Circular Security Is Complete for KDM Security

Fuyuki Kitagawa[1] and Takahiro Matsuda[2]

[1] NTT Secure Platform Laboratories, Tokyo, Japan, fuyuki.kitagawa.yh@hco.ntt.co.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan, t-matsuda@aist.go.jp

Abstract

Circular security is the most elementary form of key-dependent message (KDM) security, which allows us to securely encrypt only a copy of secret key bits. In this work, we show that circular security is *complete* for KDM security in the sense that an encryption scheme satisfying this security notion can be transformed into one satisfying KDM security with respect to all functions computable by a-priori bounded-size circuits (bounded-KDM security). This result holds in the presence of any number of keys and in any of secret-key/public-key and CPA/CCA settings. Such a completeness result was previously shown by Applebaum (EUROCRYPT 2011) for KDM security with respect to projection functions (projection-KDM security) that allows us to securely encrypt both a copy and a negation of secret key bits.

Besides amplifying the strength of KDM security, our transformation in fact can start from an encryption scheme satisfying circular security against *CPA* attacks and results in one satisfying bounded-KDM security against *CCA* attacks. This result improves the recent result by Kitagawa and Matsuda (TCC 2019) showing a CPA-to-CCA transformation for KDM secure public-key encryption schemes.

Keywords: key-dependent message security, circular security, chosen ciphertext security

Contents

1 Introduction

1.1 Background

Key-dependent message (KDM) security, introduced by Black, Rogaway, and Shrimpton [BRS03], guarantees confidentiality of communication even if an adversary can get a ciphertext of secret keys. This notion was formulated in order to capture situations where there could be correlations between secret keys and messages to be encrypted. Although it seems that such situations only arise from bugs or errors, it turned out that they naturally occur in natural usage scenarios of encryption schemes such as hard-disc encryption [BHHO08], anonymous credentials [CL02], and formal methods [ABHS05]. Moreover, until today, a number of works have shown that KDM security is useful when constructing various cryptographic primitives including fully homomorphic encryption (FHE) [Gen09], non-interactive zero-knowledge (NIZK) proofs/arguments [CCRR18, CCH+19, LQR+19, KM19], homomorphic secret sharing [BKS19], and chosen ciphertext secure encryption schemes and trapdoor functions [HK15, KMT19].

KDM security is defined with respect to a function family $\mathcal{F}$. Informally, a public-key encryption (PKE) scheme is said to be $\mathcal{F}$-KDM$^{(n)}$ secure if confidentiality of messages is protected even when an adversary can see a ciphertext of $f(\mathsf{sk}_1, \cdots, \mathsf{sk}_n)$ under the s-th public key for any $f \in \mathcal{F}$ and $s \in \{1, \cdots, n\}$, where n denotes the number of keys. Also, KDM security is considered in both the chosen plaintext attack (CPA) and chosen ciphertext attack (CCA) settings.

Completeness of Projection-KDM Security. KDM security with respect to the family of projection functions (projection-KDM security) is one of the most widely studied notions. A projection function is an elementary function in which each output bit depends on at most a single bit of an input. Therefore, roughly speaking, projection-KDM security only guarantees that an encryption scheme can securely encrypt a copy and a negation of secret key bits.

Although this security notion looks somewhat weak at first glance, Applebaum [App11] showed that it is *complete* for KDM security in the sense that we can construct an encryption scheme satisfying KDM security with respect to all functions computable by a-priori bounded-size circuits (bounded-KDM security) based on one satisfying projection-KDM security. The completeness of projection-KDM security in [App11] has generality in the sense that it is insensitive to the exact setting of KDM security. More specifically, a projection-KDM secure encryption scheme can be transformed into a bounded-KDM secure one for any number of keys and in any of secret-key/public-key and CPA/CCA settings.

Moreover, recent works [KMT19, KM19, LQR+19] also showed the power and usefulness of projection-KDM secure encryption schemes for achieving other security notions and constructing other primitives. Specifically, Kitagawa, Matsuda, and Tanaka [KMT19] showed that projection-KDM secure PKE implies IND-CCA secure PKE, and Kitagawa and Matsuda [KM19] and Lombardi, Quach, Rothblum, Wichs, and Wu [LQR+19] independently showed that it implies a reusable designated-verifier NIZK argument system for any NP language.

