5$ to $\mathbf{G}_9$, which shows that there are adversaries $\hat{\mathcal{B}}_1, \hat{\mathcal{B}}_2, \hat{\mathcal{B}}''$

$$\begin{aligned}|\mathsf{pr}_5 - \mathsf{pr}_9| \leq Q_C \cdot \mathsf{Adv}^{\mathsf{iodist}}_{\hat{\mathcal{B}}_1,\mathsf{iO}}(\lambda) + &\mathsf{Adv}^{\mathsf{keydist}}_{\hat{\mathcal{B}}_2,\mathsf{PS}}(\lambda) \\ &+ Q_K \cdot \varepsilon_{\mathsf{zk}} + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\hat{\mathcal{B}}'',\mathsf{ABE}'}(\lambda).\end{aligned}$$

To summarize, using the triangle inequality and the best reductions $\mathcal{B}^*_1, \mathcal{B}^*_2, \mathcal{B}^*_3$ of $\{\mathcal{B}_1, \hat{\mathcal{B}}_1\}, \{\mathcal{B}_2, \hat{\mathcal{B}}_2\}, \{\mathcal{B}'', \hat{\mathcal{B}}''\}$, respectively we obtain the statement. $\square$

Alg $\mathcal{B}'^{\textsc{Key}',\textsc{Ch}'}(\mathsf{mpk}')$

$(\mathsf{mpk}'', \mathsf{msk}'') \leftarrow \mathsf{Setup}''(1^\lambda)$
$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda)$
$\mathsf{mpk} := (\mathsf{mpk}', \mathsf{mpk}'', \mathsf{crs})$
return $b' \leftarrow \mathcal{A}^{\textsc{Key},\textsc{Kdm}}(\mathsf{mpk})$

Oracle $\textsc{Kdm}(\mathsf{x}, f \in \mathcal{F})$

if $\mathsf{hit}_\mathcal{P}(\{\mathsf{x}\}, \mathcal{L}_{sk})$: **return** $\perp$
$\mathcal{L}_{ch} := \mathcal{L}_{ch} \cup \{\mathsf{x}\}$
$\mathsf{m}_0 := f(\mathsf{msk}'), \mathsf{m}_1 := 0^{|f(\mathsf{msk}')|}$
$\mathsf{ct}_f \leftarrow \textsc{Ch}'(\mathsf{x}, \mathsf{m}_0, \mathsf{m}_1)$
$\mathsf{sk}''_\mathsf{x} \leftarrow \mathsf{KeyExt}''(\mathsf{msk}'', \mathsf{x})$
$\hat{\mathsf{C}} := \mathsf{iO}_p(\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}})$
return $\mathsf{ct} := \hat{\mathsf{C}}$

Oracle $\textsc{Key}(\mathsf{y})$

if $\mathsf{hit}_\mathcal{P}(\mathcal{L}_{ch}, \{\mathsf{y}\})$: **return** $\perp$
$\mathcal{L}_{sk} := \mathcal{L}_{sk} \cup \{\mathsf{y}\}$
$\mathsf{sk}'_\mathsf{y} \leftarrow \textsc{Key}'(\mathsf{y})$
$\rho \xleftarrow{\$} \{0,1\}^z$
$\mathsf{ct}_\mathsf{msk} := \mathsf{Enc}''(\mathsf{mpk}'', \mathsf{y}, 0^{|\mathsf{msk}'|}; \rho)$
$\mathsf{stmt} := (\mathsf{ct}_\mathsf{msk}, \mathsf{y}, \mathsf{mpk}', \mathsf{mpk}'')$
$\pi \leftarrow \mathsf{PSim}(\mathsf{crs}, \mathsf{td}, \mathsf{stmt})$
return $\mathsf{sk}_\mathsf{y} := (\mathsf{sk}'_\mathsf{y}, \mathsf{ct}_\mathsf{msk}, \pi)$

Figure 20: The reduction $\mathcal{B}'$, used to interpolate between games $\mathbf{G}_4$ and $\mathbf{G}_5$ in the proof of Theorem 4.7. Circuit $\mathsf{C}_{\mathsf{mpk},\mathsf{x},\mathsf{ct}_f,\mathsf{sk}''_\mathsf{x}}$ is defined as in Figure 18. Here, Z is the all-zero function.

4.3 Constructing ABE with Verifiable User Secret Keys

We show how to add verifiability of user secret keys (cf. Definition 4.1) to any perfectly complete attribute-based encryption scheme, while preserving IND-CPA security. The idea is to add a NIZK proof to user secret keys to make them verifiable. Formally, let $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$ be an attribute-based encryption scheme for some predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. We assume that ABE' is perfectly complete. For simplicity, we assume that both Setup' and KeyExt' use λ bits of randomness. Our goal is to construct a new attribute-based encryption scheme for predicate $\mathcal{P}$, such that user secret keys are verifiable (cf. Definition 4.1). The idea is to add a NIZK proof to user secret keys. More precisely, consider the relation

$$\mathcal{R} := \left\{ ((\mathsf{mpk}', \mathsf{sk}'_\mathsf{y}), (\rho_0, \mathsf{msk}', \rho)) \;\middle|\; \begin{array}{rl} \mathsf{Setup}'(1^\lambda; \rho_0) & = (\mathsf{mpk}', \mathsf{msk}') \\ \wedge\ \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y}; \rho) & = \mathsf{sk}'_\mathsf{y} \end{array} \right\},$$

and a perfectly sound and perfectly complete non-interactive zero-knowledge proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$ for $\mathcal{R}$. Our transformed scheme $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ for predicate $\mathcal{P}$ and the associated algorithm VerK are given in Figure 21. We show that ABE has verifiable

Alg $\mathsf{Setup}(1^\lambda)$

$\rho_0 \xleftarrow{\$} \{0,1\}^\lambda$
$(\mathsf{mpk}', \mathsf{msk}') \leftarrow \mathsf{Setup}'(1^\lambda; \rho_0)$
$\mathsf{crs} \leftarrow \mathsf{PGen}(1^\lambda)$
$\mathsf{mpk} := (\mathsf{mpk}', \mathsf{crs})$
$\mathsf{msk} := (\mathsf{msk}', \rho_0)$
return $(\mathsf{mpk}, \mathsf{msk})$

Alg $\mathsf{VerK}(\mathsf{mpk} = (\mathsf{mpk}', \mathsf{crs}), \mathsf{x}, \mathsf{sk}_\mathsf{y})$

let $\mathsf{sk}_\mathsf{y} = (\mathsf{sk}'_\mathsf{y}, \pi)$
if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 0$: **return** 0
$\mathsf{stmt} := (\mathsf{mpk}', \mathsf{sk}'_\mathsf{y})$
return $\mathsf{PVer}(\mathsf{crs}, \mathsf{stmt}, \pi)$

Alg $\mathsf{KeyExt}(\mathsf{msk} = (\mathsf{msk}', \rho_0), \mathsf{y})$

$\rho \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{sk}'_\mathsf{y} \leftarrow \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y}; \rho)$
$\mathsf{stmt} := (\mathsf{mpk}', \mathsf{sk}'_\mathsf{y})$
$\mathsf{witn} := (\rho_0, \mathsf{msk}', \rho)$
$\pi \leftarrow \mathsf{PProve}(\mathsf{crs}, \mathsf{stmt}, \mathsf{witn})$
return $\mathsf{sk}_\mathsf{y} := (\mathsf{sk}'_\mathsf{y}, \pi)$

Alg $\mathsf{Enc}(\mathsf{mpk} = (\mathsf{mpk}', \mathsf{crs}), \mathsf{x}, \mathsf{m})$

return $\mathsf{ct} \leftarrow \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, \mathsf{m})$

Alg $\mathsf{Dec}(\mathsf{sk}_\mathsf{y} = (\mathsf{sk}'_\mathsf{y}, \pi), \mathsf{ct})$

return $\mathsf{Dec}'(\mathsf{sk}'_\mathsf{y}, \mathsf{ct})$

Figure 21: The attribute-based encryption scheme $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ for a given attribute-based encryption scheme $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$ and a non-interactive zero-knowledge proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$.

user secret keys, and the transformation preserves IND-CPA security.

Lemma 4.8 (Verifiable User Secret Keys). *If* ABE' *is perfectly complete, and* PS *is perfectly complete and perfectly sound, then* ABE *has verifiable user secret keys.*

Proof. We consider algorithm VerK as in Figure 21. Let $(\mathsf{mpk}, \mathsf{msk}) \in \mathsf{Setup}(1^\lambda)$. Write $\mathsf{mpk} = (\mathsf{mpk}', \mathsf{crs})$. Let $\mathsf{x} \in \mathcal{X}$ and $\mathsf{sk}_\mathsf{y} = (\mathsf{sk}'_\mathsf{y}, \pi)$ be attributes such that $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$. If $\mathsf{sk}_\mathsf{y} \in \mathsf{KeyExt}(\mathsf{msk}, \mathsf{y})$, then by completeness of PS, we have $\mathsf{VerK}(\mathsf{mpk} = (\mathsf{mpk}', \mathsf{crs}), \mathsf{x}, \mathsf{sk}_\mathsf{y}) = \mathsf{PVer}(\mathsf{crs}, (\mathsf{mpk}', \mathsf{sk}'_\mathsf{y}), \pi) = 1$. On the other hand, assume that $\mathsf{VerK}(\mathsf{mpk} = (\mathsf{mpk}', \mathsf{crs}), \mathsf{x}, \mathsf{sk}_\mathsf{y}) = \mathsf{PVer}(\mathsf{crs}, (\mathsf{mpk}', \mathsf{sk}'_\mathsf{y}), \pi) = 1$. Then, by perfect soundness of PS, we know that there is some witness $(\rho_0, \mathsf{msk}', \rho)$ such that $((\mathsf{mpk}', \mathsf{crs}), (\rho_0, \mathsf{msk}', \rho)) \in \mathcal{R}$. By definition of $\mathcal{R}$, we have $(\mathsf{mpk}', \mathsf{msk}') \in \mathsf{Setup}'(1^\lambda)$ and $\mathsf{sk}'_\mathsf{y} \in \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y})$. By perfect completeness of ABE', we know that $\mathsf{Dec}(\mathsf{sk}_\mathsf{y}, \mathsf{Enc}(\mathsf{mpk}, \mathsf{x}, \mathsf{m})) = \mathsf{m}$ for all messages m. $\square$

