Key Dependent Message Security and Receiver Selective Opening Security for Identity-Based Encryption

Fuyuki Kitagawa and Keisuke Tanaka

Tokyo Institute of Technology, Tokyo, Japan
{kitagaw1,keisuke}@is.titech.ac.jp

Abstract

We construct two identity-based encryption (IBE) schemes. The first one is IBE satisfying key dependent message (KDM) security for user secret keys. The second one is IBE satisfying simulation-based receiver selective opening (RSO) security. Both schemes are secure against adaptive-ID attacks and do not have any a-priori bound on the number of challenge identities queried by adversaries in the security games. They are the first constructions of IBE satisfying such levels of security.

Our constructions of IBE are very simple. We construct our KDM secure IBE by transforming KDM secure secret-key encryption using IBE satisfying only ordinary indistinguishability against adaptive-ID attacks (IND-ID-CPA security). Our simulation-based RSO secure IBE is based only on IND-ID-CPA secure IBE.

We also demonstrate that our construction technique for KDM secure IBE is used to construct KDM secure public-key encryption. More precisely, we show how to construct KDM secure public-key encryption from KDM secure secret-key encryption and public-key encryption satisfying only ordinary indistinguishability against chosen plaintext attacks.

Keywords: Identity-based encryption, Key dependent message security, Receiver selective opening security.

Contents

1 Introduction

1.1 Background

Identity-based encryption (IBE) proposed by Shamir [Sha84] is an extension of public-key encryption (PKE). In IBE, we can use an identity of a recipient as a public-key. The secret key corresponding to an identity is generated only by the trusted authority who has the master secret key. Users can obtain secret keys corresponding to their identities by authenticating themselves to the trusted authority. By using IBE, we can avoid the need to distribute public-key certificates that is one of the major issues with public-key cryptography.

Security notions for IBE capture corruptions and collusions of users. In other words, we require that IBE guarantee confidentiality of a message encrypted under an identity id^* even if an adversary obtains secret keys corresponding to any identity other than id^*.

Security notions for IBE are classified into two categories, that is, adaptive security and selective security. an IBE scheme is said to be secure against adaptive-ID attacks [BF01] if it is secure even when an adversary adaptively chooses the challenge identity id^*. On the other hand, an IBE scheme is said to be secure against selective-ID attacks [CHK03] if it is secure when an adversary declares the challenge identity id^* before seeing the public parameter.

Security against adaptive-ID attacks is a desirable security notion for IBE when we use it in practical situations. However, since IBE has an advanced functionality compared to PKE, attack scenarios that ordinary indistinguishability against adaptive-ID attacks does not capture can naturally occur in practical situations of IBE. As such attack scenarios, in this work, we focus on the situation of encrypting secret keys and the selective opening attacks.

Black, Rogaway, and Shrimpton [BRS03] introduced the notion of *key dependent message (KDM) security* which guarantees confidentiality even in the situation of encrypting secret keys. Informally, an encryption scheme is said to be KDM secure if it is secure when an adversary can obtain encryptions of $f(\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell)$, where $\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell$ are secret keys that exist in the system and f is a function.

Alperin-Sheriff and Peikert [AP12] pointed out that KDM security with respect to user secret keys is well-motivated by some natural usage scenarios for IBE such as key distribution in a revocation system. They constructed the first IBE satisfying KDM security for user secret keys assuming the hardness of the learning with errors (LWE) problem. Galindo, Herranz, and Villar [GHV12] proposed an IBE scheme that satisfies KDM security for master secret keys based on the hardness of a rank problem on bilinear groups. However, both of these schemes are secure only against selective-ID attacks. Moreover, both schemes have some a-priori bound on the number of queries made by an adversary.[1]

In the selective opening attack, an adversary, given some ciphertexts, adaptively corrupts some fraction of users and try to break confidentiality of ciphertexts of uncorrupted users.

There are both sender corruption case and receiver corruption case in this attack scenario. Bellare, Hofheinz, and Yilek [BHY09] formalized *sender selective opening (SSO) security* for PKE that captures situations where there are many senders and a single receiver, and an adversary can obtain messages and random coins of corrupted senders. Hazay, Patra, and Warinschi [HPW15] later formalized *receiver selective opening (RSO) security* for PKE that captures situations where there are many receivers and a single sender, and an adversary can obtain messages and secret keys of corrupted receivers.

Selective opening attacks originally considered in the context of multi-party computation is natural and motivated in the context of IBE since it also considers situations where there are

[1] The scheme by Alperin-Sheriff and Peikert has an a-priori bound on the number of challenge identities in the security game. The scheme by Galindo et al. has an a-priori bound of the number of KDM encryption queries made by an adversary.

many users and some fraction are corrupted. Bellare, Waters, and Yilek [BWY11] defined SSO security for IBE and proposed SSO secure IBE schemes under the decisional linear assumption and a subgroup decision assumption in composite order bilinear groups. Their definition of SSO security for IBE captures adaptive-ID attacks in addition to sender selective opening attacks. However, it does not take receiver selective opening attacks into account.

It is known that the standard notions of indistinguishability implies neither KDM security [ABBC10, CGH12, BHW15, KW16] nor selective opening security [BDWY12, HR14, HRW16]. From this fact, we know very little about the possibility of IBE satisfying these stronger security notions than standard indistinguishability though there have been many works on the study of IBE.

Especially, it is open whether we can construct IBE that is KDM secure against adaptive-ID attacks and there is no a-priori bound on the number of queries made by an adversary. For selective opening security, we have no construction of IBE satisfying RSO security even if we require only security against selective-ID attacks.

As mentioned above, attack scenarios captured by both KDM security and selective opening security are natural and motivated for IBE. We thus think it is important to clarify these issues.

1.2 Our Results

Based on the above background, we propose KDM secure IBE and RSO secure IBE. Both schemes satisfy security against adaptive-ID attacks. They are the first schemes satisfying such levels of security.

Our constructions of IBE are very simple. We construct KDM secure IBE by transforming KDM secure secret-key encryption (SKE) using IBE satisfying ordinary indistinguishability against adaptive-ID attacks (IND-ID-CPA security) and garbled circuits. Somewhat surprisingly, Our RSO secure IBE is based only on IND-ID-CPA secure IBE. We think they shares the same sprit of construction strategy.

We show the details of each result below.

Key dependent message secure IBE. In this work, we focus on KDM security for user secret keys similarly to Alperin-Sheriff and Peikert [AP12], and let KDM security indicate KDM security for user secret keys. We show the following theorem.

Theorem 1 (Informal) *Assuming there exist IND-ID-CPA secure IBE and SKE that is KDM secure with respect to projection functions (resp. functions computable by a-priori bounded size circuits). Then, there exists IBE that is KDM secure with respect to projection functions (resp. functions computable by a-priori bounded size circuits) against adaptive-ID attacks.*

Projection function is a function whose each output bit depends on at most one bit of an input. KDM security with respect to projection functions is a generalization of circular security [CL01]. We can construct IBE satisfying KDM security with respect to any function computable by circuits of a-priori bounded size [BHHI10] by requiring the same KDM security for the underlying SKE.

As noted above, KDM secure IBE proposed by Alperin-Sheriff and Peikert is only secure against selective-ID attacks. Moreover, their scheme has an a-priori bound on the number of challenge identities in the security game. Our KDM secure IBE is secure against adaptive-ID attacks and does not have any a-priori bound on the number of queries made by an adversary in the security game.

To achieve KDM security for a-priori unbounded number of challenge identities, in our construction, the size of instances of the underlying KDM secure SKE needs to be independent of the number of users in the security game.[2]

We can construct SKE that is KDM secure with respect to projection functions and satisfies this efficiency requirement based on the decisional diffie-hellman (DDH) assumption [BHHO08] and LWE assumption [ACPS09].[3] In addition, Applebaum [App11] showed how to transform SKE that is KDM secure with respect to projection functions into SKE that is KDM secure with respect to functions computable by a-priori bounded size circuits.

We can construct IND-ID-CPA secure IBE under the LWE assumption [ABB10]. Moreover, Döttling and Garg [DG17b] recently showed how to construct IND-ID-CPA secure IBE based on the computational diffie-hellman (CDH) assumption.

Our construction also uses garbled circuits, but it is implied by one-way functions [Yao86]. Thus, from Theorem 1, we obtain the following corollary.

Corollary 1 *There exists IBE that is KDM secure with respect to functions computable by a-priori bounded size circuits against adaptive-ID attacks under the DDH assumption or LWE assumption.*

In addition to the above results, based on the construction techniques above, we also show that we can transform KDM secure SKE into KDM secure PKE by using PKE satisfies ordinary indistinguishability against chosen plaintext attacks (IND-CPA security). Specifically, we show the following theorem.

Theorem 2 (Informal) *Assuming there exist IND-CPA secure PKE and SKE that is KDM secure with respect to projection functions (resp. functions computable by a-priori bounded size circuits). Then, there exists PKE that is KDM secure with respect to projection functions (resp. functions computable by a-priori bounded size circuits).*

It seems that we cannot construct KDM secure PKE from KDM secure SKE via straightforward hybrid encryption methodology. It leads to dead rock of secret keys of the underlying primitives and thus it seems difficult to prove the security of hybrid encryption construction. Thus, we believe this result is of independent interest.

Receiver selective opening secure IBE. Before our work, RSO security for IBE has never been studied while an IBE scheme that is SSO secure was proposed by Bellare et al. [BWY11]. Therefore, we first define RSO security for IBE formally. Our definition is a natural extension of simulation-based RSO security for PKE proposed by Hazay et al. [HPW15]. We then show the following theorem.

Theorem 3 (Informal) *Assuming there exists IND-ID-CPA secure IBE. Then, there exists IBE that satisfies simulation-based RSO security against adaptive-ID attacks.*

Somewhat surprisingly, the above theorem says that all we need is IND-ID-CPA secure IBE to achieve simulation-based RSO secure IBE. We can obtain the result via a simple double encryption paradigm [NY90].

The reason we can obtain the above result via a simple double encryption paradigm is that in receiver selective opening attacks for IBE, we have to consider the revelation of secret keys

[2] For more details, see Remark 1 in Section 2.3.

[3] More precisely, these works showed now to construct PKE that is KDM secure with respect to projection functions and satisfies the efficiency requirement.

itself but not the random coins for key generation since secret keys are generated by the trusted authority in IBE.

From the above, we also observe that if we consider only revelations of secret keys and not the random coins for key generation, we can construct PKE satisfying such simulation-based RSO security using any PKE satisfying ordinary IND-CPA security. This fact is somewhat obvious from some previous results [CHK05, HPW15] though these works did not explicitly state it. For self-containment, we formally show the fact in the appendix. Formally, we have the following theorem.

Theorem 4 (Informal) *Assuming there exist IND-CPA secure PKE. Then, there exists PKE that satisfies simulation-based RSO security considering the revelation of only secret keys.*

To prove simulation-based RSO security against the revelation of random coins for key generation, it seems that the underlying PKE needs to be *key simulatable* [DN00, HPW15] in some sense. In this case, it seems difficult to construct simulation-based RSO secure PKE without relying on some specific algebraic or lattice assumptions.

We summarize our results in Figure 1.

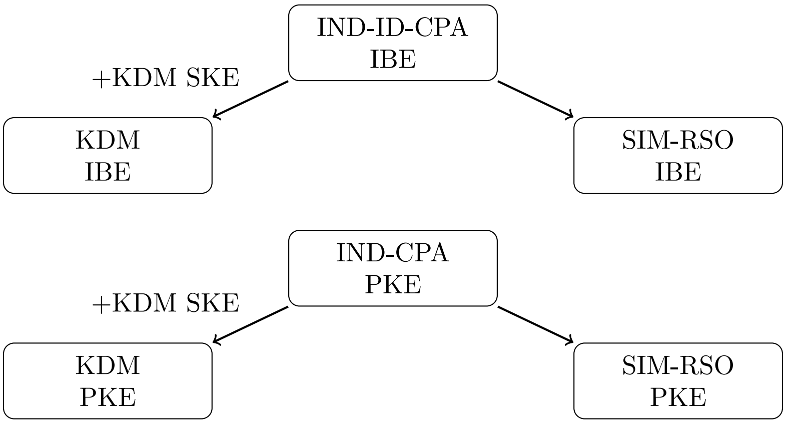

Figure 1: Our results.

1.3 Overview of Our Techniques

We first give an intuition for our KDM secure IBE.

KDM secure IBE from KDM secure SKE. Our construction methodology for KDM secure IBE is somewhat based on the recent beautiful construction of IBE proposed by Döttling and Garg [DG17b, DG17a] using new primitives called *chameleon encryption* or *one-time signatures with encryption*. The essence of their transformations is the mechanism that an encryptor who does not know the exact value of a public-key ek of PKE can generate an "encoding" of a PKE's ciphertext under the public-key ek. Moreover, in their construction, the security of IBE is directly reduced to that of PKE in the last step of the security proof.

This hints that by realizing the mechanism that an encryptor who does not know the value of the key K of SKE can generate an encoding of an SKE's ciphertext under the key K of SKE, we can transform SKE into public primitives such as PKE and IBE shifting the security level of SKE to them. We show that this intuition is true by demonstrating constructions of

KDM secure IBE (resp. PKE) based on KDM secure SKE and IND-ID-CPA secure IBE (resp. IND-CPA secure PKE).

We emphasize that we need neither chameleon encryption nor one-time signatures with encryption. This is because our aim is to construct IBE satisfying a strong security notion, that is, KDM security, and we use IBE satisfying ordinary IND-ID-CPA security as a building block while the goal of Döttling and Garg in the above work is to construct IND-ID-CPA secure IBE.

Our constructions are very simple and use garbled circuits. For simplicity, we focus on constructing KDM secure PKE to give an intuition. Suppose that we construct a KDM secure PKE scheme KdmPKE from a KDM secure SKE scheme SKE and IND-CPA secure PKE scheme PKE.

Basically speaking, the encryption algorithm of KdmPKE first garbles an encryption circuit of SKE that has a message to be encrypted hardwired, that is, $\mathsf{E_{ske}}(\cdot, m)$, and then encrypts labels of the garbled circuit by PKE under different keys. This process can be done without any secret-key of SKE and thus we achieve the "encoding" mechanism mentioned above. This construction is similar to that of "semi-adaptively" secure functional encryption based on selectively secure one proposed by Goyal, Koppula, and Waters [GKW16], but our techniques for the security proof explained below are different from theirs.

Why IND-CPA security of the underlying PKE is sufficient? One might wonder why IND-CPA security of the underlying PKE scheme PKE is sufficient to construct the KDM secure PKE scheme KdmPKE. To see the answer for this question, we closer look at the construction of KdmPKE.

Let the length of a secret-key K of SKE be $\mathsf{len_K}$. A public-key $\mathsf{Kdm.ek}$ of KdmPKE consists of $2 \cdot \mathsf{len_K}$ public-keys of PKE, $\{\mathsf{ek}_{j,\alpha}\}_{j\in[\mathsf{len_K}],\alpha\in\{0,1\}}$. The secret-key $\mathsf{Kdm.dk}$ corresponding to $\mathsf{Kdm.ek}$ consists of a secret-key K of SKE and $\mathsf{len_K}$ secret-keys of PKE corresponding to the bit representation of $\mathsf{K} = \mathsf{K}[1]\ldots\mathsf{K}[\mathsf{len_K}]$, that is, $\left\{\mathsf{dk}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}$. We note that secret-keys of PKE that do not correspond to the bit representation of K are not included in $\mathsf{Kdm.dk}$.

As mentioned above, when encrypting a message m under the public-key $\mathsf{Kdm.ek} := \{\mathsf{ek}_{j,\alpha}\}_{j\in[\mathsf{len_K}],\alpha\in\{0,1\}}$, the encryption algorithm of KdmPKE first garbles an encryption circuit of SKE in which m is hardwired, that is, $\mathsf{E_{ske}}(\cdot, m)$. This results in a single garbled circuit $\widetilde{\mathsf{E}}$ and $2 \cdot \mathsf{len_K}$ labels $\{\mathsf{lab}_{j,\alpha}\}_{j\in[\mathsf{len_K}],\alpha\in\{0,1\}}$. Then, the encryption algorithm of KdmPKE encrypts $\mathsf{lab}_{j,\alpha}$ by $\mathsf{ek}_{j,\alpha}$ for every $j \in [\mathsf{len_K}]$ and $\alpha \in \{0,1\}$. The resulting ciphertext of KdmPKE consists of $\widetilde{\mathsf{E}}$ and these $2 \cdot \mathsf{len_K}$ ciphertexts of PKE.