Completeness of Circular Security? The focus in this work is on *circular security*, which is another elementary form of KDM security that has been widely studied from both the positive side [CL02, Gen09, HK15, CCH+19] and the negative side [ABBC10, CGH12, Rot13, KRW15, GKW17, HK17]. Circular security is a weaker security notion compared to even projection-KDM security since circular security allows us to securely encrypt only a copy of secret key bits.[1] In this work, we clarify whether this most elementary form of KDM security is also complete in the above sense or not.

Let us explain the motivations for studying the completeness of circular security for KDM security. From the practical aspect, although it is an elementally form of KDM security, it is known to be sufficient for many practical applications of KDM security such as anonymous credentials, formal methods, and FHE listed above. Thus, studying circular security is expected to give us insights on these applications. From the theoretical aspect, it has impacts on the study of public-key cryptography since several recent works [KMT19, KM19, LQR$^+$19] showed that a projection-KDM secure encryption scheme is useful as a building block for constructing two important and central primitives of IND-CCA secure PKE and reusable designated-verifier NIZK argument systems, among which we will expand explanations on the former in the paragraph below. Furthermore, studying whether the ability to securely encrypt only a copy of secret key bits has a similar power to that to securely encrypt both a copy and a negation of secret key bits at the same time, is well-motivated from the viewpoint of "negation-complexity" of cryptographic primitives [GI12, GMOR15]. For example, Goldreich and Izsak [GI12] showed that a one-way function can be computed by a monotone circuit and yet a pseudorandom generator cannot. It is interesting to investigate whether such a barrier exists in the context of KDM security.

Implications to the Study of CPA vs CCA. The question whether an IND-CCA secure PKE scheme can be constructed from an IND-CPA secure one has been standing as a major open question in cryptography. The completeness of circular security for KDM security also has a deep connection to this question: Hajiabadi and Kapron [HK15] tackled the above question, and they built an IND-CCA secure PKE scheme based on a PKE scheme satisfying circular security and a randomness re-usability property called reproducibility [BBS03]. Also, Kitagawa et al. [KMT19] showed that an IND-CCA secure PKE scheme can be constructed from a projection KDM secure PKE scheme.

The above two results surely made a progress on the study of CCA security versus CPA security by showing that an IND-CCA secure PKE scheme can be constructed from a PKE scheme satisfying only security notions against "CPA" (i.e. no decryption queries). Here, the above results are incomparable since the former result requires a structural property while the latter requires projection-KDM security that is stronger than circular security for the building block scheme. It is an open question whether we can construct an IND-CCA secure PKE scheme based on a PKE scheme satisfying only circular security without requiring any structural property for the building block scheme. We see that this question is solved affirmatively if we can prove the completeness of circular security for KDM security by combining it with the previous results [KMT19, LQR$^+$19, KM19].

1.2 Our Results

In this work, we show that circular security is complete in the sense that an encryption scheme satisfying this security notion can be transformed into a bounded-KDM secure one. In this work, unless stated otherwise, circular security indicates a security notion that guarantees that an encryption scheme can securely encrypt a copy of each of secret key bits separately. We show that this result has the same level of generality as the completeness of projection-KDM security shown by Applebaum [App11]. Namely, we obtain the following theorem. Below, we denote circular security against CPA under n key pairs as CIRC$^{(n)}$ security.

[1] Note that the phrase "circular security" is sometimes used to mean a (similar but) different notion, such as security when encrypting key cycles.

Theorem 1 (Informal) *If there exists a CIRC$^{(n)}$ secure PKE (resp. SKE) scheme, then there exists a bounded-KDM$^{(n)}$-CCA secure PKE (resp. SKE) scheme for any number of keys n.*

Note that the above theorem implies the completeness of circular security in both the CPA and CCA settings at the same time since we start with a scheme satisfying circular security against *CPA* and obtain a scheme satisfying bounded-KDM security against *CCA*. We obtain Theorem 1 in the following way.