Lemma 4.9 (Security). *Let* PS *be a* $\varepsilon_{\mathsf{zk}}$*-NIZK proof system for the relation* $\mathcal{R}$*, and assume that* ABE' *is* $\mathsf{IND\text{-}CPA}$ *secure. Then* ABE *is* $\mathsf{IND\text{-}CPA}$ *secure. In particular, for every PPT algorithm* $\mathcal{A}$ *there are PPT algorithms* $\mathcal{B}_1, \mathcal{B}_2$ *with* $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{A})$ *and*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) \leq 2Q_K \cdot \varepsilon_{\mathsf{zk}} + 2 \cdot \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}_1,\mathsf{PS}}(\lambda) + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_2,\mathsf{ABE}'}(\lambda).$$

Proof. Let $\mathcal{A}$ be an efficient adversary against the $\mathsf{IND\text{-}CPA}$ security of ABE. We prove the statement via a sequence of games $\mathbf{G}_0$ - $\mathbf{G}_5$. For each game $\mathbf{G}_i$, we denote the probability that it outputs 1 by pr_i, namely,

$$\mathsf{pr}_i := \Pr\left[\mathbf{G}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right].$$

Game $\mathbf{G}_0$: $\mathbf{G}_0$ is $\mathbf{IND\text{-}CPA}_0$. To recall, the game first generates $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathcal{A}$ is given mpk and access to oracles $\textsc{Key}, \textsc{Ch}$, where $\textsc{Ch}$ returns an encryption of m_0 under attribute x on input $\mathsf{x}, \mathsf{m}_0, \mathsf{m}_1$. Write $\mathsf{mpk} = (\mathsf{mpk}', \mathsf{crs})$, where $(\mathsf{mpk}', \mathsf{msk}') \in \mathsf{Setup}'(1^\lambda)$ and $\mathsf{crs} \in \mathsf{PGen}(1^\lambda)$. We have

$$\mathsf{pr}_0 = \Pr\left[\mathbf{IND\text{-}CPA}^{\mathcal{A}}_{0,\mathsf{ABE}}(\lambda) \Rightarrow 1\right].$$

Game $\mathbf{G}_1$: In $\mathbf{G}_1$, we modify how crs (contained in mpk) is generated. Namely, we generate crs with a trapdoor td via $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda)$. A straight-forward reduction $\mathcal{B}_1$ shows that

$$|\mathsf{pr}_0 - \mathsf{pr}_1| \leq \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}_1,\mathsf{PS}}(\lambda).$$

Game $\mathbf{G}_2$: In $\mathbf{G}_2$, we change how proofs π that are part of user secret keys sk_y are generated in queries of the form $\textsc{Key}(\mathsf{y})$. Namely, instead of using the witness $(\rho_0, \mathsf{msk}', \rho)$, we now compute them using algorithm PSim, i.e. $\pi \leftarrow \mathsf{PSim}(\mathsf{crs}, \mathsf{td}, \mathsf{stmt})$. The games are statistically close by the zero-knowledge property of PS. Concretely, we have

$$|\mathsf{pr}_1 - \mathsf{pr}_2| \leq Q_K \cdot \varepsilon_{\mathsf{zk}}.$$

Game $\mathbf{G}_3$: In $\mathbf{G}_3$, we change oracle $\textsc{Ch}$. Namely, from now on, it returns an encryption of m_1 under attribute x on input $\mathsf{x}, \mathsf{m}_0, \mathsf{m}_1$. It is easy to see that games $\mathbf{G}_2$ and $\mathbf{G}_3$ are indistinguishable, assuming $\mathsf{IND\text{-}CPA}$ security of ABE'. This is because in game $\mathbf{G}_2$, we only need msk' to simulate the oracle $\textsc{Key}$. Therefore, a reduction $\mathcal{B}_2$ against the $\mathsf{IND\text{-}CPA}$ security of ABE' can interpolate between $\mathbf{G}_2$ and $\mathbf{G}_3$. Concretely, reduction $\mathcal{B}_2$ gets mpk' as input and oracle access to oracles $\textsc{Key}', \textsc{Ch}'$. It generates $\mathsf{crs} \leftarrow \mathsf{PTrapGen}(1^\lambda)$ and sets $\mathsf{mpk} = (\mathsf{mpk}', \mathsf{crs})$. Then, it runs $\mathcal{A}$ on inoput mpk. It then uses its own key and challenge oracles $\textsc{Key}', \textsc{Ch}'$ to simulate the oracles $\textsc{Key}, \textsc{Ch}'$ for $\mathcal{A}$. Finally, it outputs whatever $\mathcal{A}$ outputs. We have

$$|\mathsf{pr}_2 - \mathsf{pr}_3| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_2,\mathsf{ABE}'}(\lambda).$$

Game $\mathbf{G}_4$: In this game, we undo the change that we introduced in game $\mathbf{G}_2$. Namely, we generate proofs π using the witness again. As before, we have

$$|\mathsf{pr}_3 - \mathsf{pr}_4| \leq Q_K \cdot \varepsilon_{\mathsf{zk}}.$$

Game $\mathbf{G}_5$: In this game, we undo the change that we introduced in game $\mathbf{G}_1$. Namely, we generate crs via $\mathsf{crs} \leftarrow \mathsf{PGen}(1^\lambda)$ again. As before, we have

$$|\mathsf{pr}_4 - \mathsf{pr}_5| \leq \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}_1,\mathsf{PS}}(\lambda).$$

Finally, we note that game $\mathbf{G}_5$ is identical to game $\mathbf{IND\text{-}CPA}_1$, finishing the proof. $\square$

4.4 Constructing ABE with Uniquely Verifiable Master Secret Keys

We show how to achieve uniqueness of the master secret key from any attribute-based encryption scheme, nearly for free. The only ingredient we need is a public key encryption scheme having unique secret keys, which is easier to achieve. At a high level, we can add an encryption of the master secret key under the public key encryption scheme to the master public key. Using the fact that we defined verifiability for master secret keys in a syntactical way, we can show that this satisfies the definition of uniquely verifiable master secret keys. In this way, our construction may still have many different master secret keys per master public key that are functional, but only one that is a possible output of the honest algorithm Setup. Let us define this type of public key encryption scheme formally.

Definition 4.10 (Uniquely Verifiable Secret Keys). Consider a public key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. We say that PKE has uniquely verifiable secret keys, if there exists a deterministic polynomial time algorithm $\mathsf{VerK}_{\mathsf{PKE}}$ satisfying the following properties:

- $\mathsf{VerK}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk})$ takes as input a public key mpk and a secret key sk and outputs a bit $b \in \{0, 1\}$.
- For all $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda)$ and all sk' we have:
$$\mathsf{VerK}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}') = 1 \iff (\mathsf{pk}, \mathsf{sk}') \in \mathsf{Gen}(1^\lambda).$$
- For all $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda)$ there does not exists a key $\mathsf{sk}' \neq \mathsf{sk}$ such that $\mathsf{VerK}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}') = 1$.

Now, let $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$ be an attribute-based encryption scheme for some predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$. Further, let $\mathsf{PKE} = (\mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}})$ be a public key encryption scheme. We assume that PKE has uniquely verifiable secret keys, perfect completeness, and that we can encrypt master secret keys of ABE' using PKE. Then, we construct a new attribute-based encryption scheme $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ for the same predicate $\mathcal{P}$ in Figure 22. The message space remains unchanged. We show that if ABE' is perfectly complete and $\mathsf{IND\text{-}CPA}$ secure, then ABE also satisfies these properties and additionally has uniquely verifiable master secret keys. Further, if ABE' has verifiable user secret keys, then so has ABE. The idea is to add an encryption under PKE of the master secret key to the public key. We highlight that this construction relies heavily on the fact that we defined verifiability for master secret keys in a syntactical way. That is, our construction may still have many different master secret keys per master public key that are functional, but only one that is a possible output of Setup. As key extraction and encryption essentially remained unchanged, it is clear

Alg $\mathsf{Setup}(1^\lambda)$
- $(\mathsf{mpk}', \mathsf{msk}') \leftarrow \mathsf{Setup}'(1^\lambda)$
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)$
- $\mathsf{ct}_{\mathsf{msk}'} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{msk}')$
- $\mathsf{mpk} := (\mathsf{mpk}', \mathsf{pk}, \mathsf{ct}_{\mathsf{msk}'})$
- $\mathsf{msk} := (\mathsf{msk}', \mathsf{sk})$
- **return** $(\mathsf{mpk}, \mathsf{msk})$

Alg $\mathsf{KeyExt}(\mathsf{msk} = (\mathsf{msk}', \mathsf{sk}), \mathsf{y})$
- **return** $\mathsf{sk}_\mathsf{y} \leftarrow \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y})$

Alg $\mathsf{Enc}(\mathsf{mpk} = (\mathsf{mpk}', \mathsf{pk}, \mathsf{ct}_{\mathsf{msk}'}), \mathsf{x}, \mathsf{m})$
- **return** $\mathsf{ct} \leftarrow \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, \mathsf{m})$

Alg $\mathsf{Dec}(\mathsf{sk}_\mathsf{y}, \mathsf{ct})$
- **return** $\mathsf{Dec}'(\mathsf{sk}_\mathsf{y}, \mathsf{ct})$

Figure 22: The attribute-based encryption scheme $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ for a given attribute-based encryption scheme $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$ and an encryption scheme $\mathsf{PKE} = (\mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}})$.

that completeness and verifiability of user secret keys is preserved. We will now show that ABE has uniquely verifiable master secret keys and remains $\mathsf{IND\text{-}CPA}$ secure.