When decrypting this ciphertext with $\mathsf{Kdm.dk} := \left(\mathsf{K}, \left\{\mathsf{dk}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}\right)$, we first obtain labels corresponding to K from $\mathsf{len_K}$ out of $2\cdot\mathsf{len_K}$ ciphertexts of PKE using $\left\{\mathsf{dk}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}$ and evaluate $\widetilde{\mathsf{E}}$ with those labels. This results in an SKE's ciphertext $\mathsf{E_{ske}}(\mathsf{K}, m)$. Thus, by decrypting it with K, we obtain m.

In this construction, secret-keys of PKE corresponding to K, that is, $\left\{\mathsf{dk}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}$ are included in $\mathsf{Kdm.dk}$, but the rest of secret-keys $\left\{\mathsf{dk}_{j,1-\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}$ are not included in $\mathsf{Kdm.dk}$. Thus, even if adversaries for KdmPKE obtain encryptions of key dependent messages, they cannot get information of $\left\{\mathsf{dk}_{j,1-\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}$ while they potentially get information of $\left\{\mathsf{dk}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}$ from those encryptions. This is the reason the IND-CPA security of PKE is sufficient to construct a KDM secure PKE scheme KdmPKE since we use the security of PKE of instances related to $\left\{\mathsf{dk}_{j,1-\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}$, but not $\left\{\mathsf{dk}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len_K}]}$ in the security proof. To see the fact, we show the outline of the proof below.

In the proof, using the security of garbled circuits, we change the security game without affecting the behavior of adversaries so that we generate a challenge ciphertext under the key pair $(\mathsf{Kdm.ek}, \mathsf{Kdm.dk})$ with simulated garbled circuits computed from an SKE's ciphertext of the challenge key dependent message m^* under the key K, that is, $\mathsf{E}_{\mathsf{ske}}(\mathsf{K}, m^*)$, where K is the secret-key of SKE contained in $\mathsf{Kdm.dk}$. By this change, we do not need m^* itself, and the ciphertext $\mathsf{E}_{\mathsf{ske}}(\mathsf{K}, m^*)$ is sufficient to simulate the security game. Thus, at this point, we can reduce the KDM security of KdmPKE to that of the underlying SKE.

In the above proof, before using the security of garbled circuits, we have to eliminate the labels of garbled circuits that do not correspond to the bit representation of K, that is, $\left\{\mathsf{lab}_{j,1-\mathsf{K}[j]}\right\}_{j\in[\mathsf{len}_{\mathsf{K}}]}$ from the view of adversaires. This can be done by using the IND-CPA security of PKE of only instances related to $\left\{\mathsf{dk}_{j,1-\mathsf{K}[j]}\right\}_{j\in[\mathsf{len}_{\mathsf{K}}]}$ from the construction of KdmPKE. Thus, we can complete the proof by using IND-CPA security of PKE of instances related to $\left\{\mathsf{dk}_{j,1-\mathsf{K}[j]}\right\}_{j\in[\mathsf{len}_{\mathsf{K}}]}$, but not $\left\{\mathsf{dk}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len}_{\mathsf{K}}]}$.

Conversions of functions. One additional non-trivial point is the conversion of functions by reductions.

In the security game of KDM security, adversaries query a function and obtain an encryption of the function of secret keys. Thus, KDM security is parameterized by function classes indicating functions that adversaries can query.

In the above construction, a secret key $\mathsf{Kdm.dk}$ of KdmPKE contains some secret-keys of PKE in addition to a secret key of SKE. Therefore, a function queried by an adversary for KdmPKE is a function of secret-keys of PKE and secret keys of SKE. On the other hand, a function that a reduction algorithm can query is a function of only secret keys of SKE. This means that the reduction algorithm needs to convert a function queried by an adversary for PKE.

Such conversion is clearly possible if we do not care classes of functions. However, when considering KDM security, classes of functions are important since they determine the level of KDM security. It is not clear how such conversions affect a class of functions. Especially, it is not clear whether we can perform such conversions for functions without changing the class of functions.

We show that such conversions are possible for projection functions and functions computable by a-priori bounded size circuits. Thus, we can reduce the KDM security for those function classes of KdmPKE to that of SKE.

These arguments hold if we replace the underlying IND-CPA secure PKE with IND-ID-CPA secure IBE. The above construction can be seen as the special case where the size of instances of the underlying IBE linearly depends on the size of identity space. Thus, we can obtain KDM secure IBE from KDM secure SKE and IND-ID-CPA secure IBE.

RSO secure IBE from IND-ID-CPA secure IBE. Our starting point of the construction of RSO secure IBE is the above KDM secure IBE based on KDM secure SKE. It seems that the above construction can be used to carry over strong security notions of SKE to IBE that we need to simulate secret-keys in some sense in the security game. One such example, we focus on RSO security.[4] Actually, in the above construction, if the underlying SKE has non-committing property (such as one-time pad), the resulting IBE gains simulation-based RSO security.

However, the construction turns out to be redundant and we can dramatically simplify the construction. The reason we can do such a simplification is related to whether we consider the revelation of the random coins for key generation in addition to secret-keys or not in receiver selective opening attacks.

[4] We observe that another example is leakage resilience. We do not focus on it in this paper.

Secret key vs random coins for key generation. Hazay et al. [HPW15] considered the revelation of both secret keys and random coins for key generation when they defined RSO security for PKE. It might be better to consider the revelation of random coins of key generation for many applications of PKE. However, for IBE, it is sufficient to consider the revelation of only secret keys.

In IBE, the trusted authority generates user secret keys and distributes them to users. Thus, if an adversary corrupts a user, the adversary cannot obtain the random coin used to generate the secret key of the user since the user do not know it. For this reason, we do not have to consider the revelation of random coins of key generation in IBE.[5]

Construction based on double encryption paradigm. When we do not consider the revelation of random coins of key generation in IBE, we can construct simulation-based RSO secure IBE via a simple double encryption paradigm [NY90] without using garbled circuits.

More precisely, using an IBE scheme IBE whose identity space is $\mathcal{ID} \times \{0,1\}$, we construct the following new IBE scheme RsoIBE whose message space and identity space are $\{0,1\}$ and $\mathcal{ID}$, respectively.

The setup algorithm of RsoIBE is the same as that of IBE. When generating a secret-key $\mathsf{Rso.sk_{id}}$ for identity $\mathsf{id} \in \mathcal{ID}$, the key generation algorithm of RsoIBE generates an IBE's secret-key $\mathsf{sk}_{\mathsf{id},r}$ for the identity (id, r), where r is a freshly generated random bit, and outputs $\mathsf{Rso.sk_{id}} := (r, \mathsf{sk}_{\mathsf{id},r})$. When encrypting a message $m \in \{0,1\}$ for identity $\mathsf{id} \in \mathcal{ID}$, the encryption algorithm of RsoIBE generates a pair of ciphertexts $(\mathsf{CT}_0, \mathsf{CT}_1)$, where CT_α is an encryption of m under the identity (id, α) for every $\alpha \in \{0,1\}$. The decryption algorithm of RsoIBE, given a pair of ciphertexts $(\mathsf{CT}_0, \mathsf{CT}_1)$ and a secret-key $\mathsf{Rso.sk_{id}} := (r, \mathsf{sk}_{\mathsf{id},r})$, outputs the decryption result of CT_r with $\mathsf{sk}_{\mathsf{id},r}$.

This construction achieves non-committing property. Suppose that we generate CT_r as an encryption of 0 under the identity (id, r) and CT_{1-r} as an encryption of 1 under the identity $(\mathsf{id}, 1-r)$ when generating a ciphertext $(\mathsf{CT}_0, \mathsf{CT}_1)$ for the identity id, where r is the random bit contained in the secret key $\mathsf{Rso.sk_{id}} := (r, \mathsf{sk}_{\mathsf{id},r})$ for id. We can open this ciphertext to any $m \in \{0,1\}$ by pretending as if the secret key $\mathsf{Rso.sk_{id}}$ for id is $(r \oplus m, \mathsf{sk}_{\mathsf{id},r\oplus m})$. Due to this non-committing property, we prove the simulation-based RSO security of RsoIBE.

From this result, we observe that if we consider the revelation of only secret keys, we can also construct SIM-RSO secure PKE based on any IND-CPA secure PKE. Our results on simulation-based RSO secure IBE and PKE highlight the gap of difficulties between achieving RSO security against revelation of only secret-keys and achieving that against both secret-keys and random coins for key generation. To achieve the latter RSO security for PKE, it seems that the underlying scheme needs to be *key simulatable* [DN00, HPW15] in some sense.

1.4 Organization

In Section 2, we introduce some notations and review definitions of cryptographic primitives that we use as building blocks. In Section 3, we define IBE, and introduce KDM security and RSO security for it. In Section 4, we show how to construct KDM secure IBE from KDM secure SKE and IND-ID-CPA secure IBE. In Section 5, we show the construction of simulation-based RSO secure IBE based on IND-ID-CPA secure IBE. In Section 6, we show how to construct KDM secure PKE from KDM secure SKE and IND-CPA secure PKE. In Appendix A, we show how to construct simulation-based RSO secure PKE based on IND-CPA secure PKE.

[5] One additional reason is that we can always make a key generation algorithm of IBE deterministic by using pseudorandom functions.

2 Preliminaries

In this section, we define some notations and cryptographic primitives.

2.1 Notations

In this paper, $x \xleftarrow{\mathsf{r}} X$ denotes selecting an element from a finite set X uniformly at random, and $y \leftarrow \mathsf{A}(x)$ denotes assigning to y the output of an algorithm A on an input x. For strings x and y, $x \| y$ denotes the concatenation of x and y. For an integer ℓ, $[\ell]$ denote the set of integers $\{1, \ldots, \ell\}$. For a string x and positive integer $j \leq |x|$, $x[j]$ denotes the j-th bit of x.

λ denotes a security parameter. PPT stands for probabilistic polynomial time. A function $f(\lambda)$ is a negligible function if $f(\lambda)$ tends to 0 faster than $\frac{1}{\lambda^c}$ for every constant $c > 0$. We write $f(\lambda) = \mathsf{negl}(\lambda)$ to denote $f(\lambda)$ being a negligible function.

2.2 Garbled Circuits

We define garbled circuits. We can realize garbled circuits for all efficiently computable circuits based on one-way functions [Yao86].

Definition 1 (Garbled circuits) *Let $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ be a family of circuits where each circuit in $\mathcal{C}_n$ takes n-bit inputs. A circuit garbling scheme* GC *is a two tuple* $(\mathsf{Garble}, \mathsf{Eval})$ *of PPT algorithms.*

- *The garbling algorithm* Garble*, given a security parameter 1^λ and circuit $C \in \mathcal{C}_n$, outputs a garbled circuit $\widetilde{C}$, together with $2n$ labels $\{\mathsf{lab}_{j,\alpha}\}_{j \in [n], \alpha \in \{0,1\}}$.*
- *The evaluation algorithm, given a garbled circuit $\widetilde{C}$ and n labels $\{\mathsf{lab}_j\}_{j \in [n]}$, outputs y.*

Correctness *We require* $\mathsf{Eval}\left(\widetilde{C}, \left\{\mathsf{lab}_{j,x[j]}\right\}_{j \in [n]}\right) = C(m)$ *for every $n \in \mathbb{N}$, $x \in \{0,1\}^n$, where* $\left(\widetilde{C}, \{\mathsf{lab}_{j,\alpha}\}_{j \in [n], \alpha \in \{0,1\}}\right) \leftarrow \mathsf{Garble}(1^\lambda, C)$.

Security *Let* Sim *be a PPT simulator. We define the following game between a challenger and an adversary $\mathcal{A}$ as follows.*

1. *First, the challenger chooses a bit $b \xleftarrow{\mathsf{r}} \{0,1\}$ and sends a security parameter 1^λ to $\mathcal{A}$. Then, $\mathcal{A}$ sends a circuit $C \in \mathcal{C}_n$ and an input $x \in \{0,1\}^n$ for the challenger. Next, if $b = 1$, the challenger computes $\left(\widetilde{C}, \{\mathsf{lab}_{j,\alpha}\}_{j \in [n], \alpha \in \{0,1\}}\right) \leftarrow \mathsf{Garble}(1^\lambda, C)$ and returns $\left(\widetilde{C}, \left\{\mathsf{lab}_{j,x[j]}\right\}_{j \in [n]}\right)$ to $\mathcal{A}$. Otherwise, the challenger returns $\left(\widetilde{C}, \{\mathsf{lab}_j\}_{j \in [n]}\right) \leftarrow \mathsf{Sim}(1^\lambda, |C|, C(x))$ to $\mathcal{A}$.*
2. *$\mathcal{A}$ outputs $b' \in \{0,1\}$.*

In this game, we define the advantage of the adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{gc}}_{\mathsf{GC},\mathcal{A},\mathsf{Sim}}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right| .$$

We say that GC *is secure if for any PPT adversary $\mathcal{A}$, we have* $\mathsf{Adv}^{\mathsf{gc}}_{\mathsf{GC},\mathcal{A},\mathsf{Sim}}(\lambda) = \mathsf{negl}(\lambda)$.

2.3 Public Key Encryption

We define public key encryption (PKE).

Definition 2 (Public key encryption) *A PKE scheme* PKE *is a three tuple* $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *of PPT algorithms. Below, let* $\mathcal{M}$ *be the message space of* PKE.

- *The key generation algorithm* KG*, given a security parameter* 1^λ*, outputs a public key* ek *and a secret key* dk.
- *The encryption algorithm* Enc*, given a public key* ek *and message* $m \in \mathcal{M}$*, outputs a ciphertext* CT.
- *The decryption algorithm* Dec*, given a secret key* dk *and ciphertext* c*, outputs a message* $\tilde{m} \in \{\bot\} \cup \mathcal{M}$.

Correctness *We require* $\mathsf{Dec}(\mathsf{dk}, \mathsf{Enc}(\mathsf{ek}, m)) = m$ *for every* $m \in \mathcal{M}$ *and* $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KG}(1^\lambda)$.

We introduce indistinguishability against chosen plaintext attacks (IND-CPA security) for PKE.

Definition 3 (IND-CPA security) *Let* PKE *be a PKE scheme. We define the IND-CPA game between a challenger and an adversary* $\mathcal{A}$ *as follows. We let* $\mathcal{M}$ *be the message space of* PKE.

1. *First, the challenger chooses a challenge bit* $b \xleftarrow{\mathsf{r}} \{0,1\}$*. Next, the challenger generates a key pair* $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KG}(1^\lambda)$ *and sends* ek *to* $\mathcal{A}$.
2. $\mathcal{A}$ *sends* $(m_0, m_1) \in \mathcal{M}^2$ *to the challenger. We require that* $|m_0| = |m_1|$*. The challenger computes* $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{ek}, m_b)$ *and returns* CT *to* $\mathcal{A}$.
3. $\mathcal{A}$ *outputs* $b' \in \{0,1\}$.

In this game, we define the advantage of the adversary $\mathcal{A}$ *as*

$$\mathsf{Adv}^{\mathsf{indcpa}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right| .$$

We say that PKE *is IND-CPA secure if for any PPT adversary* $\mathcal{A}$*, we have* $\mathsf{Adv}^{\mathsf{indcpa}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.

Next, we define key dependent message (KDM) security for PKE [BRS03].

Definition 4 (KDM-CPA security) *Let* PKE *be a PKE scheme,* $\mathcal{F}$ *function family, and* ℓ *the number of keys. We define the* $\mathcal{F}$*-KDM-CPA game between a challenger and an adversary* $\mathcal{A}$ *as follows. Let* $\mathcal{DK}$ *and* $\mathcal{M}$ *be the secret key space and message space of* PKE*, respectively.*

1. *First, the challenger chooses a challenge bit* $b \xleftarrow{\mathsf{r}} \{0,1\}$*. Next, the challenger generates* ℓ *key pairs* $\left(\mathsf{ek}^{(k)}, \mathsf{dk}^{(k)}\right) \leftarrow \mathsf{KG}(1^\lambda)$ $(k \in [\ell])$*. The challenger sets* $\mathbf{dk} := \left(\mathsf{dk}^{(1)}, \ldots, \mathsf{dk}^{(\ell)}\right)$ *and sends* $\left(\mathsf{ek}^{(1)}, \ldots, \mathsf{ek}^{(\ell)}\right)$ *to* $\mathcal{A}$.
2. $\mathcal{A}$ *may adaptively make polynomially many KDM queries.*

KDM queries *$\mathcal{A}$ sends $(k, f) \in [\ell] \times \mathcal{F}$ to the challenger. We require that f is a function such that $f : \mathcal{DK}^{\ell} \to \mathcal{M}$. If $b = 1$ then the challenger returns $\mathsf{CT} \leftarrow \mathsf{Enc}\left(\mathsf{ek}^{(k)}, f(\mathbf{dk})\right)$ to $\mathcal{A}$. Otherwise, the challenger returns $\mathsf{CT} \leftarrow \mathsf{Enc}\left(\mathsf{ek}^{(k)}, 0^{|f(\cdot)|}\right)$ to $\mathcal{A}$.*

3. $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

In this game, we define the advantage of the adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{PKE},\mathcal{F},\mathcal{A},\ell}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| .$$

We say that PKE is $\mathcal{F}$-KDM-CPA secure if for any PPT adversary $\mathcal{A}$ and polynomial $\ell = \ell(\lambda)$, we have $\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{PKE},\mathcal{F},\mathcal{A},\ell}(\lambda) = \mathsf{negl}(\lambda)$.