How to Obtain Completeness in the Public-Key Setting. We first focus on the case where there is only a single key pair. In Section 4, as our main technical result, we show that an encryption primitive called *targeted encryption (TE)*, formalized by Barak, Haitner, Hofheinz, and Ishai [BHHI10], can be constructed from the combination of a CIRC$^{(1)}$ secure SKE scheme and an IND-CPA secure PKE scheme. Since both of the building blocks are implied by CIRC$^{(1)}$ secure PKE, and a TE scheme in turn can be transformed into a bounded-KDM$^{(1)}$-CPA secure PKE scheme as shown by Barak et al. [BHHI10], this result implies that a CIRC$^{(1)}$ secure PKE scheme can be transformed into a bounded-KDM$^{(1)}$-CPA secure PKE scheme. Once we construct a bounded-KDM$^{(1)}$-CPA secure PKE scheme, by combining with the result by Kitagawa and Matsuda [KM19], we can transform it into a bounded-KDM$^{(1)}$-CCA secure PKE scheme, which is stated in Section 5.

We then turn our attention to the case where there are multiple key pairs. Similarly to the above, we can construct a bounded-KDM$^{(n)}$-CPA secure PKE scheme based on a CIRC$^{(n)}$ secure one for any n through a primitive called augmented TE [BHHI10] that is an extension of TE. However, in the case of multiple key pairs, there is no transformation from a KDM-CPA secure PKE scheme to a KDM-CCA secure one regardless of the function family with respect to which we consider KDM security. Thus, in this case, we cannot easily carry the result in the CPA setting to that in the CCA setting.

To overcome the above problem, in Section 6, we first introduce a primitive that we call *conformed TE (CTE)*. CTE is an extension of TE (with several similarities to augmented TE of Barak et al. [BHHI10]) that is conformed to the construction of a KDM-CCA secure PKE scheme in the presence of multiple key pairs. We then construct a CTE scheme based on a CIRC$^{(n)}$ secure SKE scheme and an IND-CPA secure PKE scheme. Finally, in Section 7, we construct a bounded-KDM$^{(n)}$-CCA secure PKE scheme from a CTE scheme, a garbling scheme, an IND-CCA secure PKE scheme, and a (reusable) DV-NIZK argument system. The last two components are implied by a circular secure PKE scheme from our result in the case of a single key pair and the results by Kitagawa and Matsuda [KM19] and Lombardi et al. [LQR$^+$19]. This implies that circular security is complete in both the CPA and CCA settings even when there are multiple key pairs. Note that this result improves that of Kitagawa and Matsuda [KM19] in the following two aspects: Not only our construction can start from a circular secure PKE scheme, but also it works in the case of multiple key pairs.

How to Obtain Completeness in the Secret-Key Setting. From the result shown by Backes, Pfitzmann, and Scedrov [BPS07], we can transform a bounded-KDM$^{(n)}$-CPA secure SKE scheme into a bounded-KDM$^{(n)}$-CCA secure one for any n. Thus, in the secret-key setting, all we have to do is to construct a bounded-KDM$^{(n)}$-CPA secure SKE scheme based on a CIRC$^{(n)}$ secure one. Similarly to the public-key setting, this is possible via the secret-key version of TE for the case of a single key pair and via the secret-key version of augmented TE for the case of multiple key pairs. These constructions are almost the same as the public-key counterparts, and thus we omit their formal descriptions in the paper. (In Section 2, this construction is outlined since we explain a technical overview of our results using the secret-key version of TE.)

Implications of Our Completeness Result. We obtain the following additional results: We show that the construction of the bounded-KDM$^{(1)}$-CPA secure PKE scheme mentioned above, is in fact a fully black-box construction [RTV04] if we restrict the function family to projection functions. Thus, by combining this fact with the result by Kitagawa et al. [KMT19], we obtain a fully black-box construction of an IND-CCA secure PKE scheme from a circular secure one.[2] Moreover, by simply combining Theorem 1 with the result independently achieved by Kitagawa and Matsuda [KM19] and Lombardi et al. [LQR$^+$19], we see that a reusable DV-NIZK argument system can also be constructed from a circular secure PKE scheme.