Lemma 4.11 (Uniquely Verifiable Master Secret Keys). *If* PKE *is perfectly complete and has uniquely verifiable secret keys, then* ABE *has uniquely verifiable master secret keys.*

Proof. We present a deterministic polynomial time algorithm VerMK in Figure 23. Here, we assume that PKE has uniquely verifiable secret keys with algorithm $\mathsf{VerK}_{\mathsf{PKE}}$. Let $\mathsf{mpk} = (\mathsf{mpk}', \mathsf{pk}, \mathsf{ct}_{\mathsf{msk}'})$ be a

Alg $\mathsf{VerMK}(\mathsf{mpk}, \mathsf{msk})$
let $\mathsf{mpk} = (\mathsf{mpk}', \mathsf{pk}, \mathsf{ct}_{\mathsf{msk}'})$
let $\mathsf{msk} = (\mathsf{msk}', \mathsf{sk})$
if $\mathsf{VerK}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}) = 0$: **return** 0
if $\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{sk}, \mathsf{ct}_{\mathsf{msk}'}) \neq \mathsf{msk}'$: **return** 0
return 1

Figure 23: The deterministic polynomial time algorithm VerMK for attribute-based encryption scheme ABE. Here, we assume that PKE has uniquely verifiable secret keys with algorithm $\mathsf{VerK}_{\mathsf{PKE}}$.

honestly generated master public key, i.e. $(\mathsf{mpk}, \mathsf{msk}) \in \mathsf{Setup}(1^\lambda)$ for some msk. First of all, it easily follows from the definitions and perfect completeness of PKE that for all $\widetilde{\mathsf{msk}}$ we have

$$(\mathsf{mpk}, \widetilde{\mathsf{msk}}) \in \mathsf{Setup}(1^\lambda) \Longrightarrow \mathsf{VerMK}(\mathsf{mpk}, \widetilde{\mathsf{msk}}) = 1.$$

Next, we show that for any $\mathsf{msk}_0, \mathsf{msk}_1$, we have:

$$(\mathsf{VerMK}(\mathsf{mpk}, \mathsf{msk}_0) = 1 \wedge \mathsf{VerMK}(\mathsf{mpk}, \mathsf{msk}_1) = 1) \Longrightarrow \mathsf{msk}_0 = \mathsf{msk}_1.$$

Note that this is already sufficient to show unique verifiability of master secret keys. To prove this, assume $\mathsf{VerMK}(\mathsf{mpk}, \mathsf{msk}_0) = 1$ and $\mathsf{VerMK}(\mathsf{mpk}, \mathsf{msk}_1) = 1$ and let $\mathsf{msk}_0 = (\mathsf{msk}'_0, \mathsf{sk}_0), \mathsf{msk}_1 = (\mathsf{msk}'_1, \mathsf{sk}_1)$. Then, by definition of VerMK it holds that $\mathsf{VerK}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}_0) = 1$ and $\mathsf{VerK}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}_1) = 1$. As PKE has uniquely verifiable secret keys, we have $\mathsf{sk}_0 = \mathsf{sk}_1$. Further, by definition of VerMK it holds that $\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{sk}_0, \mathsf{ct}_{\mathsf{msk}'}) = \mathsf{msk}'_0$ and $\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{sk}_1, \mathsf{ct}_{\mathsf{msk}'}) = \mathsf{msk}'_1$. In combination we obtain

$$\mathsf{msk}'_0 = \mathsf{Dec}_{\mathsf{PKE}}(\mathsf{sk}_0, \mathsf{ct}_{\mathsf{msk}'}) = \mathsf{Dec}_{\mathsf{PKE}}(\mathsf{sk}_1, \mathsf{ct}_{\mathsf{msk}'}) = \mathsf{msk}'_1,$$

which finishes the proof. Finally, we want to note that we do not need any verifiability of the master secret keys of ABE', as Definition 4.2 only deals with honestly generated master public keys. □

Lemma 4.12 (Security). *If* PKE *is* $\mathsf{IND\text{-}CPA}$ *secure and* ABE' *is* $\mathsf{IND\text{-}CPA}$ *secure, then* ABE *is* $\mathsf{IND\text{-}CPA}$ *secure. In particular, for every PPT algorithm* $\mathcal{A}$ *there are PPT algorithms* $\mathcal{B}_1, \mathcal{B}_2$ *with* $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{A})$ *and*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_1,\mathsf{PKE}}(\lambda) + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_2,\mathsf{ABE}'}(\lambda).$$

Proof. Let $\mathcal{A}$ be an efficient adversary against the $\mathsf{IND\text{-}CPA}$ security of ABE. We prove the statement via a sequence of games $\mathbf{G}_0$ - $\mathbf{G}_3$, as defined in Figure 24. For each game $\mathbf{G}_i$, we denote the probability that it outputs 1 by pr_i, namely,

$$\mathsf{pr}_i := \Pr\left[\mathbf{G}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right].$$

Game $\mathbf{G}_0$: $\mathbf{G}_0$ is defined to be the game $\mathbf{IND\text{-}CPA}_0$. That is, the game first generates $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathcal{A}$ is given mpk and access to oracles $\textsc{Key}, \textsc{Ch}$, where $\textsc{Ch}$ returns an encryption of m_0 under attribute x on input $\mathsf{x}, \mathsf{m}_0, \mathsf{m}_1$. Recall that $\mathsf{mpk} = (\mathsf{mpk}', \mathsf{pk}, \mathsf{ct}_{\mathsf{msk}'})$, where $(\mathsf{mpk}', \mathsf{msk}') \in \mathsf{Setup}'(1^\lambda)$, pk is a public key of the scheme PKE and $\mathsf{ct}_{\mathsf{msk}'}$ is an encryption of msk' under pk. Clearly, we have

$$\mathsf{pr}_0 = \Pr\left[\mathbf{IND\text{-}CPA}^{\mathcal{A}}_{0,\mathsf{ABE}}(\lambda) \Rightarrow 1\right].$$

Game $\mathbf{G}_1$: In $\mathbf{G}_1$, we change the public key mpk that is given to the adversary. In particular, we set $\mathsf{ct}_{\mathsf{msk}'} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, 0^{|\mathsf{msk}'|})$. Indistinguishability follows from a straight-forward reduction $\mathcal{B}_1$ against the $\mathsf{IND\text{-}CPA}$ security of PKE. Note that this is possible, as sk is never needed during the simulation of the game, in particular, although it is formally part of msk, only msk' is needed to simulate $\textsc{Key}$. We obtain

$$|\mathsf{pr}_0 - \mathsf{pr}_1| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_1,\mathsf{PKE}}(\lambda).$$

Game $\mathbf{G}_2$: In $\mathbf{G}_2$, we change the way oracle $\textsc{Ch}$ is simulated. In particular, $\textsc{Ch}$ now returns an encryption of m_1 under attribute x on input $\mathsf{x}, \mathsf{m}_0, \mathsf{m}_1$. Note that in game $\mathbf{G}_1$ we only need msk' to simulate the oracle Key. Thus, a reduction $\mathcal{B}_2$ against the IND-CPA security of ABE′ can interpolate between $\mathbf{G}_1$ and $\mathbf{G}_2$. That is, reduction $\mathcal{B}_2$ gets mpk' as input and oracle access to oracles Key', Ch'. It generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)$ and runs $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk}', \mathsf{pk}, \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, 0^{|\mathsf{msk}'|}))$. It then uses its own key and challenge oracles Key', Ch' to simulate the oracles Key, Ch' for $\mathcal{A}$. Finally, it returns whatever $\mathcal{A}$ outputs. We have

$$|\mathsf{pr}_1 - \mathsf{pr}_2| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_2,\mathsf{ABE}'}(\lambda).$$

<u>**Game $\mathbf{G}_3$:**</u> In $\mathbf{G}_3$, we undo the change we did in $\mathbf{G}_1$. That is, we set $\mathsf{ct}_{\mathsf{msk}'} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{msk}')$ as in the real scheme. Similarly to the transition from $\mathbf{G}_0$ to $\mathbf{G}_1$, we obtain

$$|\mathsf{pr}_2 - \mathsf{pr}_3| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_1,\mathsf{PKE}}(\lambda).$$