Remark 1 (Flexibility of the number of users) The above definition implicitly requires that the size of instances such as public keys, secret keys, and ciphertexts be independent of the number of users ℓ. We require the same condition for KDM secure SKE. This requirement is necessary for our constructions of KDM secure IBE (and PKE) based on KDM secure SKE.

When we reduce the KDM security of our IBE to that of the underlying SKE, the number of users ℓ in the security game of SKE corresponds to the number of challenge identities queried by an adversary for IBE. If the size of instances of SKE depends on ℓ, we can prove the KDM security of the resulting IBE only when the number of challenge identities is a-priori bounded.

Function families. As we can see, KDM security is defined with respect to function families. In this paper, we focus on KDM security with respect to the following function families.

Projection functions. A projection function is a function in which each output bit depends on at most a single bit of an input. Let f be a function and $y = y_1 \dots y_m$ be the output of the function f on an input $x = x_1 \dots x_n$, that is $f(x) = y$. We say that f is a projection function if for any $j \in [m]$, there exists $i \in [n]$ such that $y_j \in \{0, 1, x_i, 1 - x_i\}$.

In this paper, we let $\mathcal{P}$ denote the family of projection functions, and we say that PKE is $\mathcal{P}$-KDM-CPA secure if it is KDM-CPA secure with respect to projection functions.

Functions computable by a-priori bounded size circuits. In the security game of KDM-CPA security with respect to this function family, an adversary can query a function computable by a circuit of a-priori bounded size and input and output length. We allow the size of instances of a scheme to depend on these a-priori bounds on functions while we do not allow it to depend on the number of total users as we noted in Remark 1.

In this paper, we say that PKE is $\mathcal{B}$-KDM-CPA secure if it is KDM-CPA secure with respect to functions computable by a-priori bounded size circuits.

$\mathcal{P}$-KDM-CPA security is a generalization of circular security [CL01] and strong enough for many applications. Boneh, Helevi, Hamburg, and Ostrovsky [BHHO08] and Applebaum, Cash, Peikert, and Sahai [ACPS09] showed how to construct $\mathcal{P}$-KDM-CPA secure PKE under the decisional diffie-hellman (DDH) assumption and learning with errors (LWE) assumption, respectively.[6]

[6] Brakerski and Goldwasser [BG10] proposed $\mathcal{P}$-KDM-CPA secure PKE under the quadratic residuosity (QR) assumption and decisional composite residuosity (DCR) assumption, but their schemes do not satisfy the flexibility of the number of users in the sense of Remark 1.

Barak, Haitner, Hofheinz, and Ishai [BHHI10] showed how to construct $\mathcal{B}$-KDM-CPA secure PKE under the DDH assumption or LWE assumption. Applebaum [App11] showed how to transform $\mathcal{P}$-KDM-CPA secure PKE into $\mathcal{B}$-KDM-CPA secure one using garbled circuits.

We next introduce the definition of receiver selective opening (RSO) security for PKE. We adopt the simulation-based definition proposed by Hazay et al. [HPW15].

Definition 5 (SIM-RSO security) *Let* PKE *be a PKE scheme, and* ℓ *the number of keys. Let* $\mathcal{A}$ *and* $\mathcal{S}$ *be a PPT adversary and simulator, respectively. We define the following pair of games.*

Real game

1. *First, the challenger generates* ℓ *key pairs* $\left(\mathsf{ek}^{(k)}, \mathsf{dk}^{(k)}\right) \leftarrow \mathsf{KG}(1^\lambda)$ $(k \in [\ell])$ *and sends* $\left(\mathsf{ek}^{(1)}, \ldots, \mathsf{ek}^{(\ell)}\right)$ *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *sends a message distribution* Dist *to the challenger. The challenger generates* $\{m^{(k)}\}_{k\in[\ell]} \leftarrow \mathsf{Dist}$*, computes* $\mathsf{CT}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}^{(k)}, m^{(k)}\right)$ *for every* $k \in [\ell]$*, and sends* $\left\{\mathsf{CT}^{(k)}\right\}_{k\in[\ell]}$ *to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *sends a subset* $\mathcal{I}$ *of* $[\ell]$ *to the challenger. The challenger sends* $\left\{\left(\mathsf{dk}^{(k)}, m^{(k)}\right)\right\}_{k\in\mathcal{I}}$ *to* $\mathcal{A}$*.*
4. $\mathcal{A}$ *sends a string* out *to the challenger.*
5. *The challenger outputs* $\mathsf{out}_{\mathsf{real}} := \left(\{m^{(k)}\}_{k\in[\ell]}, \mathsf{Dist}, \mathcal{I}, \mathsf{out}\right)$*.*

Simulated game

1. *First, the challenger sends* 1^λ *to* $\mathcal{S}$*.*
2. $\mathcal{S}$ *sends a message distribution* Dist *to the challenger. The challenger generates* $\{m^{(k)}\}_{k\in[\ell]} \leftarrow \mathsf{Dist}$*.*
3. $\mathcal{S}$ *sends a subset* $\mathcal{I}$ *of* $[\ell]$ *to the challenger. The challenger sends* $\{m^{(k)}\}_{k\in\mathcal{I}}$ *to* $\mathcal{S}$*.*
4. $\mathcal{S}$ *sends a string* out *to the challenger.*
5. *The challenger outputs* $\mathsf{out}_{\mathsf{sim}} := \left(\{m^{(k)}\}_{k\in[\ell]}, \mathsf{Dist}, \mathcal{I}, \mathsf{out}\right)$*.*

We say that PKE *is SIM-RSO secure if for any PPT adversary* $\mathcal{A}$ *and polynomial* $\ell = \ell(\lambda)$*, there exists a PPT simulator* $\mathcal{S}$ *such that for any PPT distinguisher* $\mathcal{D}$ *with binary output we have*

$$\mathsf{Adv}^{\mathsf{simrso}}_{\mathsf{PKE},\mathcal{A},\ell,\mathcal{S},\mathcal{D}}(\lambda) = |\Pr[\mathcal{D}(\mathsf{out}_{\mathsf{real}}) = 1] - \Pr[\mathcal{D}(\mathsf{out}_{\mathsf{sim}}) = 1]| = \mathsf{negl}(\lambda) \ .$$

The above definition considers non-adaptive corruptions by adversaries. Namely, adversaries need to corrupt users in one go.

We note that our construction of RSO secure PKE based on IND-CPA secure PKE works well even if we consider adaptive corruptions of adversaries. For simplicity, we define RSO security for PKE against non-adaptive corruptions in this paper.

Secret key vs key generation randomness. We define SIM-RSO security considering only the revelation of secret keys throughout the paper. Namely, we assume that an adversary gets only a secret key itself of a corrupted user and not the random coin used to generate the secret key.

Hazay et al. [HPW15] considered the revelation of both secret keys and random coins for key generation when they defined RSO security for PKE. It might be better to consider the revelation of random coins of key generation for some applications.

We show that by requiring only security against the revelation of secret keys, we can obtain RSO secure PKE from IND-CPA secure PKE. If we consider RSO security against the revelation of random coins for key generation, it seems difficult to construct RSO secure PKE based only on IND-CPA secure PKE without assuming that secure erasure is possible or the underlying scheme is *key simulatable* [DN00, HPW15] in some sense.

2.4 Secret Key Encryption

In this subsection, we define secret key encryption (SKE).

Definition 6 (Secret key encryption) *An SKE scheme* SKE *is a three tuple* $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *of PPT algorithms. Below, let* $\mathcal{M}$ *be the message space of* SKE*.*

- *The key generation algorithm* KG*, given a security parameter* 1^λ*, outputs a secret key* K*.*
- *The encryption algorithm* Enc*, given a secret key* K *and a message* $m \in \mathcal{M}$*, outputs a ciphertext* CT*.*
- *The decryption algorithm* Dec*, given a secret key* K *and a ciphertext* CT*, outputs a message* $\tilde{m} \in \{\bot\} \cup \mathcal{M}$*.*

Correctness *We require* $\mathsf{Dec}(\mathsf{K}, \mathsf{Enc}(\mathsf{K}, m)) = m$ *for every* $m \in \mathcal{M}$ *and* $\mathsf{K} \leftarrow \mathsf{KG}(1^\lambda)$*.*

Next, we define KDM-CPA security for SKE.

Definition 7 (KDM-CPA security for SKE) *Let* SKE *be an SKE scheme whose key space and message space are* $\mathcal{K}$ *and* $\mathcal{M}$*, respectively. Let* $\mathcal{F}$ *be a function family, and* ℓ *the number of keys. We define the* $\mathcal{F}$*-KDM-CPA game between a challenger and an adversary* $\mathcal{A}$ *as follows.*

1. *First, the challenger chooses a challenge bit* $b \xleftarrow{\mathsf{r}} \{0,1\}$*. Next, the challenger generates* ℓ *secret keys* $\mathsf{K}^{(k)} \leftarrow \mathsf{KG}(1^\lambda) (k = 1, \dots, \ell)$*, sets* $\mathbf{K} := \left(\mathsf{K}^{(1)}, \dots, \mathsf{K}^{(\ell)}\right)$*, and sends* 1^λ *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *may adaptively make polynomially many KDM queries.*

   KDM queries $\mathcal{A}$ *sends* $(k, f) \in [\ell] \times \mathcal{F}$ *to the challenger. We require that* f *is a function such that* $f : \mathcal{K}^\ell \rightarrow \mathcal{M}$*. If* $b = 1$*, the challenger returns* $\mathsf{CT} \leftarrow \mathsf{Enc}\left(\mathsf{K}^{(k)}, f(\mathbf{K})\right)$*. Otherwise, the challenger returns* $\mathsf{CT} \leftarrow \mathsf{Enc}\left(\mathsf{K}^{(k)}, 0^{|f(\cdot)|}\right)$*.*
3. $\mathcal{A}$ *outputs* $b' \in \{0,1\}$*.*

In this game, we define the advantage of the adversary $\mathcal{A}$ *as*

$$\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{SKE},\mathcal{F},\mathcal{A},\ell}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right| .$$

We say that SKE *is* $\mathcal{F}$*-KDM-CPA secure if for any PPT adversary* $\mathcal{A}$ *and polynomial* $\ell = \ell(\lambda)$*, we have* $\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{SKE},\mathcal{F},\mathcal{A},\ell}(\lambda) = \mathsf{negl}(\lambda)$*.*

As we noted at Remark 1 after the definition of KDM security for PKE, we require that the size of instances of a KDM-CPA secure SKE scheme be independent of the number of users ℓ. This requirement is necessary for our construction of KDM secure IBE (and PKE) based on KDM secure SKE.

Similarly to KDM security for PKE, we focus on KDM security for SKE with respect to projection functions and that with respect to functions computable by a-priori bounded size circuits. We say that SKE is $\mathcal{P}$-KDM-CPA secure if it is KDM-CPA secure with respect to projection functions. We say that SKE is $\mathcal{B}$-KDM-CPA secure if it is KDM-CPA secure with respect to functions computable by a-priori bounded size circuits.

3 Identity-Based Encryption

We define identity-based encryption encryption (IBE). Then, we introduce KDM security and RSO security for IBE.

Definition 8 (Identity-based encryption) *An IBE scheme* IBE *is a four tuple* $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *of PPT algorithms. Below, let* $\mathcal{ID}$ *and* $\mathcal{M}$ *be the identity space and message space of* IBE*, respectively.*

- *The setup algorithm* Setup*, given a security parameter* 1^λ*, outputs a public parameter* PP *and a master secret key* MSK*.*
- *The key generation algorithm* KG*, given a master secret key* MSK *and identity* $\mathsf{id} \in \mathcal{ID}$*, outputs a user secret key* $\mathsf{sk}_{\mathsf{id}}$*.*
- *The encryption algorithm* Enc*, given a public parameter* PP*, identity* $\mathsf{id} \in \mathcal{ID}$*, and message* $m \in \mathcal{M}$*, outputs a ciphertext* CT*.*
- *The decryption algorithm* Dec*, given a user secret key* $\mathsf{sk}_{\mathsf{id}}$ *and ciphertext* CT*, outputs a message* $\tilde{m} \in \{\bot\} \cup \mathcal{M}$*.*

Correctness *We require* $\mathsf{Dec}(\mathsf{KG}(\mathsf{MSK}, \mathsf{id}), \mathsf{Enc}(\mathsf{PP}, \mathsf{id}, m)) = m$ *for every* $m \in \mathcal{M}$*,* $\mathsf{id} \in \mathcal{ID}$*, and* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$*.*

We define indistinguishability against adaptive-ID attacks (IND-ID-CPA security [BF01]) for IBE.

Definition 9 (IND-ID-CPA security for IBE) *Let* IBE *be an IBE scheme whose identity space and message spare are* $\mathcal{ID}$ *and* $\mathcal{M}$*, respectively. We define the IND-ID-CPA game between a challenger and an adversary* $\mathcal{A}$ *as follows.*

1. *First, the challenger chooses a challenge bit* $b \xleftarrow{\mathsf{r}} \{0,1\}$*. Next, the challenger generates* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and sends* PP *to* $\mathcal{A}$*. Finally, the challenger prepares a list* L_{ext} *which is initially empty.*

   At any step of the game, $\mathcal{A}$ *can make key extraction queries.*

   Extraction queries $\mathcal{A}$ *sends* $\mathsf{id} \in \mathcal{ID}$ *to the challenger. The challenger returns* $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KG}(\mathsf{MSK}, \mathsf{id})$ *to* $\mathcal{A}$ *and adds* id *to* L_{ext}*.*

2. $\mathcal{A}$ *sends* $(\mathsf{id}^*, m_0, m_1) \in \mathcal{ID} \times \mathcal{M} \times \mathcal{M}$ *to the challenger. We require that* $|m_0| = |m_1|$ *and* $\mathsf{id}^* \notin L_{\mathsf{ext}}$*. The challenger computes* $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{id}, m_b)$ *and returns* CT *to* $\mathcal{A}$*.*

   Below, $\mathcal{A}$ *is not allowed to make an extraction query for* id^**.*

3. *$\mathcal{A}$ outputs $b' \in \{0,1\}$.*

In this game, we define the advantage of the adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{indidcpa}}_{\mathsf{IBE},\mathcal{A}}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| \ .$$

We say that IBE *is IND-ID-CPA secure if for any PPT adversary $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{indidcpa}}_{\mathsf{IBE},\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.*

3.1 KDM Security for IBE

Next, we define KDM security for IBE. Alperin-Sheriff and Peikert [AP12] defined KDM security for IBE by extending selective security for IBE [CHK03]. On the other hand, the following definition is an extension of adaptive security for IBE [BF01]. For the difference between the definition of Alperin-Sheriff and Peikert and ours, see Remark 2 after Definition 10.

Definition 10 (KDM-CPA security for IBE) *Let* IBE *be an IBE scheme, and $\mathcal{F}$ a function family. We define the $\mathcal{F}$-KDM-CPA game between a challenger and an adversary $\mathcal{A}$ as follows. Let $\mathcal{SK}$, $\mathcal{ID}$, and $\mathcal{M}$ be the user secret key space, identity space, and message space of* IBE*, respectively.*

1. *First, the challenger chooses a challenge bit $b \xleftarrow{\mathsf{r}} \{0,1\}$. Next, the challenger generates $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends* PP *to $\mathcal{A}$. Finally, the challenger prepares lists $L_{\mathsf{ext}}, L_{\mathsf{ch}}$, and* **sk** *all of which are initially empty.*

2. *$\mathcal{A}$ may adaptively make the following three types of queries polynomially many times.*

   Extraction queries *$\mathcal{A}$ sends $\mathsf{id} \in \mathcal{ID} \setminus (L_{\mathsf{ext}} \cup L_{\mathsf{ch}})$ to the challenger. The challenger returns $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KG}(\mathsf{MSK}, \mathsf{id})$ to $\mathcal{A}$ and adds* id *to L_{ext}.*

   Registration queries *$\mathcal{A}$ sends $\mathsf{id} \in \mathcal{ID} \setminus (L_{\mathsf{ext}} \cup L_{\mathsf{ch}})$ to the challenger. The challenger generates $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KG}(\mathsf{MSK}, \mathsf{id})$ and adds* id *to L_{ch} and $\mathsf{sk}_{\mathsf{id}}$ to* **sk**.