1.3 Paper Organization

The rest of the paper is organized as follows: In Section 2, we give a technical overview of our results. In Section 3 (and Section A), we review definitions of cryptographic primitives. In Section 4, we present our construction of TE. In Section 5, we show several implications of our TE scheme, and in particular the completeness of circular security for the single-key setting. In Section 6, we introduce CTE and present its construction. Finally, in Section 7, we present the completeness of circular security in the multi-key setting using CTE.

2 Technical Overview

In this section, we provide a technical overview of our results. Our main technical contribution is to show that we can realize TE (and conformed TE) based only on a circular secure encryption scheme in a completely generic way. Thus, in this overview, we mainly focus on this part after briefly explaining how to construct a bounded-KDM secure scheme based on TE. For simplicity, we explain our ideas in this part by showing how to construct the secret-key version of a TE scheme based only on a CIRC$^{(1)}$ secure SKE scheme. In the following, for a natural number n, we let $[n]$ denote the set $\{1, \dots, n\}$.

2.1 Secret-Key TE

We first introduce the secret-key version of TE [BHHI10]. A secret-key TE scheme consists of the three algorithms TKG, TEnc, and TDec.[3] Similarly to an ordinary SKE scheme, TKG is given a security parameter and outputs a secret key sk. We let ℓ_{sk} denote the secret key length. On the other hand, TEnc and TDec have a functionality of a somewhat special form. As we will soon see below, they are optimized for encrypting labels of garbled circuits [Yao86]. In addition to the secret key sk, TEnc is given an index $i \in [\ell_{\mathsf{sk}}]$ and a pair of messages (X_0, X_1), and outputs a ciphertext as $\mathsf{ct} \leftarrow \mathsf{TEnc}(\mathsf{sk}, i, X_0, X_1)$. Correspondingly, given the secret key sk, the index $i \in [\ell_{\mathsf{sk}}]$, and the ciphertext ct, TDec outputs (only) $X_{\mathsf{sk}[i]}$, where $\mathsf{sk}[i]$ denotes the i-th bit of sk. (Thus, it is similar to an oblivious transfer.) For TE, we consider two security notions: *security against the receiver* and *security against outsiders*. Security against the receiver ensures that ct hides the information of $X_{1 \oplus \mathsf{sk}[i]}$ even against the receiver who holds sk. Security against outsiders ensures that ct hides both X_0 and X_1 against adversaries who do not hold sk.[4]

[2] Note that this result does not simply follow from Theorem 1 since the construction of KDM-CCA secure PKE used to show it is non-black-box due to the use of a DV-NIZK argument.

[3] Here, we adopt the syntax that is slightly different from the one we use in the subsequent sections, in that the latter allows to encrypt X_v for each $v \in \{0, 1\}$ separately. The syntax used here makes the following explanations easier and cleaner. For the formal definition, see Section 3.3.

[4] Hereafter, we refer to adversaries that do not hold the secret key as outsiders.

Bounded-KDM$^{(1)}$-CPA Security via TE. As shown by Barak et al. [BHHI10], we can construct a bounded-KDM$^{(1)}$-CPA secure SKE scheme based on a secret-key TE scheme by using garbled circuits.[5] The construction is fairly simple. The secret key of the resulting SKE scheme is that of the underlying secret-key TE scheme itself. When encrypting a message m, we first garble an ℓ_{sk}-bit-input constant function $\mathsf{C_m}$ that outputs m for any input. This results in a single garbled circuit $\widetilde{\mathsf{C}}$ and $2\ell_{\mathsf{sk}}$ labels $(\mathsf{lab}_{i,v})_{i\in[\ell_{\mathsf{sk}}],v\in\{0,1\}}$. Then, for every index $i \in [\ell_{\mathsf{sk}}]$, we encrypt the pair of labels $(\mathsf{lab}_{i,0}, \mathsf{lab}_{i,1})$ under the index i into ct_i using TEnc. The resulting ciphertext for the SKE scheme consists of $\widetilde{\mathsf{C}}$ and $(\mathsf{ct}_i)_{i\in[\ell_{\mathsf{sk}}]}$. When decrypting this ciphertext, we first obtain $(\mathsf{lab}_{i,\mathsf{sk}[i]})_{i\in[\ell_{\mathsf{sk}}]}$ from $(\mathsf{ct}_i)_{i\in[\ell_{\mathsf{sk}}]}$ by using TDec with sk. Then, we evaluate the garbled circuit $\widetilde{\mathsf{C}}$ with these labels. This results in m from the correctness of the garbling scheme.