Finally, note that $\mathbf{G}_3$ is equivalent to the real IND-CPA game with respect to ABE and bit $b = 1$, namely

$$\mathsf{pr}_3 = \Pr\left[\textbf{IND-CPA}^{\mathcal{A}}_{1,\mathsf{ABE}}(\lambda) \Rightarrow 1\right],$$

which finishes the proof. □

Game $\mathbf{G}_0$-$\mathbf{G}_3$		**Oracle $\text{Ch}_b(\mathsf{x}, \mathsf{m}_0, \mathsf{m}_1)$**	
$(\mathsf{mpk}', \mathsf{msk}') \leftarrow \mathsf{Setup}'(1^\lambda)$		**if** $\mathsf{hit}_{\mathcal{P}}(\{\mathsf{x}\}, \mathcal{L}_{sk})$: **return** $\bot$	
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)$		$\mathcal{L}_{ch} := \mathcal{L}_{ch} \cup \{\mathsf{x}\}$	
$\mathsf{ct}_{\mathsf{msk}'} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{msk}')$	// $\mathbf{G}_0, \mathbf{G}_3$	**if** $\lvert\mathsf{m}_0\rvert \neq \lvert\mathsf{m}_1\rvert$: **return** $\bot$	
$\mathsf{ct}_{\mathsf{msk}'} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, 0^{\lvert\mathsf{msk}'\rvert})$	// $\mathbf{G}_1, \mathbf{G}_2$	$\mathsf{ct} \leftarrow \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, \mathsf{m}_0)$	// $\mathbf{G}_0, \mathbf{G}_1$
$\mathsf{mpk} := (\mathsf{mpk}', \mathsf{pk}, \mathsf{ct}_{\mathsf{msk}'})$		$\mathsf{ct} \leftarrow \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, \mathsf{m}_1)$	// $\mathbf{G}_2, \mathbf{G}_3$
$\mathsf{msk} := (\mathsf{msk}', \mathsf{sk})$		**return** ct	
return $b' \leftarrow \mathcal{A}^{\text{Key},\text{Ch}}(\mathsf{mpk})$			

Figure 24: The games $\mathbf{G}_0$-$\mathbf{G}_3$ in the proof of Lemma 4.12. Lines with highlighted comments are only executed in the corresponding games. Oracle Key is as in Figure 5.

4.5 From mKDM-CPA to mKDM-CCA

In this section we show how to turn any mKDM-CPA secure attribute-based encryption scheme into an mKDM-CCA secure one. In combination with the construction in Section 4.2 we obtain a generic mKDM-CCA secure construction in the standard model.

To do that, we use an IND-CPA secure public key encryption scheme and a simulation-sound NIZK proof system. The intuition is to encrypt with both schemes and add a proof, which is similar to the well-known construction of Naor and Yung [NY90, CCS09] for public key encryption. Let $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$ be an mKDM-CPA secure attribute-based encryption scheme for a predicate $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, and $\mathsf{PKE} = (\mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}})$ be an IND-CPA secure public key encryption scheme. We assume that both support the same message space and encryption randomness of both has length $z = z(\lambda)$. Further, we let $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$ be a simulation-sound NIZK proof system for the relation

$$\mathcal{R}_{cca} := \left\{ ((\mathsf{mpk}', \mathsf{pk}'', \mathsf{x}, \mathsf{ct}', \mathsf{ct}''), (\mathsf{m}, \rho', \rho'')) \middle| \begin{array}{ll} \mathsf{ct}' & = \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, \mathsf{m}; \rho') \wedge \\ \mathsf{ct}'' & = \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}'', \mathsf{m}; \rho'') \end{array} \right\}.$$

That is, PS allows to prove that two ciphertexts encrypt the same message.

Using these building blocks, we define a new scheme $\mathsf{ABE}_{cca}[\mathsf{ABE}', \mathsf{PKE}, \mathsf{PS}]$ for the same predicate $\mathcal{P}$ in Figure 25. Completeness of $\mathsf{ABE}_{cca}[\mathsf{IBE}', \mathsf{PKE}, \mathsf{PS}]$ follows immediately from the completeness of $\mathsf{ABE}', \mathsf{PKE}, \mathsf{PS}$.

Theorem 4.13 *Let $\mathcal{F}$ be some class of functions, and* $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$ *be an* $\mathcal{F}$-mKDM-CPA *secure attribute-based encryption scheme, and* $\mathsf{PKE} = (\mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}})$ *be an* IND-CPA *secure public key encryption scheme. Let* $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$ *be an* $\varepsilon_{\mathsf{sso}}$*-simulation-sound* $(\rho, \varepsilon_{\mathsf{so}}, \varepsilon_{\mathsf{zk}})$*-NIZK proof system for the relation* $\mathcal{R}_{cca}$.

Alg $\mathsf{Setup}(1^\lambda)$

$(\mathsf{mpk}', \mathsf{msk}') \leftarrow \mathsf{Setup}'(1^\lambda)$
$(\mathsf{pk}'', \mathsf{sk}'') \leftarrow \mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)$
$\mathsf{crs} \leftarrow \mathsf{PGen}(1^\lambda)$
$\mathsf{mpk} := (\mathsf{mpk}', \mathsf{pk}'', \mathsf{crs})$
$\mathsf{msk} := \mathsf{msk}'$
return $(\mathsf{mpk}, \mathsf{msk})$

Alg $\mathsf{Dec}(\mathsf{sk}_\mathsf{y}, \mathsf{ct} = (\mathsf{x}, \mathsf{ct}', \mathsf{ct}'', \pi))$

$\mathsf{stmt} := (\mathsf{mpk}', \mathsf{pk}'', \mathsf{x}, \mathsf{ct}', \mathsf{ct}'')$
if $\mathsf{PVer}(\mathsf{crs}, \mathsf{stmt}, \pi) = 0$: **return** $\perp$
if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 0$: **return** $\perp$
return $\mathsf{Dec}'(\mathsf{sk}_\mathsf{y}, \mathsf{ct}')$

Alg $\mathsf{Enc}(\mathsf{mpk} = (\mathsf{mpk}', \mathsf{pk}'', \mathsf{crs}), \mathsf{x}, \mathsf{m})$

$\rho', \rho'' \xleftarrow{\$} \{0,1\}^z$
$\mathsf{ct}' \leftarrow \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, \mathsf{m}; \rho')$
$\mathsf{ct}'' \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}'', \mathsf{m}; \rho'')$
$\mathsf{stmt} := (\mathsf{mpk}', \mathsf{pk}'', \mathsf{x}, \mathsf{ct}', \mathsf{ct}'')$
$\mathsf{witn} := (\mathsf{m}, \rho', \rho'')$
$\pi \leftarrow \mathsf{PProve}(\mathsf{crs}, \mathsf{stmt}, \mathsf{witn})$
return $\mathsf{ct} := (\mathsf{x}, \mathsf{ct}', \mathsf{ct}'', \pi)$

Alg $\mathsf{KeyExt}(\mathsf{msk} = \mathsf{msk}', \mathsf{y})$

$\mathsf{sk}_\mathsf{y} \leftarrow \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y})$
return sk_y

Figure 25: The attribute-based encryption scheme $\mathsf{ABE}_{cca}[\mathsf{ABE}', \mathsf{PKE}, \mathsf{PS}] = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ for a given attribute-based encryption scheme $\mathsf{ABE}' = (\mathsf{Setup}', \mathsf{KeyExt}', \mathsf{Enc}', \mathsf{Dec}')$, public key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}})$ and a proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PTrapGen}, \mathsf{PProve}, \mathsf{PVer}, \mathsf{PSim})$.

Then $\mathsf{ABE} := \mathsf{ABE}_{cca}[\mathsf{ABE}', \mathsf{PKE}, \mathsf{PS}]$ *is* $\mathcal{F}$-mKDM-CCA *secure. In particular, for every PPT algorithm* $\mathcal{A}$ *making* Q_C, Q_K, Q_D *queries to the oracles* KDM, KEY, DEC, *respectively, there are PPT algorithms* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *with* $\mathbf{T}(\mathcal{B}_i) \approx \mathbf{T}(\mathcal{A})$ *for* $i \in \{1,2,3\}$ *and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{mKDM\text{-}CCA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) \leq 2Q_D \cdot \varepsilon_{\mathsf{sso}} + 2Q_C \cdot \varepsilon_{\mathsf{zk}} + \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}_1,\mathsf{PS}}(\lambda)\\ + Q_C \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_2,\mathsf{PKE}}(\lambda) + \mathsf{Adv}^{\mathsf{mKDM\text{-}CPA}}_{\mathcal{B}_3,\mathsf{ABE}'}(\lambda).\end{aligned}$$

Proof. We show the statement via a sequence of games $\mathbf{G}_0$-$\mathbf{G}_8$. The most important games are formally presented in Figure 26. For $0 \leq i \leq 8$ we define

$$\mathsf{pr}_i := \Pr[\mathbf{G}_i \Rightarrow 1].$$

Recall that we have to show that

$$\left|\Pr\left[\mathbf{mKDM\text{-}CCA}^{\mathcal{A}}_{0,\mathsf{ABE}}(\lambda) \Rightarrow 1\right] - \Pr\left[\mathbf{mKDM\text{-}CCA}^{\mathcal{A}}_{1,\mathsf{ABE}}(\lambda) \Rightarrow 1\right]\right|$$

is negligible.

Game $\mathbf{G}_0$: We set $\mathbf{G}_0 = \mathbf{mKDM\text{-}CCA}_{1,\mathsf{ABE}}$. Recall that in this game the adversary $\mathcal{A}$ gets access to oracles KEY, KDM and DEC and $\text{KDM}(\mathsf{x}, f)$ returns an encryption of $f(\mathsf{msk})$ under attribute x. In the scheme ABE this encryption contains attribute x, ciphertexts ct' and ct'', as well as a proof π showing that both encrypt the same message. Also, recall that the decryption oracle DEC uses a user secret key derived from msk' to decrypt given ciphertext. We already introduce a (purely conceptual) change in oracle DEC. Namely, while in the real game this oracle on input $(\mathsf{y}, \mathsf{ct})$ checks if there is some x such that $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$ and $(\mathsf{x}, \mathsf{ct}) \in \mathcal{L}_{ct}$, our game now only checks if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$ and $(\mathsf{x}, \mathsf{ct}) \in \mathcal{L}_{ct}$ for the x that is contained in $\mathsf{ct} = (\mathsf{x}, \mathsf{ct}', \mathsf{ct}'', \pi)$. This is equivalent, by the definition of list $\mathcal{L}_{ct}$.