   KDM queries *$\mathcal{A}$ sends $(\mathsf{id}, f) \in L_{\mathsf{ch}} \times \mathcal{F}$ to the challenger. We require that f is a function such that $f : \mathcal{SK}^{|L_{\mathsf{ch}}|} \rightarrow \mathcal{M}$. If $b = 1$, the challenger returns $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{id}, f(\mathbf{sk}))$ to $\mathcal{A}$. Otherwise, the challenger returns $\mathsf{CT} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathsf{id}, 0^{|f(\cdot)|}\right)$ to $\mathcal{A}$.*

3. *$\mathcal{A}$ outputs $b' \in \{0,1\}$.*

In this game, we define the advantage of the adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{IBE},\mathcal{F},\mathcal{A}}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| \ .$$

We say that IBE *is $\mathcal{F}$-KDM-CPA secure if for any PPT adversary $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{IBE},\mathcal{F},\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.*

Similarly to KDM security for PKE, we focus on KDM security for IBE with respect to projection functions and that with respect to functions computable by a-priori bounded size circuits. We say that IBE is $\mathcal{P}$-KDM-CPA secure if it is KDM-CPA secure with respect to projection functions. We say that IBE is $\mathcal{B}$-KDM-CPA secure if it is KDM-CPA secure with respect to functions computable by a-priori bounded size circuits.

Remark 2 (Difference with [AP12]) Alperin-Sheriff and Peikert [AP12] defined KDM security for IBE. Their definition is a natural extension of selective security for IBE [CHK03]. In their definition, an adversary must declare the set of challenge identities L_{ch} at the beginning of the security game. On the other hand, our definition of KDM security for IBE is an extension of adaptive security for IBE [BF01]. In our definition, an adversary can adaptively declare challenge identities through registration queries.[7]

One additional difference between our definition and that of Alperin-Sheriff and Peikert is whether the size of instances of IBE such as a public parameter is allowed to depend on the number of challenge identities or not. In the definition of Alperin-Sheriff and Peikert, the setup algorithm of IBE takes the upper bound on the number of challenge identities as an input, and the size of instances of IBE depend on the number of challenge identities. In our definition, there is no a-priori bound on the number of challenge identities, and thus the size of instances of IBE is required to be independent of the number of challenge identities.

3.2 RSO Security for IBE

We next define RSO security for IBE. We extends the simulation-based definition for PKE proposed by Hazay et al. [HPW15].

Definition 11 (SIM-RSO security for IBE) *Let* IBE *be an IBE scheme whose identity space and message space are* $\mathcal{ID}$ *and* $\mathcal{M}$*, respectively. Let* $\mathcal{A}$ *and* $\mathcal{S}$ *be a PPT adversary and simulator, respectively. We define the following pair of games.*

Real game

1. *The challenger generates public parameter and master secret key* $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and sends* PP *to* $\mathcal{A}$*. The challenger then prepares a list* L_{ext} *which is initially empty. At any step of the game,* $\mathcal{A}$ *can make key extraction queries.*

   Extraction queries $\mathcal{A}$ *sends* $\mathsf{id} \in \mathcal{ID} \setminus L_{\mathsf{ext}}$ *to the challenger. The challenger returns* $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KG}(\mathsf{MSK}, \mathsf{id})$ *to* $\mathcal{A}$ *and adds* id *to* L_{ext}*.*

2. $\mathcal{A}$ *sends* q *identities* $\left\{\mathsf{id}^{(k)} \in \mathcal{ID} \setminus L_{\mathsf{ext}}\right\}_{k\in[q]}$ *and a message distribution* Dist *on* $\mathcal{M}^q$ *to the challenger, where* q *is an a-priori unbounded polynomial of* λ*. The challenger generates* $\left\{m^{(k)}\right\}_{k\in[q]} \leftarrow \mathsf{Dist}$*, computes* $\mathsf{CT}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathsf{id}^{(k)}, m^{(k)}\right)$ *for every* $k \in [q]$*, and sends* $\left\{\mathsf{CT}^{(k)}\right\}_{k\in[q]}$ *to* $\mathcal{A}$*.*

   In the rest of the game, $\mathcal{A}$ *is not allowed to make extraction queries for* $\left\{\mathsf{id}^{(k)}\right\}_{k\in[q]}$*.*

3. $\mathcal{A}$ *sends a subset* $\mathcal{I}$ *of* $[q]$ *to the challenger. The challenger computes* $\mathsf{sk}_{\mathsf{id}^{(k)}} \leftarrow \mathsf{KG}\left(\mathsf{MSK}, \mathsf{id}^{(k)}\right)$ *for every* $k \in \mathcal{I}$ *and sends* $\left\{\left(\mathsf{sk}_{\mathsf{id}^{(k)}}, m^{(k)}\right)\right\}_{k\in\mathcal{I}}$ *to* $\mathcal{A}$*.*

4. $\mathcal{A}$ *sends a string* out *to the challenger.*

5. *The challenger outputs* $\mathsf{out}_{\mathsf{real}} = \left(\left\{\mathsf{id}^{(k)}\right\}_{k\in[q]}, \left\{m^{(k)}\right\}_{k\in[q]}, \mathsf{Dist}, \mathcal{I}, \mathsf{out}\right)$*.*

Simulated game

1. *First, the challenger sends* 1^λ *to* $\mathcal{S}$*.*

[7] One might think it is a restriction to force an adversary to register challenge identities before making KDM queries. This is not the case since the adversary is allowed to adaptively make registration and KDM queries. Our definition with registration queries makes the security proof of our IBE simple.

2. *$\mathcal{S}$ sends q identities $\left\{\mathsf{id}^{(k)} \in \mathcal{ID} \setminus L_{\mathsf{ext}}\right\}_{k\in[q]}$ and a message distribution Dist on $\mathcal{M}^q$ to the challenger, where q is an a-priori unbounded polynomial of λ. The challenger generates $\{m^{(k)}\}_{k\in[q]} \leftarrow \mathsf{Dist}$.*
3. *$\mathcal{S}$ sends a subset $\mathcal{I}$ of $[q]$ to the challenger. The challenger sends $\{m^{(k)}\}_{k\in\mathcal{I}}$ to $\mathcal{S}$.*
4. *$\mathcal{S}$ sends a string out to the challenger.*
5. *The challenger outputs $\mathsf{out}_{\mathsf{sim}} := \left(\left\{\mathsf{id}^{(k)}\right\}_{k\in[q]}, \{m^{(k)}\}_{k\in[q]}, \mathsf{Dist}, \mathcal{I}, \mathsf{out}\right)$.*

Then, we say that IBE *is SIM-RSO secure if for any PPT adversary $\mathcal{A}$, there exists a PPT simulator $\mathcal{S}$ such that for any PPT distinguisher $\mathcal{D}$ with binary output we have*

$$\mathsf{Adv}^{\mathsf{simrso}}_{\mathsf{IBE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) = |\Pr[\mathcal{D}(\mathsf{out}_{\mathsf{real}}) = 1] - \Pr[\mathcal{D}(\mathsf{out}_{\mathsf{sim}}) = 1]| = \mathsf{negl}(\lambda) \ .$$

As we noted after defining SIM-RSO security for PKE, for simplicity, we consider non-adaptive corruptions by adversaries in this paper. We note that our construction of RSO secure IBE based on IND-ID-CPA secure IBE works well if we consider adaptive corruptions by adversaries.

Remark 3 (On the syntax of simulators) In the above definition, not only an adversary but also a simulator is required to output challenge identities with a message distribution, and these identities are given to a distinguisher of games. One might think this is somewhat strange since these identities output by a simulator are never used in the simulated game. This syntax of simulators is similar to that used by Bellare et al. [BWY11] when they defined simulation-based sender selective opening security for IBE.

It seems not to be a big issue whether we require a simulator to output identities or not. This intuition comes from the fact that we allow an adversary and simulator to output arbitrary length strings, and thus they can always include challenge identities into the output strings.

However, this subtle issue might divide notions of selective opening security for IBE. Especially, it looks hard to prove that the definition with simulators without outputting identities imply that with simulators outputting identities, while it is easy to prove the implication of the opposite direction. This means that the former definition is possibly weaker than the latter one.

From these facts, similarly to Bellare et al. [BWY11], we adopt the definition with simulators explicitly outputting identities in this work.

4 KDM Secure IBE from KDM Secure SKE and IND-ID-CPA Secure IBE

We show how to construct KDM secure IBE based on KDM secure SKE and IND-ID-CPA secure IBE. The construction also uses a circuit garbling scheme.

Let $\mathsf{SKE} = (\mathsf{G}, \mathsf{E}, \mathsf{D})$ be an SKE scheme whose message space is $\mathcal{M}$. Let $\mathsf{len}_{\mathsf{K}}$ and $\mathsf{len}_{\mathsf{r}}$ denote the length of a secret key and encryption randomness of SKE, respectively. Let $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE scheme whose identity space is $\mathcal{ID} \times \{0,1\}^{\mathsf{len}_{\mathsf{K}}} \times \{0,1\}$. Let $\mathsf{GC} = (\mathsf{Garble}, \mathsf{Eval})$ be a garbling scheme. Using $\mathsf{SKE}, \mathsf{IBE}$, and GC, we construct the following IBE scheme $\mathsf{KdmIBE} = (\mathsf{Kdm.Setup}, \mathsf{Kdm.KG}, \mathsf{Kdm.Enc}, \mathsf{Kdm.Dec})$ whose message space and identity space are $\mathcal{M}$ and $\mathcal{ID}$, respectively.

Construction. KdmIBE consists of the following algorithms.

$\mathsf{Kdm.Setup}(1^\lambda)$:

- Return $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$.

$\mathsf{Kdm.KG}(\mathsf{MSK}, \mathsf{id})$:

- Generate $\mathsf{K_{id}} \leftarrow \mathsf{G}(1^\lambda)$.
- Generate $\mathsf{sk}_{\mathsf{id},j,\mathsf{K_{id}}[j]} \leftarrow \mathsf{KG}(\mathsf{MSK}, (\mathsf{id}, j, \mathsf{K_{id}}[j]))$ for every $j \in [\mathsf{len_K}]$.
- Return $\mathsf{Kdm.sk_{id}} := \left(\mathsf{K_{id}}, \left\{\mathsf{sk}_{\mathsf{id},j,\mathsf{K_{id}}[j]}\right\}_{j\in[\mathsf{len_K}]}\right)$.

$\mathsf{Kdm.Enc}(\mathsf{PP}, \mathsf{id}, m)$:

- Generate $r_\mathsf{E} \xleftarrow{\mathsf{r}} \{0,1\}^{\mathsf{len_r}}$ and compute $\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_{j,\alpha}\}_{j\in[\mathsf{len_K}],\alpha\in\{0,1\}}\right) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{E}(\cdot, m; r_\mathsf{E}))$, where $\mathsf{E}(\cdot, m; r_\mathsf{E})$ is the encryption circuit E of SKE into which m and r_E are hardwired.
- For every $j \in [\mathsf{len_K}]$ and $\alpha \in \{0,1\}$, compute $\mathsf{CT}_{j,\alpha} \leftarrow \mathsf{Enc}(\mathsf{PP}, (\mathsf{id}, j, \alpha), \mathsf{lab}_{j,\alpha})$.
- Return $\mathsf{Kdm.CT} := \left(\widetilde{\mathsf{E}}, \{\mathsf{CT}_{j,\alpha}\}_{j\in[\mathsf{len_K}],\alpha\in\{0,1\}}\right)$.

$\mathsf{Kdm.Dec}(\mathsf{Kdm.sk_{id}}, \mathsf{Kdm.CT})$:

- Parse $\left(\mathsf{K_{id}}, \{\mathsf{sk}_{\mathsf{id},j}\}_{j\in[\mathsf{len_K}]}\right) \leftarrow \mathsf{Kdm.sk_{id}}$.
- Parse $\left(\widetilde{\mathsf{E}}, \{\mathsf{CT}_{j,\alpha}\}_{j\in[\mathsf{len_K}],\alpha\in\{0,1\}}\right) \leftarrow \mathsf{Kdm.CT}$.
- For every $j \in [\mathsf{len_K}]$, compute $\mathsf{lab}_j \leftarrow \mathsf{Dec}\left(\mathsf{sk}_{\mathsf{id},j}, \mathsf{CT}_{j,\mathsf{K_{id}}[j]}\right)$.
- Compute $\mathsf{CT_{ske}} \leftarrow \mathsf{Eval}\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_j\}_{j\in[\mathsf{len_K}]}\right)$.
- Return $m \leftarrow \mathsf{D}(\mathsf{K_{id}}, \mathsf{CT_{ske}})$.

Correctness. When decrypting a ciphertext of KdmIBE that encrypts a message m, we first obtain a ciphertext of SKE that encrypts m from the correctness of IBE and GC. The correctness of KdmIBE then follows from that of SKE.

We prove the following theorem.

Theorem 5 *Let* SKE *be an SKE scheme that is $\mathcal{P}$-KDM-CPA secure (resp. $\mathcal{B}$-KDM-CPA secure). Let* IBE *be an IND-ID-CPA secure IBE scheme and* GC *a secure garbling scheme. Then,* KdmIBE *is an IBE scheme that is $\mathcal{P}$-KDM-CPA secure (resp. $\mathcal{B}$-KDM-CPA secure).*

Proof of Theorem 5. Let $\mathcal{A}$ be an adversary that attacks the $\mathcal{P}$-KDM-CPA security of KdmIBE and makes at most q_ch registration queries and q_kdm KDM queries. We proceed the proof via a sequence of games. For every $t \in \{0, \ldots, 2\}$, let $\mathtt{SUC}_t$ be the event that $\mathcal{A}$ succeeds in guessing the challenge bit b in Game t.

Game 0: This is the original $\mathcal{P}$-KDM-CPA game regarding KdmIBE. Then, we have $\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{KdmIBE},\mathcal{P},\mathcal{A}} = \left|\Pr[\mathtt{SUC}_0] - \frac{1}{2}\right|$. The detailed description is as follows.

1. The challenger chooses a challenge bit $b \xleftarrow{\mathsf{r}} \{0,1\}$, generates $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$, and sends PP to $\mathcal{A}$. The challenger also prepares lists $L_\mathsf{ext}, L_\mathsf{ch}$, and $\mathbf{sk}_\mathsf{kdm}$ all of which are initially empty.

2. $\mathcal{A}$ may adaptively make the following three types of queries.

Extraction queries $\mathcal{A}$ sends $\mathsf{id} \in \mathcal{ID} \setminus (L_{\mathsf{ext}} \cup L_{\mathsf{ch}})$ to the challenger. The challenger responds as follows.

- The challenger generates $\mathsf{K}_{\mathsf{id}} \leftarrow \mathsf{G}(1^\lambda)$.
- The challenger generates $\mathsf{sk}_{\mathsf{id},j,\mathsf{K}_{\mathsf{id}}[j]} \leftarrow \mathsf{KG}(\mathsf{MSK}, (\mathsf{id}, j, \mathsf{K}_{\mathsf{id}}[j]))$ for every $j \in [\mathsf{len}_{\mathsf{K}}]$ and $\alpha \in \{0,1\}$.
- The challenger returns $\mathsf{Kdm.sk}_{\mathsf{id}} := \left(\mathsf{K}_{\mathsf{id}}, \left\{\mathsf{sk}_{\mathsf{id},j,\mathsf{K}_{\mathsf{id}}[j]}\right\}_{j \in [\mathsf{len}_{\mathsf{K}}]}\right)$ to $\mathcal{A}$ and adds id to L_{ext}.

Registration queries $\mathcal{A}$ sends $\mathsf{id} \in \mathcal{ID} \setminus (L_{\mathsf{ext}} \cup L_{\mathsf{ch}})$ to the challenger. The challenger generates $\mathsf{Kdm.sk}_{\mathsf{id}}$ in the same way as the answer to an extraction query. The challenger then adds id to L_{ch} and $\mathsf{Kdm.sk}_{\mathsf{id}}$ to $\mathbf{sk}_{\mathsf{kdm}}$.

KDM queries $\mathcal{A}$ sends $(\mathsf{id}, f) \in L_{\mathsf{ch}} \times \mathcal{P}$ to the challenger. The challenger responds as follow.

(a) The challenger sets $m_1 := f(\mathbf{sk}_{\mathsf{kdm}})$ and $m_0 := 0^{|m_1|}$.

(b) The challenger computes $\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_{j,\alpha}\}_{j \in [\mathsf{len}_{\mathsf{K}}], \alpha \in \{0,1\}}\right) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{E}(\cdot, m_b; r_{\mathsf{E}}))$, where $r_{\mathsf{E}} \xleftarrow{\mathsf{r}} \{0,1\}^{\mathsf{len}_{\mathsf{r}}}$.