Game $\mathbf{G}_1$: In this game, we change how the public key is generated. Namely, we generate crs in combination with a trapdoor td using algorithm $\mathsf{PTrapGen}$ instead of using algorithm PGen. Note that a direct reduction $\mathcal{B}_1'$ from the CRS indistinguishability of PS shows that

$$|\mathsf{pr}_0 - \mathsf{pr}_1| \leq \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}_1',\mathsf{PS}}(\lambda).$$

Game $\mathbf{G}_2$: Recall that a challenge ciphertext (i.e. a ciphertext returned by $\text{KDM}(\mathsf{x}, f)$) has the form $\mathsf{ct} = (\mathsf{x}, \mathsf{ct}', \mathsf{ct}'', \pi)$. In $\mathbf{G}_2$, we change how π is generated when $\mathcal{A}$ calls $\text{KDM}(\mathsf{x}, f)$. That is, we generate it by using the simulator PSim instead of PProve. Note that we can apply a hybrid over all Q_C queries using the zero-knowledge property of PS to obtain

$$|\mathsf{pr}_1 - \mathsf{pr}_2| \leq Q_C \cdot \varepsilon_{\mathsf{zk}}.$$

Also note that from now on, we do not longer need the witness $(f(\mathsf{msk}), \rho', \rho'')$ to answer challenge queries.

Game $\mathbf{G}_3$: In $\mathbf{G}_3$ we change the challenge ciphertexts again. This time, we change how ct'' is generated. Recall that until $\mathbf{G}_2$, it was computed as an encryption of $f(\mathsf{msk})$, i.e $\mathsf{ct}'' = \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}'', f(\mathsf{msk}); \rho'')$. Now, we generate it as $\mathsf{ct}'' := \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}'', 0^{|f(\mathsf{msk})|}; \rho'')$. Note that at this point, the game can be simulated without knowing sk'', as $\mathsf{msk} = \mathsf{msk}'$ does not contain sk''. Thus, a sequence of Q_C direct reductions from the $\mathsf{IND\text{-}CPA}$ security of PKE shows

$$|\mathsf{pr}_2 - \mathsf{pr}_3| \leq Q_C \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_2,\mathsf{PKE}}(\lambda).$$

Game $\mathbf{G}_4$: In $\mathbf{G}_4$, we change the way decryption queries $\textsc{Dec}(\mathsf{y}, \mathsf{ct})$ for $\mathsf{ct} = (\mathsf{x}, \mathsf{ct}', \mathsf{ct}'', \pi)$ are answered. Recall that until this point, the decryption oracle derives a user secret key sk_y for attribute y from msk' and decrypts ct using sk_y. Also, note that this decryption process involves verifying the proof π, checking if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$, and decrypting ct' using sk_y. In $\mathbf{G}_4$, we still verify the proof and check if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1$, but decrypt ct'' using sk'' instead. Note that this can only result in a difference visible to the adversary, if ct' and ct'' encrypt different messages but the proof π still verifies. Denote the event that this happens in the ith query by bad_i for $i \in [Q_D]$. For each i, we can bound the probability of bad_i using the simulation-soundness of PS. That is, we construct a (non-efficient) reduction $\hat{B}_i$ that wins the game **SIMSO** if bad_i happens. The reduction gets as input crs and sets up all the keys as in $\mathbf{G}_3$. To simulate oracle queries of the form $\textsc{Kdm}(\mathsf{x}, f)$, it uses its own oracle $\textsc{Sim}$. In the ith query of the form $\textsc{Dec}(\mathsf{y}, \mathsf{ct} = (\mathsf{x}, \mathsf{ct}', \mathsf{ct}'', \pi))$ it returns $\perp$ if $(\mathsf{x}, \mathsf{ct})$ is in list $\mathcal{L}_{ct}$. Otherwise, it checks if bad_i happens (this is why the reduction is not efficient) and if so, it returns the statement $\mathsf{stmt} := (\mathsf{mpk}', \mathsf{pk}'', \mathsf{x}, \mathsf{ct}', \mathsf{ct}'')$ and the proof π to its own challenger. It remains to argue that this pair is fresh. Suppose it were not fresh, i.e. $\hat{B}_i$ queried $\textsc{Sim}(\mathsf{stmt})$ at some point and received π. This can only happen during a query of the form $\textsc{Kdm}(\mathsf{x}, f)$, in which $\hat{B}_i$ would have added $(\mathsf{x}, \mathsf{ct})$ to list $\mathcal{L}_{ct}$, a contradiction. Thus, we obtain

$$|\mathsf{pr}_3 - \mathsf{pr}_4| \leq \sum_{i=1}^{Q_D} \Pr[\mathsf{bad}_i] \leq \sum_{i=1}^{Q_D} \Pr\left[\mathbf{SIMSO}^{\hat{B}_i}_{\mathsf{PS}} \Rightarrow 1\right] \leq Q_D \cdot \varepsilon_{\mathsf{sso}}.$$

Game $\mathbf{G}_5$: In $\mathbf{G}_5$ we change the challenge ciphertexts again. This time, we change how ct' is generated. Namely, we generate it as $\mathsf{ct}' = \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, 0^{|f(\mathsf{msk})|}; \rho')$. Note that in $\mathbf{G}_4$ the only remaining direct dependencies on $\mathsf{msk} = \mathsf{msk}'$ are the ciphertexts ct' and the oracle $\textsc{Key}$. In particular, we do not need msk' to simulate the oracle $\textsc{Dec}$. Thus, a direct reduction $\mathcal{B}_3$ from the $\mathsf{mKDM\text{-}CPA}$ security of IBE' can be constructed and we obtain

$$|\mathsf{pr}_4 - \mathsf{pr}_5| \leq \mathsf{Adv}^{\mathsf{mKDM\text{-}CPA}}_{\mathcal{B}_3,\mathsf{ABE}'}(\lambda).$$

Games $\mathbf{G}_6$-$\mathbf{G}_8$: From $\mathbf{G}_6$ to $\mathbf{G}_8$ we revert changes that we did. To be precise, in $\mathbf{G}_6$ we use msk' again to simulate decryption queries, which can be analyzed in a similar way to the step from $\mathbf{G}_3$ to $\mathbf{G}_4$. In $\mathbf{G}_7$ we generate the proofs π in challenge queries honestly again. In $\mathbf{G}_8$ we generate crs using PGen again. Note that all previously used arguments apply and we have

$$|\mathsf{pr}_5 - \mathsf{pr}_8| \leq Q_D \cdot \varepsilon_{\mathsf{sso}} + Q_C \cdot \varepsilon_{\mathsf{zk}} + \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}''_1,\mathsf{PS}}(\lambda),$$

for some reduction $\mathcal{B}''_1$. Further, $\mathbf{G}_8$ is equivalent to game $\mathbf{mKDM\text{-}CCA}_{0,\mathsf{ABE}}$. Thus, setting $\mathcal{B}_1$ to be $\arg\max_{\mathcal{B} \in \{\mathcal{B}'_1, \mathcal{B}''_1\}} \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B},\mathsf{PS}}(\lambda)$ we obtain the result. □

4.6 Instantiation and Extension

Here, we want to reference to example instantiations for the building blocks of our construction in the standard model. We highlight that this is only a prototypical proof-of-concept instantiation, and due to the use of obfuscation, a practical instantiation is out of scope. First of all, for iO, we can use the work by Jai, Lin and Sahai [JLS21] relying on subexponential hardness of the assumptions $\mathsf{LWE}, \mathsf{LPN}, \mathsf{SXDH}$ and PRG_0, where PRG_0 stands for a pseudorandom generator that can be evaluated in constant depth. We can use any perfectly complete attribute-based encryption scheme. If PKE is needed (cf. Section 4.4), we can use ElGamal encryption [ElG84]. We can instantiate the proof system PS using GOS proofs [GOS12]. For the simulation-sound proof system PS' we can use the system by Groth [Gro06]. Both proof systems are based on the DLIN assumption and can be used for any **NP** relation using Karp reductions.

Game $\mathbf{G}_0$-$\mathbf{G}_5$

$(\mathsf{mpk}', \mathsf{msk}') \leftarrow \mathsf{Setup}'(1^\lambda)$
$(\mathsf{pk}'', \mathsf{sk}'') \leftarrow \mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)$
$\mathsf{crs} \leftarrow \mathsf{PGen}(1^\lambda)$ // $\mathbf{G}_0$
$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{PTrapGen}(1^\lambda)$ // $\mathbf{G}_1$-$\mathbf{G}_5$
$\mathsf{mpk} := (\mathsf{mpk}', \mathsf{pk}'', \mathsf{crs})$
$\mathrm{O} := (\textsc{Key}, \textsc{Kdm}, \textsc{Dec})$
$b' \leftarrow \mathcal{A}^{\mathrm{O}}(\mathsf{mpk})$
return b'

Oracle $\textsc{Dec}(\mathsf{y}, \mathsf{ct} = (\mathsf{x}, \mathsf{ct}', \mathsf{ct}'', \pi))$

if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 1 \wedge (\mathsf{x}, \mathsf{ct}) \in \mathcal{L}_{ct}$:
  return $\perp$
$\mathsf{stmt} := (\mathsf{mpk}', \mathsf{pk}'', \mathsf{x}, \mathsf{ct}', \mathsf{ct}'')$
if $\mathsf{PVer}(\mathsf{crs}, \mathsf{stmt}, \pi) = 0$: **return** $\perp$
if $\mathcal{P}(\mathsf{x}, \mathsf{y}) = 0$: **return** $\perp$
$\mathsf{sk}_\mathsf{y} \leftarrow \mathsf{KeyExt}'(\mathsf{msk}', \mathsf{y})$ // $\mathbf{G}_0$-$\mathbf{G}_3$
$\mathsf{m} := \mathsf{Dec}'(\mathsf{sk}_\mathsf{y}, \mathsf{ct}')$ // $\mathbf{G}_0$-$\mathbf{G}_3$
$\mathsf{m} := \mathsf{Dec}_{\mathsf{PKE}}(\mathsf{sk}'', \mathsf{ct}'')$ // $\mathbf{G}_4, \mathbf{G}_5$
return m

Oracle $\textsc{Kdm}(\mathsf{x}, f \in \mathcal{F})$

if $\mathsf{hit}_{\mathcal{P}}(\{\mathsf{x}\}, \mathcal{L}_{sk})$: **return** $\perp$
$\mathcal{L}_{ch} := \mathcal{L}_{ch} \cup \{\mathsf{x}\}$, $\rho', \rho'' \xleftarrow{\$} \{0,1\}^z$, $\mathsf{m}_0 := 0^{|f(\cdot)|}, \mathsf{m}_1 := f(\mathsf{msk})$
$\mathsf{ct}' := \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, \mathsf{m}_1; \rho')$ // $\mathbf{G}_0$-$\mathbf{G}_4$
$\mathsf{ct}' := \mathsf{Enc}'(\mathsf{mpk}', \mathsf{x}, \mathsf{m}_0; \rho')$ // $\mathbf{G}_5$
$\mathsf{ct}'' := \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}'', \mathsf{m}_1; \rho'')$ // $\mathbf{G}_0$-$\mathbf{G}_2$
$\mathsf{ct}'' := \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}'', \mathsf{m}_0; \rho'')$ // $\mathbf{G}_3$-$\mathbf{G}_5$
$\mathsf{stmt} := (\mathsf{mpk}', \mathsf{pk}'', \mathsf{x}, \mathsf{ct}', \mathsf{ct}'')$
$\mathsf{witn} := (\mathsf{m}_1, \rho', \rho'')$, $\pi \leftarrow \mathsf{PProve}(\mathsf{crs}, \mathsf{stmt}, \mathsf{witn})$ // $\mathbf{G}_0, \mathbf{G}_1$
$\pi \leftarrow \mathsf{PSim}(\mathsf{crs}, \mathsf{td}, \mathsf{stmt})$ // $\mathbf{G}_2$-$\mathbf{G}_5$
$\mathsf{ct} := (\mathsf{x}, \mathsf{ct}', \mathsf{ct}'', \pi)$, $\mathcal{L}_{ct} := \mathcal{L}_{ct} \cup \{(\mathsf{x}, \mathsf{ct})\}$
return ct

Figure 26: The games $\mathbf{G}_0$-$\mathbf{G}_5$ in the proof of Theorem 4.13. Lines with highlighted comments are only executed in the corresponding games. Oracle $\textsc{Key}(\mathsf{y})$ is as in the real game (Figure 4).

Remark 4.14 (Imperfect Completeness). As written, the construction in Section 4.2 only works for identity-based and public key encryption schemes with perfect completeness. This is due to the use of indistinguishability obfuscation, as this primitive only guarantees security for perfectly functionally equivalent circuits. However, we note that one can still adopt most of the constructions in a lattice-based setting. To see that, note that in lattice-based (identity-based) encryption schemes based on dual-style Regev encryption, such as [GPV08, CHKP10], the completeness error results from two potentially long vectors that influence the decryption process: First, a user secret key corresponds to a Gaussian SIS solution, which can have a large norm with non-zero probability. Second, the ciphertext contains Gaussian errors which can be to long as well. To solve this problem, the key extraction and encryption algorithms can just abort if these vectors are to long. As this happens with negligible probability and can be done outside of any obfuscation, we can still allow such an abort in our construction. To be more precise, it is important that the modified key extraction and encryption algorithms abort with negligible probability and that if they do not abort, then decryption always succeeds. Then, to make our proof work, we change the original algorithms to the aborting ones, then we apply the obfuscation transition. Afterwards, we go back to the original algorithms that guarantee security, and follow the rest of the proof. It remains to instantiate the proof system and the public key encryption scheme with unique secret keys. We focus on the latter in Section 5. A different approach to solve the imperfect completeness issue is the error-removing transformation by Bitanski and Vaikuntanathan [BV17].

Remark 4.15 (Identity-Based Encryption). As a special case with the identity predicate (which is compatible with itself), the constructions in Sections 4.2 and 4.4 directly imply similar constructions for identity-based encryption.

Remark 4.16 (Attribute-Hiding). In the context of attribute-based encryption, one may additionally aim to achieve attribute-hiding, which means that a ciphertext generated for attribute x does not reveal x. We note that extending our results to this setting requires additional techniques. This is because in all steps of our proof, x is hardcoded in the challenge ciphertext circuit.

5 Lattice Public Key Encryption with Unique Secret Keys

In this section we modify the well-known Regev encryption scheme [Reg05] such that whenever a public key is generated and the generation algorithm does not abort, there exists only one valid secret key for it. Note that with overwhelming probability the key of the original Regev scheme is already unique. However, even during key generation, it is not straight-forward to check if this holds, because the uniqueness depends on the length of a shortest vector in the lattice given by the public key. Thus, if we want to use the Regev encryption scheme as a building block in our construction in Section 4.4 using the aborting technique as discussed in Remark 4.14 we need to apply some minor modifications to the key generation algorithm, such that a lower bound on this shortest vector is known.

Before we go into detail, we need to recall some lattice background. For notation, the Euclidean norm of a vector $\mathbf{v}$ is denoted by $\|\mathbf{v}\|$. Let $q \in \mathbb{P}$ be a prime and Λ be an m-dimensional lattice. That is, a discrete additive subgroup of $\mathbb{R}^m$. Any such lattice is of the form $\Lambda = \mathbf{B} \cdot \mathbb{Z}^k$ for some $\mathbf{B} \in \mathbb{Z}^{m \times k}$ with linearly independent columns, where $k \leq m$ is the rank of the lattice. We denote its dual lattice by Λ^*, which is defined as

$$\Lambda^* := \{\mathbf{x} \in \mathbb{R}^m : \forall \mathbf{y} \in \Lambda : \mathbf{x}^t \mathbf{y} \in \mathbb{Z}\}.$$

Also, for $1 \leq i \leq n$ we denote its i-th successive minimum by $\lambda_i(\Lambda)$, which is the smallest $B \in \mathbb{R}$ such that there are i linearly independent vectors of length at most B in Λ. For any vector $\mathbf{c} \in \mathbb{R}^m$ we denote the discrete Gaussian distribution with parameter $s > 0$ over the coset $\mathbf{c} + \Lambda$ by $D_{\mathbf{c}+\Lambda,s}$. Any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}, m > n$ defines m-dimensional lattices and lattice cosets:

$$\begin{aligned}
\Lambda_q(\mathbf{A}) &:= \{\mathbf{A}^t \mathbf{s} : \mathbf{s} \in \mathbb{Z}^n\} + q\mathbb{Z}^m, \\
\Lambda_q^{\perp}(\mathbf{A}) &:= \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = \mathbf{0} \mod q\}, \\
\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}) &:= \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = \mathbf{u} \mod q\}.
\end{aligned}$$

These lattices are dual up to scaling by q:

$$\Lambda_q(\mathbf{A}) = q\Lambda_q^{\perp}(\mathbf{A})^*, \ \Lambda_q(\mathbf{A})^* = \frac{1}{q}\Lambda_q^{\perp}(\mathbf{A}).$$

We also need some tail bounds for discrete Gaussians, see Lemmas 5.1, 5.2 and 5.3 in [GPV07] and Lemma 4.4 in [MR04].

Lemma 5.1 ([GPV07]). *Let $n, m \in \mathbb{N}$, $q \in \mathbb{P}$ at least polynomial in n, $m \geq 2n \log q$. Consider any $\omega(\sqrt{\log m})$ function and $s \geq \omega(\sqrt{\log m})$. Then for all but a negligible (in n) fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ the following distribution is statistically close to uniform over $\mathbb{Z}_q^n$: $\{\mathbf{A}\mathbf{e} \mid \mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}\}$. Furthermore, the conditional distribution of $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$ given $\mathbf{u} = \mathbf{A}\mathbf{e} \mod q$ is exactly $D_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}),s}$.*

Lemma 5.2 ([MR04, GPV07]). *Consider any $\omega(\sqrt{\log m})$ function and $s \geq \omega(\sqrt{\log m})$. Then we have*

$$\Pr\left[\|\mathbf{x}\| > s\sqrt{m} \mid \mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}\right] \leq 2^{-m+1}.$$

Lemma 5.3 ([MR04, GPV07]). *Let $n \in \mathbb{N}$, $q \in \mathbb{P}$ and $m \geq 2n \log q$. Consider any $\omega(\sqrt{\log m})$ function and $s \geq \omega(\sqrt{\log m})$. Then for all but an at most q^{-n} fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and any vector $\mathbf{u} \in \mathbb{Z}_q^n$, we have*

$$\Pr\left[\|\mathbf{x}\| > s\sqrt{m} \mid \mathbf{x} \leftarrow D_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}),s}\right] \leq 2^{-m+1}.$$

Lemma 5.4 ([Ajt96, Reg05, GPV07, GPV08]). *Let $n \in \mathbb{N}$, $q \in \mathbb{P}$ and $m \geq 2n \log q$. The for all but an at most q^{-n} fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the subset sums of the columns of $\mathbf{A}$ generate $\mathbb{Z}_q^n$, i.e. for every $\mathbf{u} \in \mathbb{Z}_q^n$ there is an $\mathbf{e} \in \{0,1\}^m$ with $\mathbf{A}\mathbf{e} = \mathbf{u} \mod q$.*

For our modification of the Regev encryption scheme we need a result by Ajtai [Ajt99]. We note that [GPV08] claims a more efficient bound $L = m^{1+\epsilon}$ for any $\epsilon > 0$, which would also result in a more efficient instantiation of our parameters. As [GPV08] give no details, we use the bound given in [Ajt99].