(c) For every $j \in [\mathsf{len}_{\mathsf{K}}]$ and $\alpha \in \{0,1\}$, the challenger computes $\mathsf{CT}_{j,\alpha} \leftarrow \mathsf{Enc}(\mathsf{PP}, (\mathsf{id}, j, \alpha), \mathsf{lab}_{j,\alpha})$.

(d) The challenger returns $\mathsf{Kdm.CT} := \left(\widetilde{\mathsf{E}}, \{\mathsf{CT}_{j,\alpha}\}_{j \in [\mathsf{len}_{\mathsf{K}}], \alpha \in \{0,1\}}\right)$.

3. $\mathcal{A}$ outputs $b' \in \{0,1\}$.

Game 1: Same as Game 0 except the following. When $\mathcal{A}$ makes a KDM query $(\mathsf{id}, f) \in L_{\mathsf{ch}} \times \mathcal{P}$, for every $j \in [\mathsf{len}_{\mathsf{K}}]$ the challenger computes $\mathsf{CT}_{j,1-\mathsf{K}_{\mathsf{id}}[j]} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, (\mathsf{id}, j, 1 - \mathsf{K}_{\mathsf{id}}[j]), \mathsf{lab}_{j,\mathsf{K}_{\mathsf{id}}[j]}\right)$, where K_{id} is the secret key of SKE generated when id was registered to L_{ch}. Recall that in Game 0, $\mathsf{CT}_{j,1-\mathsf{K}_{\mathsf{id}}[j]}$ is generated as $\mathsf{CT}_{j,1-\mathsf{K}_{\mathsf{id}}[j]} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, (\mathsf{id}, j, 1 - \mathsf{K}_{\mathsf{id}}[j]), \mathsf{lab}_{j,1-\mathsf{K}_{\mathsf{id}}[j]}\right)$. Namely, we eliminate labels of garbled circuits that do not correspond to K_{id} from the view of $\mathcal{A}$ in this game.

In order to simulate both Game 0 and 1, we do not need user secret keys of IBE that do not correspond to $\{\mathsf{K}_{\mathsf{id}}\}_{\mathsf{id} \in L_{\mathsf{ch}}}$, that is $\left\{\mathsf{sk}_{\mathsf{id},j,1-\mathsf{K}_{\mathsf{id}}[j]}\right\}_{\mathsf{id} \in L_{\mathsf{ch}}, j \in [\mathsf{len}_{\mathsf{K}}]}$ while we need $\left\{\mathsf{sk}_{\mathsf{id},j,\mathsf{K}_{\mathsf{id}}[j]}\right\}_{\mathsf{id} \in L_{\mathsf{ch}}, j \in [\mathsf{len}_{\mathsf{K}}]}$ to compute the value of $f(\mathbf{sk}_{\mathsf{kdm}})$ when $\mathcal{A}$ makes a KDM query. Therefore, we can use the IND-ID-CPA security of IBE when the challenge identity is $(\mathsf{id}, j, 1 - \mathsf{K}_{\mathsf{id}}[j])$ for every $\mathsf{id} \in L_{\mathsf{ch}}$ and $j \in [\mathsf{len}_{\mathsf{K}}]$. By using IND-ID-CPA security of IBE $\mathsf{len}_{\mathsf{K}} \cdot q_{\mathsf{kdm}}$ times, we can prove $|\Pr[\mathtt{SUC}_0] - \Pr[\mathtt{SUC}_1]| = \mathsf{negl}(\lambda)$.

Game 2: Same as Game 1 except that to respond to a KDM query from $\mathcal{A}$, the challenger generates a garbled circuit using the simulator for GC. More precisely, when $\mathcal{A}$ makes a KDM query $(\mathsf{id}, f) \in L_{\mathsf{ch}} \times \mathcal{P}$, the challenger generates $r_{\mathsf{E}} \xleftarrow{\mathsf{r}} \{0,1\}^{\mathsf{len}_{\mathsf{r}}}$ and $\mathsf{CT}_{\mathsf{ske}} \leftarrow \mathsf{E}(\mathsf{K}_{\mathsf{id}}, m_b; r_{\mathsf{E}})$, and computes $\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_j\}_{j \in [\mathsf{len}_{\mathsf{K}}]}\right) \leftarrow \mathsf{Sim}(1^\lambda, |\mathsf{E}|, \mathsf{CT}_{\mathsf{ske}})$, where Sim is the simulator for GC and $|\mathsf{E}|$ denotes the size of the encryption circuit E of SKE. Moreover, the challenger computes $\mathsf{CT}_{j,\alpha} \leftarrow \mathsf{Enc}(\mathsf{PP}, (\mathsf{id}, j, \alpha), \mathsf{lab}_j)$ for every $j \in [\mathsf{len}_{\mathsf{K}}]$ and $\alpha \in \{0,1\}$.

In the last step, we eliminate labels of garbled circuits that do not correspond to $\{\mathsf{K}_{\mathsf{id}}\}_{\mathsf{id} \in L_{\mathsf{ch}}}$. Therefore, by using the security of GC q_{kdm} times, we can show that $|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]| = \mathsf{negl}(\lambda)$.

Below, we show that $\left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$ holds by the $\mathcal{P}$-KDM-CPA security of SKE. Using the adversary $\mathcal{A}$, we construct an adversary $\mathcal{A}_{\mathsf{ske}}$ that attacks the $\mathcal{P}$-KDM-CPA security of SKE when the number of keys is q_{ch}.

Before describing $\mathcal{A}_{\mathsf{ske}}$, we note on the conversion of projection functions. We let $\mathsf{K}^{(k)}$ be the secret key of SKE generated to respond to the k-th registration query $\mathsf{id}^{(k)}$ made by $\mathcal{A}$. We let $\alpha_{k,j}$ denote the j-th bit of $\mathsf{K}^{(k)}$, that is, $\mathsf{K}^{(k)}[j]$ for every $j \in [\mathsf{len}_{\mathsf{K}}]$ and $k \in [q_{\mathsf{ch}}]$. Let f be a projection function that $\mathcal{A}$ queries as a KDM query. f is a projection function of $\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$ and $\left\{\mathsf{sk}_{\mathsf{id}^{(k)},j,\alpha_{k,j}}\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}$. To attack the $\mathcal{P}$-KDM-CPA security of SKE, $\mathcal{A}_{\mathsf{ske}}$ needs to compute a projection function g such that

$$g\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}\right) = f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}, \left\{\mathsf{sk}_{\mathsf{id}^{(k)},j,\alpha_{k,j}}\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}\right) \ . \tag{1}$$

We can compute such a function g from f and $\left\{\mathsf{sk}_{\mathsf{id}^{(k)},j,\alpha}\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}],\alpha\in\{0,1\}}$ as follows.

We first observe that for every $k \in [q_{\mathsf{ch}}]$ and $j \in [\mathsf{len}_{\mathsf{K}}]$, we can write

$$\begin{aligned}\mathsf{sk}_{\mathsf{id}^{(k)},j,\alpha_{k,j}} &= (1-\alpha_{k,j})\cdot \mathsf{sk}_{\mathsf{id}^{(k)}j,0} \oplus \alpha_{k,j}\cdot \mathsf{sk}_{\mathsf{id}^{(k)}j,1} \\ &= \alpha_{k,j}\cdot\left(\mathsf{sk}_{\mathsf{id}^{(k)},j,1}\oplus \mathsf{sk}_{\mathsf{id}^{(k)},j,0}\right)\oplus \mathsf{sk}_{\mathsf{id}^{(k)},j,0} \ .\end{aligned}$$

We suppose that $\mathsf{sk}_{\mathsf{id}^{(k)},j,1}$ and $\mathsf{sk}_{\mathsf{id}^{(k)},j,0}$ are represented as binary strings and $\oplus$ is done in the bit-wise manner. We define a function $\mathsf{sel}_{k,j}$ as $\mathsf{sel}_{k,j}(\gamma \in \{0,1\}) = \gamma \cdot \left(\mathsf{sk}_{\mathsf{id}^{(k)},j,1}\oplus \mathsf{sk}_{\mathsf{id}^{(k)},j,0}\right) \oplus \mathsf{sk}_{\mathsf{id}^{(k)},j,0}$. Then, we have

$$f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}, \left\{\mathsf{sk}_{\mathsf{id}^{(k)},j,\alpha_{k,j}}\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}\right) = f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}, \left\{\mathsf{sel}_{k,j}\left(\alpha_{k,j}\right)\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}\right) \ .$$

We define $g\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}\right) = f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}, \left\{\mathsf{sel}_{k,j}\left(\mathsf{K}^{(k)}[j]\right)\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}\right)$. Then, g satisfies Equation 1.

We show that if f is a projection function, then so is g. Let γ be an output bit of $g\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}\right) = f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}, \left\{\mathsf{sel}_{k,j}\left(\mathsf{K}^{(k)}[j]\right)\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}\right)$. We say that γ is a projective bit for f (resp. g) if it depends on a single bit of an input for f (resp. g). We also say that γ is a constant bit for f (resp. g) if it does not depend on any bit of an input for f (resp. g).

Since f is a projection function, γ is a constant bit or projective bit for f that depends on either part of $\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$ or $\left\{\mathsf{sel}_{k,j}\left(\mathsf{K}^{(k)}[j]\right)\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}$. Thus, we consider the following three cases. (i) If γ is a constant bit for f, γ is clearly a constant bit for g. (ii) If γ is a projective bit for f and depends on a single bit of $\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$, γ is a projective bit for g since $\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$ is also an input for g. (iii) If γ is a projective bit for f and depends on some bit of $\left\{\mathsf{sel}_{k,j}\left(\mathsf{K}^{(k)}[j]\right)\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}$, γ is a projective bit for g since each bit of $\left\{\mathsf{sel}_{k,j}\left(\mathsf{K}^{(k)}[j]\right)\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}$ depends on a bit $\mathsf{K}^{(k)}[j]$ for some $k \in [q_{\mathsf{ch}}]$ and $j \in [\mathsf{len}_{\mathsf{K}}]$, and $\mathsf{K}^{(k)}[j]$ is a part of an input to g. Therefore, γ is a projective bit or constant bit for g in any case, and thus g is a projection function.

We now describe the adversary $\mathcal{A}_{\mathsf{ske}}$ that uses the above conversion of projection functions.

1. On input 1^λ, $\mathcal{A}_{\mathsf{ske}}$ first generates $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends PP to $\mathcal{A}$. Then, the $\mathcal{A}_{\mathsf{ske}}$ prepares L_{ext} and L_{ch}.

2. $\mathcal{A}_{\mathsf{ske}}$ responds to queries made by $\mathcal{A}$ as follows.

Extraction queries When $\mathcal{A}$ sends $\mathsf{id} \in \mathcal{ID} \setminus (L_{\mathsf{ext}} \cup L_{\mathsf{ch}})$ as an extraction query, $\mathcal{A}_{\mathsf{ske}}$ responds exactly in the same way as the challenger in Game 2. We note that, in this case, $\mathcal{A}_{\mathsf{ske}}$ computes the answer $\mathsf{Kdm.sk}_{\mathsf{id}}$ using a freshly generated key K_{id} of SKE.

Registration queries When $\mathcal{A}$ makes the k-th ($k \le q_{\mathsf{ch}}$) registration query $\mathsf{id}^{(k)} \in \mathcal{ID} \setminus (L_{\mathsf{ext}} \cup L_{\mathsf{ch}})$, $\mathcal{A}_{\mathsf{ske}}$ relates $\mathsf{id}^{(k)}$ to $\mathsf{K}^{(k)}$, where $\mathsf{K}^{(k)}$ is the k-th secret key of SKE generated by the challenger. $\mathcal{A}_{\mathsf{ske}}$ generates $\mathsf{sk}_{\mathsf{id}^{(k)},j,\alpha} \leftarrow \mathsf{KG}\left(\mathsf{MSK}, \left(\mathsf{id}^{(k)}, j, \alpha\right)\right)$ for every $j \in [\mathsf{len}_{\mathsf{K}}]$ and $\alpha \in \{0,1\}$. They are used for the conversion of functions. $\mathcal{A}_{\mathsf{ske}}$ then adds $\mathsf{id}^{(k)}$ to L_{ch}.

KDM queries When $\mathcal{A}$ makes a KDM query $(\mathsf{id}, f) \in L_{\mathsf{ch}} \times \mathcal{P}$, $\mathcal{A}_{\mathsf{ske}}$ responds as follows.

(a) $\mathcal{A}_{\mathsf{ske}}$ first computes a projection function g satisfying

$$g\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}\right) = f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}, \left\{\mathsf{sk}_{\mathsf{id}^{(k)},j,\mathsf{K}^{(k)}[j]}\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}\right)$$

as we noted above from $\left\{\mathsf{sk}_{\mathsf{id}^{(k)},j,\alpha}\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}$.

(b) Let $k \in [q_{\mathsf{ch}}]$ be the number that related to id. Since id was added to L_{ch}, such $k \in [q_{\mathsf{ch}}]$ exists. $\mathcal{A}_{\mathsf{ske}}$ queries (k, g) to the challenger as a KDM query and gets the answer $\mathsf{CT}_{\mathsf{ske}}$.

(c) $\mathcal{A}_{\mathsf{ske}}$ computes $\left(\widetilde{\mathsf{E}}, \left\{\mathsf{lab}_j\right\}_{j\in[\mathsf{len}_{\mathsf{K}}]}\right) \leftarrow \mathsf{Sim}\left(1^\lambda, |\mathsf{E}|, \mathsf{CT}_{\mathsf{ske}}\right)$ and for every $j \in [\mathsf{len}_{\mathsf{K}}]$ and $\alpha \in \{0,1\}$, computes $\mathsf{CT}_{j,\alpha} \leftarrow \mathsf{Enc}(\mathsf{PP}, (\mathsf{id}, j, \alpha), \mathsf{lab}_j)$.

(d) $\mathcal{A}_{\mathsf{ske}}$ returns $\mathsf{Kdm.CT} := \left(\widetilde{\mathsf{E}}, \{\mathsf{CT}_{j,\alpha}\}_{j\in[\mathsf{len}_{\mathsf{K}}],\alpha\in\{0,1\}}\right)$ to $\mathcal{A}$.

3. When $\mathcal{A}$ terminates with output $b' \in \{0,1\}$, $\mathcal{A}_{\mathsf{ske}}$ outputs $\beta' = b'$.

$\mathcal{A}_{\mathsf{ske}}$ perfectly simulates Game 2 for $\mathcal{A}$ in which the challenge bit is the same as that of $\mathcal{P}$-KDM-CPA game of SKE between the challenger and $\mathcal{A}_{\mathsf{ske}}$. Moreover, $\mathcal{A}_{\mathsf{ske}}$ just outputs $\mathcal{A}$'s output. Thus, $\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{SKE},\mathcal{P},\mathcal{A}_{\mathsf{ske}},q_{\mathsf{ch}}}(\lambda) = \left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right|$ holds. Since SKE is $\mathcal{P}$-KDM-CPA secure, $\left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$ holds.

From the above arguments, we see that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{KdmIBE},\mathcal{P},\mathcal{A}}(\lambda) &= \left|\Pr[\mathtt{SUC}_0] - \frac{1}{2}\right| \\ &= \sum_{t=0}^{2} |\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]| + \left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right| = \mathsf{negl}(\lambda) \ .\end{aligned}$$

Since the choice of $\mathcal{A}$ is arbitrary, KdmIBE satisfies $\mathcal{P}$-KDM-CPA security.

On the transformation of $\mathcal{B}$-KDM-CPA secure schemes. We can also construct $\mathcal{B}$-KDM-CPA secure IBE based on $\mathcal{B}$-KDM-CPA secure SKE via the construction. The security proof of $\mathcal{B}$-KDM-CPA secure IBE is in fact almost the same as that of $\mathcal{P}$-KDM-CPA secure IBE. The only issue we need to care is whether the conversion of functions performed by $\mathcal{A}_{\mathsf{ske}}$ is successful or not also when we construct $\mathcal{B}$-KDM-CPA secure IBE.

Let f be a function queried by an adversary $\mathcal{A}$ for KdmIBE. As above, consider a function g such that

$$g\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}\right) = f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}, \left\{\mathsf{sel}_{k,j}\left(\mathsf{K}^{(k)}[j]\right)\right\}_{k\in[q_{\mathsf{ch}}],j\in[\mathsf{len}_{\mathsf{K}}]}\right) \ ,$$

where the function $\mathsf{sel}_{k,j}$ is the function we defined earlier. Since $\mathsf{sel}_{k,j}$ is computable by a circuit of a-priori bounded size, we see that if f is computable by a circuit of a-priori bounded size, then so is g. Therefore, $\mathcal{A}_{\mathsf{ske}}$ can successfully perform the conversion of functions also when constructing $\mathcal{B}$-KDM-CPA secure IBE. □ (**Theorem 5**)

5 SIM-RSO Secure IBE Based on IND-ID-CPA Secure IBE

We can construct SIM-RSO secure IBE based on any IND-ID-CPA secure IBE.