Lemma 5.5 ([Ajt99, GPV08]). *Let $n = \Theta(\lambda)$. For any prime $q \in \mathbb{P}$ polynomial in n, any $m \geq 5n \log q$, there is an $L = m^{3.5}$ and a PPT algorithm* $\mathsf{GenWithBasis}$ *such that* $\mathsf{GenWithBasis}(1^n, 1^m, q)$ *takes as input n, m and q and outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{S} \subset \Lambda_q^{\perp}(\mathbf{A})$. For these outputs it holds that $\mathbf{A}$ is distributed statistically close to uniform over $\mathbb{Z}_q^{n \times m}$ and all vectors in $\mathbf{S}$ have length at most L.*

For the proof of uniqueness we also need the following lemma and show a corollary.

Lemma 5.6 (Transference Theorem [Ban93]). *For any lattice Λ of rank m it holds that $1 \le \lambda_1(\Lambda) \cdot \lambda_m(\Lambda^*) \le m$.*

Corollary 5.7 *Let $n = \Theta(\lambda)$, $q \in \mathbb{P}$ polynomial in n, $m \ge 5n \log q, L = m^{3.5}$ and* $\mathsf{GenWithBasis}$ *be as in Lemma 5.5. Let $(\mathbf{A}, \mathbf{S}) \leftarrow \mathsf{GenWithBasis}(1^n, 1^m, q)$. Define the map*

$$g_{\mathbf{A}} : \mathbb{Z}_q^n \times \{\mathbf{e} \in \mathbb{Z}^m \mid \|\mathbf{e}\| \le B\} \longrightarrow \mathbb{Z}_q^m$$
$$(\mathbf{s}, \mathbf{e}) \longmapsto \mathbf{A}^t\mathbf{s} + \mathbf{e}.$$

If $\mathbf{A}$ is full-rank and $2 \cdot L \cdot B < q$, then $g_{\mathbf{A}}$ is injective.

Proof. Let $\mathcal{D} := \mathbb{Z}_q^n \times \{\mathbf{e} \in \mathbb{Z}^m \mid \|\mathbf{e}\| \le B\}$ and assume $g_{\mathbf{A}}(\mathbf{s}_1, \mathbf{e}_1) = g_{\mathbf{A}}(\mathbf{s}_2, \mathbf{e}_2)$. This implies that

$$\mathbf{e}_1 - \mathbf{e}_2 = \mathbf{A}^t(\mathbf{s}_2 - \mathbf{s}_1).$$

Thus, $\bar{\mathbf{e}} := \mathbf{e}_1 - \mathbf{e}_2$ (reduced modulo q) is a lattice vector in $\Lambda_q(\mathbf{A})$. In particular, by the triangle inequality we have $\|\bar{\mathbf{e}}\| \le 2 \cdot B$. Next, we want to lower bound $\lambda_1(\Lambda_q(\mathbf{A}))$. To do so, recall that by Lemma 5.5 we have $\lambda_m(\Lambda_q^{\perp}(\mathbf{A})) \le L$. Using the Transference Theorem (Lemma 5.6) and the duality of $\Lambda_q(\mathbf{A})$ and $\Lambda_q^{\perp}(\mathbf{A})$ (up to scaling) we obtain

$$\lambda_1(\Lambda_q(\mathbf{A})) \ge \frac{1}{\lambda_m(\Lambda_q(\mathbf{A})^*)} \ge \frac{q}{\lambda_m(\Lambda_q^{\perp}(\mathbf{A}))} \ge \frac{q}{L}.$$

Thus, using the assumption $2 \cdot L \cdot B < q$ we see that $\bar{\mathbf{e}}$ is shorter than $\lambda_1(\Lambda_q(\mathbf{A}))$, implying $\mathbf{e}_1 = \mathbf{e}_2$. Finally, this also implies $\mathbf{A}^t\mathbf{s}_1 = \mathbf{A}^t\mathbf{s}_2$. As $\mathbf{A}$ is full-rank, we have $\mathbf{s}_1 = \mathbf{s}_2$. ☐

We define the LWE assumption.

Definition 5.8 (Learning With Errors Assumption (LWE)). Let $\lambda, n = n(\lambda) \in \mathbb{N}, q = q(n)$ be prime number and $\chi = \chi(n)$ be a distribution over $\mathbb{Z}$. We say that the $\mathsf{LWE}_{n,q,\chi}$ assumption holds, if for every PPT algorithm $\mathcal{B}$ and every polynomial $m = \mathsf{poly}(n)$ the following advantage is negligible in λ:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}_{n,q,\chi}}(\lambda) := |&\Pr\left[\mathcal{B}(\mathbf{A}, \mathbf{b}) = 1 \mid \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{b} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}\right] \\ &-\Pr\left[\mathcal{B}(\mathbf{A}, \mathbf{A}^t\mathbf{s} + \mathbf{e}) = 1 \mid \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \leftarrow \chi^m\right]|.\end{aligned}$$

A sequence of works [Reg05, Pei09, BLP+13] shows the hardness of LWE for discrete Gaussian error distributions of parameter αq with $\alpha q \ge 2\sqrt{n}$ based on the worst-case hardness of lattice approximation problems.

We define our modified Regev encryption in Figure 27 with message space $\mathcal{M} = \{0, 1\}$. Note that it is basically the Regev encryption scheme, but the matrix $\mathbf{A}$ is generated using Lemma 5.5 and some aborts are added. It makes use of LWE parameters $n, q, m \ge 5n \log q$ and $\alpha > 0$ with $\alpha q \ge 2\sqrt{n}$ and $q \in \mathbb{P}$. Then we have $\alpha q \ge \omega(\sqrt{\log m})$, meaning that we can apply Lemmata 5.1 and 5.3. For completeness we need $4\alpha^2 mq < 1$ and for uniqueness of secret keys $2\alpha m^4 < 1$. To be concrete, fixing $m := 5n \log q$ and $\alpha q := 2\sqrt{m}$ an easy calculation shows that any $q > \max\{16m^2, 4m^{4.5}\}$ is sufficient. We will now show that if neither Gen nor Enc do abort, then we always have correct decryption. Further, Gen and Enc only abort with negligible probability.

Lemma 5.9 (Completeness). *If $4\alpha^2 mq < 1$, then for* $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ *as defined in Figure 27 the following hold:*

- $\mathsf{Gen}(1^\lambda)$ *aborts with negligible probability.*
- *For all* $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda)$ *and any* $\mathsf{m} \in \mathcal{M}$ *the algorithm* $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ *aborts with negligible probability.*
- *For all* $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda)$, *any* $\mathsf{m} \in \mathcal{M}$ *and any* $\mathsf{ct} \in \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ *we have* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{m}$.

Alg $\mathsf{Gen}(1^\lambda)$

set parameters as in the text.
$(\mathbf{A}, \mathbf{S}) \leftarrow \mathsf{GenWithBasis}(1^n, 1^m, q)$
if $\mathbf{A}$ not full-rank: **return** $\perp$
$\mathbf{e} \leftarrow D^m_{\mathbb{Z},\alpha q}$
if $\|\mathbf{e}\| > \alpha q\sqrt{m}$: **return** $\perp$
$\mathbf{s} \xleftarrow{\$} \mathbb{Z}^n_q, \mathbf{b} := \mathbf{A}^t\mathbf{s} + \mathbf{e} \in \mathbb{Z}^m_q$
$\mathsf{pk} := \bar{\mathbf{A}} := \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^t \end{bmatrix} \in \mathbb{Z}_q^{(n+1)\times m}$
return $(\mathsf{pk}, \mathsf{sk} := \mathbf{s})$

Alg $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$

if $|[-\mathbf{s}^t|1]\mathsf{ct}| > q/2$: **return** 1
return 0

Alg $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$

$\mathbf{x} \leftarrow D^m_{\mathbb{Z},\alpha q}$
if $\|\mathbf{x}\| > \alpha q\sqrt{m}$: **return** $\perp$
return $\mathsf{ct} := \bar{\mathbf{A}}\mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathsf{m}\lfloor q/2 \rceil \end{bmatrix}$

Alg $\mathsf{VerK}_{\mathsf{PKE}}(\bar{\mathbf{A}}, \mathbf{s})$

let $\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^t \end{bmatrix} \in \mathbb{Z}_q^{(n+1)\times m}$
$\mathbf{e} := \mathbf{b} - \mathbf{A}^t\mathbf{s}$
if $\|\mathbf{e}\| > \alpha q\sqrt{m}$: **return** 0
return 1

Figure 27: The public key encryption scheme $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ and the associated key verification algorithm $\mathsf{VerK}_{\mathsf{PKE}}$. The scheme is a modification of the classical Regev encryption scheme [Reg05] such that unique keys are guaranteed.