Let $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE scheme whose message space and identity space are $\{0,1\}$ and $\mathcal{ID} \times \{0,1\}$, respectively. Using IBE, we construct the following IBE scheme $\mathsf{RsoIBE} = (\mathsf{Rso.Setup}, \mathsf{Rso.KG}, \mathsf{Rso.Enc}, \mathsf{Rso.Dec})$ whose message space and identity space are $\{0,1\}$ and $\mathcal{ID}$.

Construction. The description of RsoIBE is as follows.

$\mathsf{Rso.Setup}(1^\lambda)$:

- Return $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$.

$\mathsf{Rso.KG}(\mathsf{MSK}, \mathsf{id})$:

- Generate $r \xleftarrow{\mathsf{r}} \{0,1\}$.
- Generate $\mathsf{sk}_{\mathsf{id},r}, \leftarrow \mathsf{KG}(\mathsf{MSK}, (\mathsf{id}, r))$.
- Return $\mathsf{Rso.sk}_{\mathsf{id}} := (r, \mathsf{sk}_{\mathsf{id},r})$.

$\mathsf{Rso.Enc}(\mathsf{PP}, \mathsf{id}, m \in \{0,1\})$:

- For every $\alpha \in \{0,1\}$, compute $\mathsf{CT}_\alpha \leftarrow \mathsf{Enc}(\mathsf{PP}, (\mathsf{id}, \alpha), m)$.
- Return $\mathsf{Rso.CT} := (\mathsf{CT}_0, \mathsf{CT}_1)$.

$\mathsf{Rso.Dec}(\mathsf{Rso.sk}_{\mathsf{id}}, \mathsf{Rso.CT})$:

- Parse $(r, \mathsf{sk}_{\mathsf{id},r}) \leftarrow \mathsf{Rso.dk}$.
- Parse $(\mathsf{CT}_0, \mathsf{CT}_1) \leftarrow \mathsf{Rso.CT}$.
- Return $m \leftarrow \mathsf{Dec}(\mathsf{sk}_{\mathsf{id},r}, \mathsf{CT}_r)$.

Correctness. The correctness of RsoIBE directly follows from that of IBE.

We prove the following theorem.

Theorem 6 *Let* IBE *be an IND-ID-CPA secure IBE scheme. Then,* RsoIBE *is a SIM-RSO secure IBE scheme.*

Proof of Theorem 6. Let $\mathcal{A}$ be an adversary that attacks the SIM-RSO security of RsoIBE. We show the proof via the following sequence of games.

Let $\mathcal{D}$ be an PPT distinguisher with binary output. For every $t \in \{0, \ldots, 2\}$, let $\mathtt{T}_t$ be the event that $\mathcal{D}$ outputs 1 given the output of the challenger in Game t.

Game 0: This is the real game of SIM-RSO security regarding RsoIBE. The detailed description is as follows.

1. First, the challenger generates $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends PP to $\mathcal{A}$. The challenger prepares a list L_{ext}.

   At any step of the game, $\mathcal{A}$ can make key extraction queries.

   Extraction queries $\mathcal{A}$ sends $\mathsf{id} \in \mathcal{ID} \setminus L_{\mathsf{ext}}$ to the challenger. The challenger responds as follows.

   (a) The challenger generates $r \xleftarrow{\mathsf{r}} \{0,1\}$.

   (b) The challenger generates $\mathsf{sk}_{\mathsf{id},r}, \leftarrow \mathsf{KG}(\mathsf{MSK}, (\mathsf{id}, r))$.

   (c) The challenger returns $\mathsf{Rso.sk}_{\mathsf{id}} := (r, \mathsf{sk}_{\mathsf{id},r})$.

2. $\mathcal{A}$ sends q_{ch} identities $\left\{\mathsf{id}^{(k)} \in \mathcal{ID} \setminus L_{\mathsf{ext}}\right\}_{k\in[q_{\mathsf{ch}}]}$ and a message distribution Dist on $\{0,1\}^{q_{\mathsf{ch}}}$ to the challenger, where q_{ch} is an a-priori unbounded polynomial of λ. The challenger generates $\{m^{(k)}\}_{k\in[q_{\mathsf{ch}}]} \leftarrow \mathsf{Dist}$ and computes $\mathsf{Rso.CT}^{(k)}$ for every $k \in [q_{\mathsf{ch}}]$ as follows.

   (a) The challenger computes $\mathsf{CT}_\alpha^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \left(\mathsf{id}^{(k)}, \alpha\right), m^{(k)}\right)$ for every $\alpha \in \{0,1\}$.

   (b) The challenger sets $\mathsf{Rso.CT}^{(k)} := \left(\mathsf{CT}_0^{(k)}, \mathsf{CT}_1^{(k)}\right)$.

   The challenger sends $\left\{\mathsf{Rso.CT}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$ to $\mathcal{A}$.

   Below, $\mathcal{A}$ is not allowed to make extraction queries for $\left\{\mathsf{id}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$.

3. $\mathcal{A}$ sends a subset $\mathcal{I}$ of $[q_{\mathsf{ch}}]$ to the challenger. The challenger generates $\mathsf{Rso.sk}_{\mathsf{id}^{(k)}}$ for every $k \in \mathcal{I}$ as follows.

   (a) The challenger generates $r^{(k)} \xleftarrow{\mathsf{r}} \{0,1\}$.

   (b) The challenger generates $\mathsf{sk}_{\mathsf{id}^{(k)},r^{(k)}}, \leftarrow \mathsf{KG}(\mathsf{MSK}, (\mathsf{id}^{(k)}, r^{(k)}))$.

   (c) The challenger sets $\mathsf{Rso.sk}_{\mathsf{id}} := (r^{(k)}, \mathsf{sk}_{\mathsf{id}^{(k)},r^{(k)}})$.

   The challenger sends $\left\{\left(\mathsf{Rso.sk}_{\mathsf{id}^{(k)}}, m^{(k)}\right)\right\}_{k\in\mathcal{I}}$ to $\mathcal{A}$.

4. $\mathcal{A}$ sends a string out to the challenger.

5. The challenger outputs $\mathsf{out}_{\mathsf{real}} := \left(\left\{\mathsf{id}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}, \{m^{(k)}\}_{k\in[q_{\mathsf{ch}}]}, \mathsf{Dist}, \mathcal{I}, \mathsf{out}\right)$.

Game 1: Same as Game 0 except that for every $k \in [q_{\mathsf{ch}}]$, the challenger generates

$$\mathsf{CT}_{1-r^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \left(\mathsf{id}^{(k)}, 1 - r^{(k)}\right), 1 - m^{(k)}\right) .$$

We note that the challenger generates $\mathsf{CT}_{r^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, (\mathsf{id}^{(k)}, r^{(k)}), m^{(k)}\right)$ for every $k \in [q_{\mathsf{ch}}]$ in both Games 0 and 1.

Secret keys for identities $\left\{\left(\mathsf{id}^{(k)}, 1 - r^{(k)}\right)\right\}_{k\in[q_{\mathsf{ch}}]}$ of IBE are not given to $\mathcal{A}$ regardless of which users $\mathcal{A}$ corrupts in both Games 0 and 1. Therefore, by using the security of IBE q_{ch} times, we can prove $|\Pr[\mathtt{T}_0] - \Pr[\mathtt{T}_1]| = \mathsf{negl}(\lambda)$.

Game 2: Same as Game 1 except that for every $k \in [q_{\mathsf{ch}}]$, the challenger uses $r^{(k)} \oplus m^{(k)}$ instead of $r^{(k)}$ as the random bit contained in the k-th RsoIBE's secret key $\mathsf{Rso.sk}_{\mathsf{id}^{(k)}}$ for $\mathsf{id}^{(k)}$. We note that the challenger does not need $\{r^{(k)}\}_{k\in[q_{\mathsf{ch}}]}$ before generating $\{m^{(k)}\}_{k\in[q_{\mathsf{ch}}]}$. Thus, the transition from Games 1 to 2 makes sense, and $|\Pr[\mathtt{T}_2] - \Pr[\mathtt{T}_3]| = 0$ holds since $r^{(k)} \oplus m^{(k)}$ is distributed uniformly at random for every $k \in [q_{\mathsf{ch}}]$.

In Game 2, uncorrupted messages $\{m^{(k)}\}_{k\in[q_{\mathsf{ch}}]\setminus\mathcal{I}}$ are completely hidden from the view of $\mathcal{A}$. To verify the fact, we confirm that ciphertexts $\left\{\mathsf{Rso.CT}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$ are independent of $\{m^{(k)}\}_{k\in[q_{\mathsf{ch}}]}$.

For every $k \in [q_{\mathsf{ch}}]$, the challenger generates $\mathsf{Rso.CT}^{(k)} = \left(\mathsf{CT}^{(k)}_0, \mathsf{CT}^{(k)}_1\right)$ by generating

$$\mathsf{CT}^{(k)}_{r^{(k)}\oplus m^{(k)}} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \left(\mathsf{id}^{(k)}, r^{(k)} \oplus m^{(k)}\right), m^{(k)}\right) \quad ,$$
$$\mathsf{CT}^{(k)}_{1-r^{(k)}\oplus m^{(k)}} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \left(\mathsf{id}^{(k)}, 1 - r^{(k)} \oplus m^{(k)}\right), 1 - m^{(k)}\right) \quad .$$

We see that, regardless of the value of $m^{(k)} \in \{0,1\}$, the challenger computes

$$\mathsf{CT}^{(k)}_{r^{(k)}} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \left(\mathsf{id}^{(k)}, r^{(k)}\right), 0\right) \quad ,$$
$$\mathsf{CT}^{(k)}_{1-r^{(k)}} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \left(\mathsf{id}^{(k)}, 1 - r^{(k)}\right), 1\right) \quad .$$

Therefore, we see that ciphertexts $\left\{\mathsf{Rso.CT}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$ are independent of $\{m^{(k)}\}_{k\in[q_{\mathsf{ch}}]}$ in Game 2.

Then, we construct a simulator $\mathcal{S}$ that perfectly simulate Game 2 for $\mathcal{A}$. The description of $\mathcal{S}$ is as follows.

1. On input 1^λ, $\mathcal{S}$ generates $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends PP to $\mathcal{A}$.

   Extraction queries When $\mathcal{A}$ sends $\mathsf{id} \in \mathcal{ID} \setminus L_{\mathsf{ext}}$, $\mathcal{S}$ responds as follows.
   (a) $\mathcal{S}$ generates $r \xleftarrow{\mathsf{r}} \{0,1\}$.
   (b) $\mathcal{S}$ generates $\mathsf{sk}_{\mathsf{id},r}, \leftarrow \mathsf{KG}(\mathsf{MSK}, (\mathsf{id}, r))$.
   (c) $\mathcal{S}$ returns $\mathsf{Rso.sk}_{\mathsf{id}} := (r, \mathsf{sk}_{\mathsf{id},r})$ to $\mathcal{A}$.

2. When $\mathcal{A}$ outputs a message distribution Dist with identities $\left\{\mathsf{id}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$, $\mathcal{S}$ sends them to the challenger. Then, $\mathcal{S}$ computes $\mathsf{Rso.CT}^{(k)}$ for every $k \in [q_{\mathsf{ch}}]$ as follows.
   (a) $\mathcal{S}$ computes $r^{(k)} \xleftarrow{\mathsf{r}} \{0,1\}$.
   (b) $\mathcal{S}$ computes $\mathsf{CT}^{(k)}_{r^{(k)}} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \left(\mathsf{id}^{(k)}, r^{(k)}\right), 0\right)$ and $\mathsf{CT}^{(k)}_{1-r^{(k)}} \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \left(\mathsf{id}^{(k)}, 1 - r^{(k)}\right), 1\right)$.
   (c) $\mathcal{S}$ sets $\mathsf{Rso.CT}^{(k)} := \left(\mathsf{CT}^{(k)}_0, \mathsf{CT}^{(k)}_1\right)$.

   $\mathcal{S}$ sends $\left\{\mathsf{Rso.CT}^{(k)}\right\}_{k\in[q_{\mathsf{ch}}]}$ to $\mathcal{A}$.

3. When $\mathcal{A}$ outputs a subset $\mathcal{I}$ of $[q_{\mathsf{ch}}]$, $\mathcal{S}$ sends it to the challenger, and gets $\{m^{(k)}\}_{k\in\mathcal{I}}$. $\mathcal{S}$ computes $\mathsf{sk}_{\mathsf{id}^{(k)},r^{(k)}\oplus m^{(k)}} \leftarrow \mathsf{KG}\left(\mathsf{MSK}, \left(\mathsf{id}^{(k)}, r^{(k)} \oplus m^{(k)}\right)\right)$ sets $\mathsf{Rso.sk}_{\mathsf{id}^{(k)}} := \left(r^{(k)} \oplus m^{(k)}, \mathsf{sk}_{\mathsf{id}^{(k)},r^{(k)}\oplus m^{(k)}}\right.$ for every $k \in \mathcal{I}$, and sends $\left\{\left(\mathsf{Rso.sk}_{\mathsf{id}^{(k)}}, m^{(k)}\right)\right\}_{k\in\mathcal{I}}$ to $\mathcal{A}$.

4. When $\mathcal{A}$ outputs a string out, $\mathcal{S}$ outputs it.

$\mathcal{S}$ perfectly simulates Game 2 for $\mathcal{A}$. Therefore, we have

$$\mathsf{Adv}^{\mathsf{simrso}}_{\mathsf{RsoIBE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) = |\Pr[\mathrm{T}_0] - \Pr[\mathrm{T}_2]| \le \sum_{t=0}^{2} |\Pr[\mathrm{T}_t] - \Pr[\mathrm{T}_{t+1}]| \quad . \tag{2}$$

From the above arguments, we see that each term of the right hand side of Inequality 4 is negligible in λ. Since the choice of $\mathcal{A}$ and $\mathcal{D}$ is arbitrary and the description of $\mathcal{S}$ does not depend on that of $\mathcal{D}$, we see that for any $\mathcal{A}$, there exists $\mathcal{S}$ such that for any $\mathcal{D}$ we have $\mathsf{Adv}^{\mathsf{simrso}}_{\mathsf{RsoIBE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) = \mathsf{negl}(\lambda)$. This means that RsoIBE is SIM-RSO secure. $\square$ (**Theorem 6**)

6 KDM Secure PKE from KDM Secure SKE and IND-CPA Secure PKE

We show how to construct KDM secure PKE based on KDM secure SKE and IND-CPA secure PKE. The construction is similar to that of KDM secure IBE we show in Section 4 except that IND-CPA secure PKE is used instead of IND-ID-CPA secure IBE as a building block.

Let $\mathsf{SKE} = (\mathsf{G}, \mathsf{E}, \mathsf{D})$ be an SKE scheme whose message space is $\mathcal{M}$. Let len_K and len_r denote the length of a secret key and encryption randomness of SKE, respectively. Let $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme and $\mathsf{GC} = (\mathsf{Garble}, \mathsf{Eval})$ a garbling scheme. Using $\mathsf{SKE}, \mathsf{PKE}$, and GC, we construct the following PKE scheme $\mathsf{KdmPKE} = (\mathsf{Kdm.KG}, \mathsf{Kdm.Enc}, \mathsf{Kdm.Dec})$ whose message space is $\mathcal{M}$.

Construction. KdmPKE consists of the following algorithms.

$\mathsf{Kdm.KG}(1^\lambda)$:

- Generate $\mathsf{K} \leftarrow \mathsf{G}(1^\lambda)$.
- Generate $(\mathsf{ek}_{j,\alpha}, \mathsf{dk}_{j,\alpha}) \leftarrow \mathsf{KG}(1^\lambda)$ for every $j \in [\mathsf{len}_\mathsf{K}]$ and $\alpha \in \{0,1\}$.
- Return $\mathsf{Kdm.ek} := \{\mathsf{ek}_{j,\alpha}\}_{j\in[\mathsf{len}_\mathsf{K}],\alpha\in\{0,1\}}$ and $\mathsf{Kdm.dk} := \left(\mathsf{K}, \left\{\mathsf{dk}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len}_\mathsf{K}]}\right)$.

$\mathsf{Kdm.Enc}(\mathsf{Kdm.ek}, m)$:

- Parse $\{\mathsf{ek}_{j,\alpha}\}_{j\in[\lambda],\alpha\in\{0,1\}} \leftarrow \mathsf{Kdm.ek}$.
- Generate $r_\mathsf{E} \xleftarrow{\mathsf{r}} \{0,1\}^{\mathsf{len}_\mathsf{r}}$ and compute $\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_{j,\alpha}\}_{j\in[\mathsf{len}_\mathsf{K}],\alpha\in\{0,1\}}\right) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{E}(\cdot, m; r_\mathsf{E}))$, where $\mathsf{E}(\cdot, m; r_\mathsf{E})$ is the encryption circuit E of SKE into which m and r_E are hardwired.
- For every $j \in [\mathsf{len}_\mathsf{K}]$ and $\alpha \in \{0,1\}$, compute $\mathsf{CT}_{j,\alpha} \leftarrow \mathsf{Enc}(\mathsf{ek}_{j,\alpha}, \mathsf{lab}_{j,\alpha})$.
- Return $\mathsf{Kdm.CT} := \left(\widetilde{\mathsf{E}}, \{\mathsf{CT}_{j,\alpha}\}_{j\in[\mathsf{len}_\mathsf{K}],\alpha\in\{0,1\}}\right)$.

$\mathsf{Kdm.Dec}(\mathsf{Kdm.dk}, \mathsf{Kdm.CT})$:

- Parse $\left(\mathsf{K}, \{\mathsf{dk}_j\}_{j\in[\mathsf{len}_\mathsf{K}]}\right) \leftarrow \mathsf{Kdm.dk}$.
- Parse $\left(\widetilde{\mathsf{E}}, \{\mathsf{CT}_{j,\alpha}\}_{j\in[\mathsf{len}_\mathsf{K}],\alpha\in\{0,1\}}\right) \leftarrow \mathsf{Kdm.CT}$.
- For every $j \in [\mathsf{len}_\mathsf{K}]$, compute $\mathsf{lab}_j \leftarrow \mathsf{Dec}\left(\mathsf{dk}_j, \mathsf{CT}_{j,\mathsf{K}[j]}\right)$.
- Compute $\mathsf{CT}_\mathsf{ske} \leftarrow \mathsf{Eval}\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_j\}_{j\in[\mathsf{len}_\mathsf{K}]}\right)$.
- Return $m \leftarrow \mathsf{D}(\mathsf{K}, \mathsf{CT}_\mathsf{ske})$.

Correctness. When decrypting a ciphertext of KdmPKE that encrypts a message m, we first obtain a ciphertext of SKE that encrypts m from the correctness of PKE and GC. The correctness of KdmPKE then follows from that of SKE.

We prove the following theorem.

Theorem 7 *Let* SKE *be an SKE scheme that is $\mathcal{P}$-KDM-CPA secure (resp. $\mathcal{B}$-KDM-CPA secure). Let* PKE *be an IND-CPA secure PKE scheme and* GC *a secure garbling scheme. Then,* KdmPKE *is a PKE scheme that is $\mathcal{P}$-KDM-CPA secure (resp. $\mathcal{B}$-KDM-CPA secure).*

Proof of Theorem 7. Let $\mathcal{A}$ be an adversary that attacks the $\mathcal{P}$-KDM-CPA security of KdmPKE and makes as most q_{kdm} KDM queries. Let ℓ be a polynomial of λ denoting the number of key pairs. We proceed the proof via a sequence of games. For every $t \in \{0, \ldots, 2\}$, let $\mathtt{SUC}_t$ be the event that $\mathcal{A}$ succeeds in guessing the challenge bit b in Game t.

Game 0: This is the original $\mathcal{P}$-KDM-CPA game regarding KdmPKE when the number of key pairs is ℓ. Then, we have $\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{KdmPKE},\mathcal{P},\mathcal{A},\ell} = \left|\Pr[\mathtt{SUC}_0] - \frac{1}{2}\right|$. The detailed description is as follows.

1. The challenger chooses a challenge bit $b \xleftarrow{\mathsf{r}} \{0,1\}$, and generates a key pair $\left(\mathsf{Kdm.ek}^{(k)}, \mathsf{Kdm.dk}^{(k)}\right)$ for every $k \in [\ell]$ as follows.
   (a) The challenger generates $\mathsf{K}^{(k)} \leftarrow \mathsf{G}(1^\lambda)$.
   (b) The challenger generates $\left(\mathsf{ek}^{(k)}_{j,\alpha}, \mathsf{dk}^{(k)}_{j,\alpha}\right) \leftarrow \mathsf{KG}(1^\lambda)$ for every $j \in [\mathsf{len}_{\mathsf{K}}]$ and $\alpha \in \{0,1\}$.
   (c) The challenger sets
   $$\mathsf{Kdm.ek}^{(k)} := \left\{\mathsf{ek}^{(k)}_{j,\alpha}\right\}_{j\in[\mathsf{len}_{\mathsf{K}}],\alpha\in\{0,1\}}, \mathsf{Kdm.dk}^{(k)} \quad := \left(\mathsf{K}^{(k)}, \left\{\mathsf{dk}^{(k)}_{j,\mathsf{K}[j]}\right\}_{j\in[\mathsf{len}_{\mathsf{K}}]}\right) \ .$$
   The challenger sets $\mathbf{dk}_{\mathsf{kdm}} := \left(\mathsf{Kdm.dk}^{(1)}, \ldots, \mathsf{Kdm.dk}^{(\ell)}\right)$ and sends $\left(\mathsf{Kdm.ek}^{(1)}, \ldots, \mathsf{Kdm.ek}^{(\ell)}\right)$ to $\mathcal{A}$.
2. $\mathcal{A}$ may adaptively make polynomially many KDM queries.
   KDM queries $\mathcal{A}$ sends $(k, f) \in [\ell] \times \mathcal{P}$ to the challenger. The challenger sets $m_1 := f(\mathbf{dk}_{\mathsf{kdm}})$ and $m_0 := 0^{|m_1|}$, and responds as follows.
   (a) The challenger computes $\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_{j,\alpha}\}_{j\in[\mathsf{len}_{\mathsf{K}}],\alpha\in\{0,1\}}\right) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{E}(\cdot, m_b; r_{\mathsf{E}}))$, where $r_{\mathsf{E}} \xleftarrow{\mathsf{r}} \{0,1\}^{\mathsf{len}_{\mathsf{r}}}$.
   (b) For every $j \in [\mathsf{len}_{\mathsf{K}}]$ and $\alpha \in \{0,1\}$, The challenger computes $\mathsf{CT}_{j,\alpha} \leftarrow \mathsf{Enc}\left(\mathsf{ek}^{(k)}_{j,\alpha}, \mathsf{lab}_{j,\alpha}\right)$.
   (c) The challenger returns $\mathsf{Kdm.CT} := \left(\widetilde{\mathsf{E}}, \{\mathsf{CT}_{j,\alpha}\}_{j\in[\mathsf{len}_{\mathsf{K}}],\alpha\in\{0,1\}}\right)$ to $\mathcal{A}$.
3. $\mathcal{A}$ outputs $b' \in \{0,1\}$.

Game 1: Same as Game 0 except that when $\mathcal{A}$ makes a KDM query $(k, f) \in [\ell] \times \mathcal{P}$, the challenger computes $\mathsf{CT}_{j,1-\mathsf{K}^{(k)}[j]} \leftarrow \mathsf{Enc}\left(\mathsf{ek}^{(k)}_{j,1-\mathsf{K}^{(k)}[j]}, \mathsf{lab}_{j,\mathsf{K}^{(k)}[j]}\right)$ for every $j \in [\mathsf{len}_{\mathsf{K}}]$. Namely, we eliminate labels of garbled circuits that do not correspond to $\mathsf{K}^{(k)}$.

In order to simulate both Games 0 and 1, we do not need secret keys of PKE that do not correspond to $\left\{\mathsf{K}^{(k)}\right\}_{k\in[\ell]}$, that is $\left\{\mathsf{dk}^{(k)}_{j,1-\mathsf{K}^{(k)}[j]}\right\}_{k\in[\ell],j\in[\mathsf{len}_{\mathsf{K}}]}$ while we need $\left\{\mathsf{dk}^{(k)}_{j,\mathsf{K}^{(k)}[j]}\right\}_{k\in[\ell],j\in[\mathsf{len}_{\mathsf{K}}]}$ to compute the value of $f(\mathbf{dk}_{\mathsf{kdm}})$ when $\mathcal{A}$ makes a KDM query. Therefore, we can use IND-CPA security of PKE under the keys $\left\{\mathsf{dk}^{(k)}_{j,1-\mathsf{K}^{(k)}[j]}\right\}_{k\in[\ell],j\in[\mathsf{len}_{\mathsf{K}}]}$. By using IND-CPA security of PKE $\mathsf{len}_{\mathsf{K}} \cdot q_{\mathsf{kdm}}$ times, we can prove $|\Pr[\mathtt{SUC}_0] - \Pr[\mathtt{SUC}_1]| = \mathsf{negl}(\lambda)$.

Game 2: Same as Game 1 except that in order to respond to a KDM query from $\mathcal{A}$, the challenger generates a garbled circuit using the simulator for GC. More precisely, when $\mathcal{A}$ makes a KDM query (k, f), the challenger generates $r_{\mathsf{E}} \xleftarrow{\mathsf{r}} \{0,1\}^{\mathsf{len}_{\mathsf{r}}}$ and $\mathsf{CT}_{\mathsf{ske}} \leftarrow \mathsf{E}\left(\mathsf{K}^{(k)}, m_b; r_{\mathsf{E}}\right)$, and computes $\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_j\}_{j\in[\mathsf{len}_{\mathsf{K}}]}\right) \leftarrow \mathsf{Sim}(1^\lambda, |\mathsf{E}|, \mathsf{CT}_{\mathsf{ske}})$, where Sim is the simulator for GC and $|\mathsf{E}|$ denotes the size of the encryption circuit E of SKE. Moreover, for every $j \in [\mathsf{len}_\mathsf{K}]$ and $\alpha \in \{0,1\}$, the challenger computes $\mathsf{CT}_{j,\alpha} \leftarrow \mathsf{Enc}\left(\mathsf{ek}^{(k)}_{j,\alpha}, \mathsf{lab}_j\right)$.

In the last step, we eliminate labels of garbled circuits that do not correspond to $\mathsf{K}^{(k)}$. Therefore, by using the security of GC q_kdm times, we can show that $|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]| = \mathsf{negl}(\lambda)$.

Below, we show that $\left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$ holds by the $\mathcal{P}$-KDM-CPA security of SKE. Using the adversary $\mathcal{A}$, we construct an adversary $\mathcal{A}_\mathsf{ske}$ that attacks the $\mathcal{P}$-KDM-CPA security of SKE when the number of keys is ℓ.

Before describing $\mathcal{A}_\mathsf{ske}$, we note on the conversion of projection functions. We use similar conversion we used in the proof of Theorem 5.

We let $\alpha_{k,j}$ denote $\mathsf{K}^{(k)}[j]$ for every $j \in [\mathsf{len}_\mathsf{K}]$ and $k \in [\ell]$. Let f be a projection function that $\mathcal{A}$ queries as a KDM query. f is a projection function of $\left\{\mathsf{K}^{(k)}\right\}_{k\in[\ell]}$ and $\left\{\mathsf{dk}^{(k)}_{j,\alpha_{k,j}}\right\}_{k\in[\ell],j\in[\mathsf{len}_\mathsf{K}]}$. To attack the $\mathcal{P}$-KDM-CPA security of SKE, $\mathcal{A}_\mathsf{ske}$ needs to compute a projection function g such that

$$g\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[\ell]}\right) = f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[\ell]}, \left\{\mathsf{dk}^{(k)}_{j,\alpha_{k,j}}\right\}_{k\in[\ell],j\in[\mathsf{len}_\mathsf{K}]}\right) \quad . \tag{3}$$

We can compute such a function g from f and $\left\{\mathsf{dk}^{(k)}_{j,\alpha}\right\}_{k\in[\ell],j\in[\mathsf{len}_\mathsf{K}],\alpha\in\{0,1\}}$. We define a function $\mathsf{sel}_{k,j}$ as $\mathsf{sel}_{k,j}(\gamma \in \{0,1\}) = \gamma \cdot \left(\mathsf{dk}^{(k)}_{j,1} \oplus \mathsf{dk}^{(k)}_{j,0}\right) \oplus \mathsf{dk}^{(k)}_{j,0}$. We suppose that $\mathsf{dk}^{(k)}_{j,1}$ and $\mathsf{dk}^{(k)}_{j,0}$ are represented as binary strings and $\oplus$ is done in the bit-wise manner. We then define

$$g\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[\ell]}\right) = f\left(\left\{\mathsf{K}^{(k)}\right\}_{k\in[\ell]}, \left\{\mathsf{sel}_{k,j}\left(\mathsf{K}^{(k)}[j]\right)\right\}_{k\in[\ell],j\in[\mathsf{len}_\mathsf{K}]}\right) \quad .$$

We can prove that if f is a projection function, then so is g and g satisfies Equation 3 as we show in the proof of Theorem 5.

We now describe the adversary $\mathcal{A}_\mathsf{ske}$ that uses the above conversion of projection functions.

1. On input 1^λ, $\mathcal{A}_\mathsf{ske}$ first generates a public key $\mathsf{Kdm.ek}^{(k)}$ of KdmPKE for every $k \in [\ell]$ as follows.

   (a) $\mathcal{A}_\mathsf{ske}$ generates $\left(\mathsf{ek}^{(k)}_{j,\alpha}, \mathsf{dk}^{(k)}_{j,\alpha}\right) \leftarrow \mathsf{KG}(1^\lambda)$ for every $j \in [\mathsf{len}_\mathsf{K}]$ and $\alpha \in \{0,1\}$.

   (b) $\mathcal{A}_\mathsf{ske}$ sets $\mathsf{Kdm.ek}^{(k)} := \left\{\mathsf{ek}^{(k)}_{j,\alpha}\right\}_{j\in[\mathsf{len}_\mathsf{K}],\alpha\in\{0,1\}}$.

   $\mathcal{A}_\mathsf{ske}$ sends $\left(\mathsf{Kdm.ek}^{(1)}, \ldots, \mathsf{Kdm.ek}^{(\ell)}\right)$ to $\mathcal{A}$. We note that $\mathcal{A}_\mathsf{ske}$ uses secret keys of PKE when converting functions queried by $\mathcal{A}$.

2. $\mathcal{A}_\mathsf{ske}$ responds to KDM queries made by $\mathcal{A}$ as follows.

   KDM queries When $\mathcal{A}$ makes a KDM query $(k, f) \in [\ell] \times \mathcal{P}$, $\mathcal{A}_\mathsf{ske}$ responds as follows.

   (a) $\mathcal{A}_\mathsf{ske}$ first computes a projection function g satisfying $g\left(\{\mathsf{K}_k\}_{k\in[\ell]}\right) = f\left(\left\{\mathsf{K}^{(k)}, \mathsf{dk}^{(k)}\right\}_{k\in[\ell]}\right)$ as we noted above, where $\mathsf{dk}^{(k)} = \left\{\mathsf{dk}^{(k)}_{j,\mathsf{K}^{(k)}[j]}\right\}_{j\in[\mathsf{len}_\mathsf{K}]}$.

   (b) $\mathcal{A}_\mathsf{ske}$ queries (k, g) to the challenger and gets the answer CT_ske.

(c) $\mathcal{A}_{\mathsf{ske}}$ computes $\left(\widetilde{\mathsf{E}}, \{\mathsf{lab}_j\}_{j\in[\mathsf{len}_\mathsf{K}]}\right) \leftarrow \mathsf{Sim}\left(1^\lambda, |\mathsf{E}|, \mathsf{CT}_{\mathsf{ske}}\right)$ and for every $j \in [\mathsf{len}_\mathsf{K}]$ and $\alpha \in \{0,1\}$, computes $\mathsf{CT}_{j,\alpha} \leftarrow \mathsf{Enc}\left(\mathsf{ek}^{(k)}_{j,\alpha}, \mathsf{lab}_j\right)$.

(d) $\mathcal{A}_{\mathsf{ske}}$ returns $\mathsf{Kdm.CT} := \left(\widetilde{\mathsf{E}}, \{\mathsf{CT}_{j,\alpha}\}_{j\in[\mathsf{len}_\mathsf{K}],\alpha\in\{0,1\}}\right)$ to $\mathcal{A}$.

3. When $\mathcal{A}$ terminates with output $b' \in \{0,1\}$, $\mathcal{A}_{\mathsf{ske}}$ outputs $\beta' = b'$.

We see that $\mathcal{A}_{\mathsf{ske}}$ perfectly simulates Game 2 for $\mathcal{A}$ in which the challenge bit is the same as that of $\mathcal{P}$-KDM-CPA game of SKE between the challenger and $\mathcal{A}_{\mathsf{ske}}$. Moreover, $\mathcal{A}_{\mathsf{ske}}$ just outputs $\mathcal{A}$'s output. Therefore, we have $\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{SKE},\mathcal{P},\mathcal{A}_{\mathsf{ske}},\ell}(\lambda) = \left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right|$. Since SKE is $\mathcal{P}$-KDM-CPA secure, we see that $\left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$.