Proof. For the first claim, note that Gen only aborts if $\mathbf{A}$ is not full-rank or $\|\mathbf{e}\| > \alpha q\sqrt{m}$. Note that by Lemma 5.5 the matrix $\mathbf{A}$ is statistically close to uniform. Thus, the former happens with negligible probability, by Lemma 5.4, and the latter happens with negligible probability by Lemma 5.2. Similarly, the second claim follows directly from Lemma 5.2. For the third claim, note that by the Cauchy-Schwarz inequality

$$|[-\mathbf{s}^t|1]\mathsf{ct} - \mathsf{m}\lfloor q/2 \rceil| = |\mathbf{e}^t\mathbf{x}| \le \|\mathbf{e}\|\|\mathbf{x}\| \le \alpha^2 q^2 m,$$

where the last inequality is always true if neither $\mathsf{Gen}(1^\lambda)$ nor $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ aborts. Finally, the assumption $4\alpha^2 mq < 1$ implies that this term is less than $q/4$, which finishes the proof. $\square$

Lemma 5.10 (Uniquely Verifiable Secret Keys). *If* $2\alpha m^4 < 1$*, then for* $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ *and* $\mathsf{VerK}_{\mathsf{PKE}}$ *as defined in Figure 27 the following holds:*

- *For all* $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda)$ *we have* $\mathsf{VerK}_{\mathsf{PKE}}(\bar{\mathbf{A}}, \mathbf{s}) = 1$.
- *For all* $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda)$ *and any* sk' *with* $\mathsf{VerK}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}') = 1$ *we have* $\mathsf{sk} = \mathsf{sk}'$.

Proof. The first claim is clear by the definition of algorithms $\mathsf{Gen}, \mathsf{VerK}_{\mathsf{PKE}}$. For the second claim, let $\mathsf{sk} = \mathbf{s}, \mathsf{sk}' = \mathbf{s}'$ and define

$$\mathbf{e} := \mathbf{b} - \mathbf{A}^t\mathbf{s},\ \mathbf{e}' := \mathbf{b} - \mathbf{A}^t\mathbf{s}'.$$

By assumption, matrix $\mathbf{A}$ is generated using $\mathsf{GenWithBasis}(1^n, 1^m, q)$. As Gen did not abort, we know that $\mathbf{A}$ is full-rank. Next, set $B := \alpha q\sqrt{m}$. Then $2\alpha m^4 < 1$ implies that all the conditions of Corollary 5.7 are satisfied. If $\mathsf{VerK}_{\mathsf{PKE}}$ accepts both sk and sk' then

$$(\mathbf{s}, \mathbf{e}), (\mathbf{s}', \mathbf{e}') \in \mathbb{Z}^n_q \times \{\mathbf{x} \in \mathbb{Z}^m \mid \|\mathbf{x}\| \le B\}.$$

Finally, with notation as in Corollary 5.7 and by definition of $\mathbf{e}, \mathbf{e}'$ we have

$$g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{b} = g_{\mathbf{A}}(\mathbf{s}', \mathbf{e}').$$

As $g_{\mathbf{A}}$ is injective, the statement follows. $\square$

For completeness of our presentation, we also sketch $\mathsf{IND\text{-}CPA}$ security, although it is nearly the same as the standard proof for the original Regev scheme.

Lemma 5.11 (Security). *The scheme* $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$ *as defined in Figure 27 is* $\mathsf{IND\text{-}CPA}$ *secure under the* $\mathsf{LWE}_{n,q,D_{\mathbb{Z},\alpha q}}$ *assumption. In particular, for every PPT algorithm* $\mathcal{A}$ *there is a PPT algorithm* $\mathcal{B}$ *such that* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \le 2 \cdot \mathsf{Adv}^{\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\alpha q}}}_{\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda).$$

Proof. We give a short proof sketch using a sequence of games. For each game $\mathbf{G}_i$, we denote the probability that it outputs 1 by pr_i, namely,

$$\mathsf{pr}_i := \Pr\left[\mathbf{G}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right].$$

Game $\mathbf{G}_0$ is the original $\mathbf{IND\text{-}CPA}_0$ game. In $\mathbf{G}_1$, we generate $\mathbf{A}$ uniformly random instead of using $\mathsf{GenWithBasis}$ and remove all aborts whenever the game uses algorithms $\mathsf{Gen}(1^\lambda)$ and $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)$. As $\mathbf{A}$ is statistically close to uniform in $\mathbf{G}_0$ by Lemma 5.5 and Lemma 5.9 states that the aborts only happen with negligible probability we have

$$|\mathsf{pr}_0 - \mathsf{pr}_1| \leq \mathsf{negl}(\lambda).$$

Now we are in the setting of the classical Regev proof. Thus, in game $\mathbf{G}_2$ we change the last row of the public key to random:

$$\mathbf{b} \xleftarrow{\$} \mathbb{Z}_q^m.$$

A straight-forward reduction $\mathcal{B}$ shows that this change is not noticed by the adversary, under the LWE assumption:

$$|\mathsf{pr}_1 - \mathsf{pr}_2| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\alpha q}}}(\lambda).$$

We note that now the matrix $\mathsf{pk} = \bar{\mathbf{A}}$ is uniformly random, thus Lemma 5.1 implies that the ciphertext is statistically close to uniformly random over $\mathbb{Z}_q^{n+1}$. In game $\mathbf{G}_3$ we generate the challenge ciphertext $\mathsf{ct} \xleftarrow{\$} \mathbb{Z}_q^{n+1}$ and we have

$$|\mathsf{pr}_2 - \mathsf{pr}_3| \leq \mathsf{negl}(\lambda).$$

We repeat all steps in reverse order to end up at the game $\mathbf{IND\text{-}CPA}_1$, which finishes the proof. $\square$

Acknowledgments Our previous draft was presented in the context of identity-based encryption and observed that it can be extended to attribute-based encryption (ABE). We thank an anonymous reviewer from Asiacrypt 2022 for suggesting us to present our paper with ABE. We also thank the anonymous reviewers from Designs, Codes and Cryptography for carefully reading our paper and pointing out some notation inconsistency.

References

[AHY15] Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. A framework for identity-based encryption with almost tight security. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 521–549. Springer, Heidelberg, November / December 2015.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *ICALP 99*, volume 1644 of *LNCS*, pages 1–9. Springer, Heidelberg, July 1999.

[AP12] Jacob Alperin-Sheriff and Chris Peikert. Circular and KDM security for identity-based encryption. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 334–352. Springer, Heidelberg, May 2012.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.

[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, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[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 ACM STOC*, pages 575–584. ACM Press, June 2013.

[BRS03] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, Heidelberg, August 2003.

[BV17] Nir Bitansky and Vinod Vaikuntanathan. A note on perfect correctness by derandomization. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 592–606. Springer, Heidelberg, April / May 2017.

[CCS09] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer, Heidelberg, April 2009.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, Heidelberg, May / June 2010.

[CZDC16] Yu Chen, Jiang Zhang, Yi Deng, and Jinyong Chang. KDM security for identity-based encryption: Constructions and separations. Cryptology ePrint Archive, Report 2016/1020, 2016. `https://eprint.iacr.org/2016/1020`.

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Heidelberg, August 1984.

[FGC21] Shengyuan Feng, Junqing Gong, and Jie Chen. Master-key KDM-secure ABE via predicate encoding. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 543–572. Springer, Heidelberg, May 2021.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In Hideki Imai and Yuliang Zheng, editors, *PKC'99*, volume 1560 of *LNCS*, pages 53–68. Springer, Heidelberg, March 1999.

[GDCC16] Junqing Gong, Xiaolei Dong, Jie Chen, and Zhenfu Cao. Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 624–654. Springer, Heidelberg, December 2016.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGH20] Sanjam Garg, Romain Gay, and Mohammad Hajiabadi. Master-key KDM-secure IBE from pairings. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 123–152. Springer, Heidelberg, May 2020.

[GHKP18] Romain Gay, Dennis Hofheinz, Lisa Kohl, and Jiaxin Pan. More efficient (almost) tightly secure structure-preserving signatures. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 230–258. Springer, Heidelberg, April / May 2018.

[GHV12] David Galindo, Javier Herranz, and Jorge L. Villar. Identity-based encryption with master key-dependent message security and leakage-resilience. In Sara Foresti, Moti Yung, and Fabio Martinelli, editors, *ESORICS 2012*, volume 7459 of *LNCS*, pages 627–642. Springer, Heidelberg, September 2012.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *Journal of the ACM (JACM)*, 59(3):1–35, 2012.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC'21*, pages 736–749. ACM, 2021.

[GPV07] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. Cryptology ePrint Archive, Report 2007/432, 2007. `https://eprint.iacr.org/2007/432`.

[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 ACM STOC*, pages 197–206. ACM Press, May 2008.

[Gro06] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Heidelberg, December 2006.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

[HKS15] Dennis Hofheinz, Jessica Koch, and Christoph Striecks. Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 799–822. Springer, Heidelberg, March / April 2015.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC'21*, pages 60–73. ACM, 2021.

[KMHT16] Fuyuki Kitagawa, Takahiro Matsuda, Goichiro Hanaoka, and Keisuke Tanaka. On the key dependent message security of the Fujisaki-Okamoto constructions. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 99–129. Springer, Heidelberg, March 2016.

[KT18] Fuyuki Kitagawa and Keisuke Tanaka. Key dependent message security and receiver selective opening security for identity-based encryption. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 32–61. Springer, Heidelberg, March 2018.

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM CCS 2003*, pages 155–164. ACM Press, October 2003.

[KYY18] Shuichi Katsumata, Shota Yamada, and Takashi Yamakawa. Tighter security proofs for GPV-IBE in the quantum random oracle model. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 253–282. Springer, Heidelberg, December 2018.

[LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, Heidelberg, February 2010.

[MPs16] Antonio Marcedone, Rafael Pass, and abhi shelat. Bounded KDM security from iO and OWF. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 571–586. Springer, Heidelberg, August / September 2016.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, October 2004.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Tsa19] Rotem Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 62–85. Springer, Heidelberg, August 2019.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, August 2009.