From the above arguments, we see that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{kdmcpa}}_{\mathsf{KdmPKE},\mathcal{P},\mathcal{A},\ell}(\lambda) &= \left|\Pr[\mathtt{SUC}_0] - \frac{1}{2}\right| \\ &= \sum_{t=0}^{2} \left|\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]\right| + \left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right| = \mathsf{negl}(\lambda) \ .\end{aligned}$$

Since the choice of $\mathcal{A}$ and ℓ is arbitrary, KdmPKE is $\mathcal{P}$-KDM-CPA secure.

On the transformation of $\mathcal{B}$-KDM-CPA secure schemes. As we noted after the proof of Theorem 5, the above conversions of functions are possible when we consider the transformation of $\mathcal{B}$-KDM-CPA secure schemes since $\mathsf{sel}_{k,j}$ we defined above is computable by a circuit of a-priori bounded size. Thus, we can also construct $\mathcal{B}$-KDM-CPA secure PKE based on $\mathcal{B}$-KDM-CPA secure SKE and IND-CPA secure PKE via the construction. □ (**Theorem 7**)

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Heidelberg, May 2010.

[ABBC10] Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. Cryptographic agility and its relation to circular encryption. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 403–422. Springer, Heidelberg, May 2010.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Heidelberg, August 2009.

[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.

[App11] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 527–546. Springer, Heidelberg, May 2011.

[BDWY12] Mihir Bellare, Rafael Dowsley, Brent Waters, and Scott Yilek. Standard security does not imply security against selective-opening. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 645–662. Springer, Heidelberg, April 2012.

[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.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 1–20. Springer, Heidelberg, August 2010.

[BHHI10] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 423–444. Springer, Heidelberg, May 2010.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, Heidelberg, August 2008.

[BHW15] Allison Bishop, Susan Hohenberger, and Brent Waters. New circular security counterexamples from decision linear and learning with errors. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 776–800. Springer, Heidelberg, November / December 2015.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Heidelberg, April 2009.

[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.

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

[CGH12] David Cash, Matthew Green, and Susan Hohenberger. New definitions and separations for circular security. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 540–557. Springer, Heidelberg, May 2012.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271. Springer, Heidelberg, May 2003.

[CHK05] Ran Canetti, Shai Halevi, and Jonathan Katz. Adaptively-secure, non-interactive public-key encryption. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 150–168. Springer, Heidelberg, February 2005.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Heidelberg, May 2001.

[DG17a] Nico Döttling and Sanjam Garg. From selective ibe to full ibe and selective hibe. *IACR Cryptology ePrint Archive*, 2017. To appear in TCC 2017.

[DG17b] Nico Döttling and Sanjam Garg. Identity-based encryption from the diffie-hellman assumption. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 537–569. Springer, Heidelberg, August 2017.

[DN00] Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 432–450. Springer, Heidelberg, August 2000.

[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.

[GKW16] Rishab Goyal, Venkata Koppula, and Brent Waters. Semi-adaptive security and bundling functionalities made generic and easy. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 361–388. Springer, Heidelberg, October / November 2016.

[HPW15] Carmit Hazay, Arpita Patra, and Bogdan Warinschi. Selective opening security for receivers. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 443–469. Springer, Heidelberg, November / December 2015.

[HR14] Dennis Hofheinz and Andy Rupp. Standard versus selective opening security: Separation and equivalence results. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 591–615. Springer, Heidelberg, February 2014.

[HRW16] Dennis Hofheinz, Vanishree Rao, and Daniel Wichs. Standard security does not imply indistinguishability under selective opening. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 121–145. Springer, Heidelberg, October / November 2016.

[KW16] Venkata Koppula and Brent Waters. Circular security separations for arbitrary length cycles from LWE. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 681–700. Springer, Heidelberg, August 2016.

[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.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, August 1984.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.

A SIM-RSO Secure PKE Based on IND-CPA Secure PKE

We can construct SIM-RSO secure PKE based on any IND-CPA secure PKE if we consider the revelation of only secret keys. The construction is similar to that of SIM-RSO secure IBE we show in Section 5 except that IND-CPA secure PKE is used instead of IND-ID-CPA secure IBE.

Using a PKE scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, we construct the following PKE scheme $\mathsf{RsoPKE} = (\mathsf{Rso.KG}, \mathsf{Rso.Enc}, \mathsf{Rso.Dec})$ whose message space is $\{0, 1\}$.

Construction. The description of RsoPKE is as follows.

$\mathsf{Rso.KG}(1^\lambda)$:

- Generate $(\mathsf{ek}_\alpha, \mathsf{dk}_\alpha) \leftarrow \mathsf{KG}(1^\lambda)$ for every $\alpha \in \{0, 1\}$.
- Generate $r \xleftarrow{\mathsf{r}} \{0, 1\}$.
- Return $\mathsf{Rso.ek} := (\mathsf{ek}_0, \mathsf{ek}_1)$ and $\mathsf{Rso.dk} := (r, \mathsf{dk}_r)$.

$\mathsf{Rso.Enc}(\mathsf{Rso.ek}, m \in \{0, 1\})$:

- Parse $(\mathsf{ek}_0, \mathsf{ek}_1) \leftarrow \mathsf{Rso.ek}$.
- For every $\alpha \in \{0, 1\}$, compute $\mathsf{CT}_\alpha \leftarrow \mathsf{Enc}(\mathsf{ek}_\alpha, m)$.
- Return $\mathsf{Rso.CT} := (\mathsf{CT}_0, \mathsf{CT}_1)$.

$\mathsf{Rso.Dec}(\mathsf{Rso.dk}, \mathsf{Rso.CT})$:

- Parse $(r, \mathsf{dk}_r) \leftarrow \mathsf{Rso.dk}$
- Parse $(\mathsf{CT}_0, \mathsf{CT}_1) \leftarrow \mathsf{Rso.CT}$.
- Return $m \leftarrow \mathsf{Dec}(\mathsf{dk}_r, \mathsf{CT}_r)$.

Correctness. The correctness of RsoPKE directly follows from that of PKE.

We prove the following theorem.

Theorem 8 *Let* PKE *be an IND-CPA secure PKE scheme. Then,* RsoPKE *is a SIM-RSO secure PKE scheme.*

Proof of Theorem 8. Let $\mathcal{A}$ be an adversary that attacks the SIM-RSO security of RsoPKE. Let ℓ be a polynomial of λ denoting the number of key pairs. We show the proof via the following sequence of games.

Let $\mathcal{D}$ be an PPT distinguisher with binary output. For every $t \in \{0, \ldots, 2\}$, let $\mathtt{T}_t$ be the event that $\mathcal{D}$ outputs 1 given the output of the challenger in Game t.

Game 0: This is the real game of SIM-RSO security regarding RsoPKE when the number of key pairs is ℓ. The detailed description is as follows.

1. First, the challenger generates ℓ key pairs $\left(\mathsf{Rso.ek}^{(k)}, \mathsf{Rso.dk}^{(k)}\right)$ $(k = 1, \ldots, \ell)$ as follows.

(a) The challenger generates $\left(\mathsf{ek}_{\alpha}^{(k)}, \mathsf{dk}_{\alpha}^{(k)}\right) \leftarrow \mathsf{KG}(1^{\lambda})$ for every $\alpha \in \{0,1\}$.

(b) The challenger generates $r^{(k)} \xleftarrow{\mathsf{r}} \{0,1\}$.

(c) The challenger sets $\mathsf{Rso.ek}^{(k)} := \left(\mathsf{ek}_{0}^{(k)}, \mathsf{ek}_{1}^{(k)}\right)$ and $\mathsf{Rso.dk}^{(k)} := \left(r^{(k)}, \mathsf{dk}_{r}^{(k)}\right)$.

The challenger sends $\left\{\mathsf{Rso.ek}^{(k)}\right\}_{k\in[\ell]}$ to $\mathcal{A}$.

2. $\mathcal{A}$ sends a message distribution Dist on $\{0,1\}^{\ell}$ to the challenger. The challenger generates $\left\{m^{(k)}\right\}_{k\in[\ell]} \leftarrow \mathsf{Dist}$, and computes $\mathsf{Rso.CT}^{(k)}$ for every $k \in [\ell]$ as follows.

   (a) For every $\alpha \in \{0,1\}$, the challenger computes $\mathsf{CT}_{\alpha}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{\alpha}^{(k)}, m^{(k)}\right)$.

   (b) The challenger sets $\mathsf{Rso.CT}^{(k)} := \left(\mathsf{CT}_{0}^{(k)}, \mathsf{CT}_{1}^{(k)}\right)$.

   The challenger sends $\left\{\mathsf{CT}^{(k)}\right\}_{k\in[\ell]}$ to $\mathcal{A}$.

3. $\mathcal{A}$ sends a subset $\mathcal{I}$ of $[\ell]$ to the challenger. The challenger sends $\left\{(\mathsf{Rso.dk}^{(k)}, m^{(k)})\right\}_{k\in\mathcal{I}}$ to $\mathcal{A}$.

4. $\mathcal{A}$ sends a string out to the challenger.

5. The challenger outputs $\mathsf{out}_{\mathsf{real}} := \left(\left\{m^{(k)}\right\}_{k\in[\ell]}, \mathsf{Dist}, \mathcal{I}, \mathsf{out}\right)$.

Game 1: Same as Game 0 except that for every $k \in [\ell]$, the challenger generates

$$\mathsf{CT}_{1-r^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{1-r^{(k)}}^{(k)}, 1-m^{(k)}\right) \quad .$$

We note that for every $k \in [\ell]$, the challenger generates $\mathsf{CT}_{r^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{r^{(k)}}^{(k)}, m^{(k)}\right)$ in both Games 0 and 1.

Secret keys $\left\{\mathsf{dk}_{1-r^{(k)}}^{(k)}\right\}_{k\in[\ell]}$ of PKE are not given to $\mathcal{A}$ regardless of which users $\mathcal{A}$ corrupts in both Games 0 and 1. Therefore, by using the security of PKE ℓ times, we can prove $|\Pr[\mathtt{T}_0] - \Pr[\mathtt{T}_1]| = \mathsf{negl}(\lambda)$.

Game 2: Same as Game 1 except that for every $k \in [\ell]$, the challenger uses $r^{(k)} \oplus m^{(k)}$ instead of $r^{(k)}$ as the random bit contained in the k-th RsoPKE's secret key $\mathsf{Rso.dk}^{(k)}$. We note that the challenger does not need $\left\{r^{(k)}\right\}_{k\in[\ell]}$ before generating $\left\{m^{(k)}\right\}_{k\in[\ell]}$. Thus, the transition from Games 1 to 2 makes sense, and $|\Pr[\mathtt{T}_2] - \Pr[\mathtt{T}_3]| = 0$ holds since $r^{(k)} \oplus m^{(k)}$ is distributed uniformly at random for every $k \in [\ell]$.

In Game 2, uncorrupted messages $\left\{m^{(k)}\right\}_{k\in[\ell]\setminus\mathcal{I}}$ are completely hidden from the view of $\mathcal{A}$. To verify the fact, we confirm that ciphertexts $\left\{\mathsf{Rso.CT}^{(k)}\right\}_{k\in[\ell]}$ are independent of $\left\{m^{(k)}\right\}_{k\in[\ell]}$.

For every $k \in [\ell]$, the challenger generates $\mathsf{Rso.CT}^{(k)} = \left(\mathsf{CT}_{0}^{(k)}, \mathsf{CT}_{1}^{(k)}\right)$ by generating

$$\mathsf{CT}_{r^{(k)}\oplus m^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{r^{(k)}\oplus m^{(k)}}^{(k)}, m^{(k)}\right) \quad ,$$
$$\mathsf{CT}_{1-r^{(k)}\oplus m^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{1-r^{(k)}\oplus m^{(k)}}^{(k)}, 1-m^{(k)}\right) \quad .$$

We see that, regardless of the value of $m^{(k)} \in \{0,1\}$, the challenger computes

$$\mathsf{CT}_{r^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{r^{(k)}}^{(k)}, 0\right) \quad ,$$
$$\mathsf{CT}_{1-r^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{1-r^{(k)}}^{(k)}, 1\right) \quad .$$

Therefore, we see that ciphertexts $\left\{\mathsf{Rso.CT}^{(k)}\right\}_{k\in[\ell]}$ are independent of $\left\{m^{(k)}\right\}_{k\in[\ell]}$ in Game 2.

Then, we construct a simulator $\mathcal{S}$ that perfectly simulate Game 2 for $\mathcal{A}$. The description of $\mathcal{S}$ is as follows.

1. On input 1^λ, $\mathcal{S}$ generates $\left(\mathsf{ek}_\alpha^{(k)}, \mathsf{dk}_\alpha^{(k)}\right) \leftarrow \mathsf{KG}(1^\lambda)$ for every $k \in [\ell]$ and $\alpha \in \{0,1\}$ and sets $\mathsf{Rso.ek}^{(k)} := \left(\mathsf{ek}_0^{(k)}, \mathsf{ek}_1^{(k)}\right)$ for every $k \in [\ell]$. $\mathcal{S}$ then sends $\left\{\mathsf{Rso.ek}^{(k)}\right\}_{k\in[\ell]}$ to $\mathcal{A}$.

2. When $\mathcal{A}$ outputs a message distribution Dist, $\mathcal{S}$ sends it to the challenger. Then, $\mathcal{S}$ computes $\mathsf{Rso.CT}^{(k)}$ for every $k \in [\ell]$ as follows.

   (a) $\mathcal{S}$ computes $r^{(k)} \xleftarrow{\mathsf{r}} \{0,1\}$.

   (b) $\mathcal{S}$ computes $\mathsf{CT}_{r^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{r^{(k)}}^{(k)}, 0\right)$ and $\mathsf{CT}_{1-r^{(k)}}^{(k)} \leftarrow \mathsf{Enc}\left(\mathsf{ek}_{1-r^{(k)}}^{(k)}, 1\right)$.

   (c) $\mathcal{S}$ sets $\mathsf{Rso.CT}^{(k)} := \left(\mathsf{CT}_0^{(k)}, \mathsf{CT}_1^{(k)}\right)$.

   $\mathcal{S}$ sends $\left\{\mathsf{Rso.CT}^{(k)}\right\}_{k\in[\ell]}$ to $\mathcal{A}$.

3. When $\mathcal{A}$ outputs a subset $\mathcal{I}$ of $[\ell]$, $\mathcal{S}$ sends it to the challenger, and gets $\left\{m^{(k)}\right\}_{k\in\mathcal{I}}$. $\mathcal{S}$ sets $\mathsf{Rso.dk}^{(k)} := \left(r^{(k)} \oplus m^{(k)}, \mathsf{dk}_{r^{(k)}\oplus m^{(k)}}^{(k)}\right)$ for every $k \in \mathcal{I}$, and sends $\left\{(\mathsf{Rso.dk}^{(k)}, m^{(k)})\right\}_{k\in\mathcal{I}}$ to $\mathcal{A}$.

4. When $\mathcal{A}$ outputs a string out, $\mathcal{S}$ outputs it.

$\mathcal{S}$ perfectly simulates Game 2 for $\mathcal{A}$. Therefore, we have

$$\mathsf{Adv}^{\mathsf{simrso}}_{\mathsf{RsoPKE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) = |\Pr[\mathtt{T}_0] - \Pr[\mathtt{T}_2]| \leq \sum_{t=0}^{2} |\Pr[\mathtt{T}_t] - \Pr[\mathtt{T}_{t+1}]| \quad . \tag{4}$$

From the above arguments, we see that each term of the right hand side of Inequality 4 is negligible in λ. Since the choice of $\mathcal{A}$,ℓ and $\mathcal{D}$ is arbitrary and the description of $\mathcal{S}$ does not depend on that of $\mathcal{D}$, we see that for any $\mathcal{A}$ and ℓ, there exists $\mathcal{S}$ such that for any $\mathcal{D}$ we have $\mathsf{Adv}^{\mathsf{simrso}}_{\mathsf{RsoPKE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) = \mathsf{negl}(\lambda)$. This means that RsoPKE is SIM-RSO secure. □ (**Theorem 8**)

The above construction of RsoPKE based on IND-CPA secure PKE is SIM-RSO secure in the sense of Definition 5 where an adversary can get only secret keys itself and not random coins for key generation on the corruption of users. We see that if an adversary can also get random coins for key generation, it seems difficult to prove that RsoPKE is SIM-RSO secure.

In this case, the adversary can get secret keys of PKE, $\left\{\mathsf{dk}_{1-r^{(k)}}^{(k)}\right\}_{k\in\mathcal{I}}$ in addition to $\left\{\mathsf{dk}_{r^{(k)}}^{(k)}\right\}_{k\in\mathcal{I}}$ and thus we cannot complete the transition from Games 0 to 1. To prove SIM-RSO security against the revelation of random coins for key generation, it seems that the underlying scheme needs to be *key simulatable* [DN00, HPW15] in some sense.