We can prove that the above construction is bounded-KDM$^{(1)}$-CPA secure. In a high level, we can generate a simulated encryption of $f(\mathsf{sk})$ without using sk itself that is indistinguishable from a real ciphertext based on the security against the receiver of the underlying secret-key TE scheme and the security of the underlying garbling scheme, where f is a function queried by an adversary as a KDM-encryption query. We then finish the security proof by relying on the security against outsiders of the secret-key TE scheme. For more details, see [BHHI10].

2.2 Secret-Key TE Based on Circular Secure SKE

Below, we explain how to construct a secret-key TE scheme based on a CIRC$^{(1)}$ secure SKE scheme. We first show that a secret-key TE scheme can be naturally realized from a projection-KDM$^{(1)}$ secure SKE scheme. We then show how to weaken the starting point to a CIRC$^{(1)}$ secure SKE scheme.

Secret-Key TE Based on Projection-KDM Secure SKE. Consider the following naive way to realize a secret-key TE scheme based on an SKE scheme SKE. A secret key sk of SKE is used as that of the secret-key TE scheme. When encrypting (X_0, X_1) under an index $i \in [\ell_{\mathsf{sk}}]$, we just encrypt $X_{\mathsf{sk}[i]}$ into ct by using the encryption algorithm Enc of SKE with the secret key sk. We call this naive realization Naive. Naive clearly satisfies security against the receiver since ct is independent of $X_{1\oplus\mathsf{sk}[i]}$. However, it is not clear whether we can prove the security against outsiders of Naive if we only assume that SKE satisfies IND-CPA security. This is because the encrypted message $X_{\mathsf{sk}[i]}$ is dependent on the secret key sk. On the other hand, we can prove the security against outsiders of Naive if SKE satisfies projection-KDM$^{(1)}$-CPA security which allows us to securely encrypt both *a copy and a negation* of $\mathsf{sk}[i]$.

To see this in detail, we suppose that $X_{\mathsf{sk}[i]}$ is encrypted by SKE in a bit-by-bit manner, and its length is μ. We denote the j-th bit of X_0 (resp. X_1) by $X_0[j]$ (resp. $X_1[j]$). We can classify the indices in $[\mu]$ into the following four types:

Type 1: $j \in [\mu]$ such that $X_0[j] = X_1[j] = 0$.

Type 2: $j \in [\mu]$ such that $X_0[j] = X_1[j] = 1$.

Type 3: $j \in [\mu]$ such that $X_0[j] = 0$ and $X_1[j] = 1$.

Type 4: $j \in [\mu]$ such that $X_0[j] = 1$ and $X_1[j] = 0$.

We have to generate the following ciphertexts of SKE for each type to encrypt $X_{\mathsf{sk}[i]}$:

- For j of Type 1, we have to generate $\mathsf{Enc}(\mathsf{sk}, 0)$ regardless of the value of $\mathsf{sk}[i]$.

[5] Note that the actual transformation shown by Barak et al. is in the public-key setting. Also, the following explanations assume that the reader is familiar with a garbling scheme. See Section A.3 for its formal definition.

- For j of Type 2, we have to generate $\mathsf{Enc}(\mathsf{sk}, 1)$ regardless of the value of $\mathsf{sk}[i]$.
- For j of Type 3, we have to generate $\mathsf{Enc}(\mathsf{sk}, \mathsf{sk}[i])$, that is, an encryption of a *copy* of $\mathsf{sk}[i]$.
- For j of Type 4, we have to generate $\mathsf{Enc}(\mathsf{sk}, 1 \oplus \mathsf{sk}[i])$, that is, an encryption of a *negation* of $\mathsf{sk}[i]$.

Namely, when some bit of X_0 is 0 and the corresponding bit of X_1 is 1, we have to generate an encryption of a copy of $\mathsf{sk}[i]$. Similarly, when some bit of X_0 is 1 and the corresponding bit of X_1 is 0, we have to generate an encryption of a negation of $\mathsf{sk}[i]$. However, if SKE is projection-KDM$^{(1)}$-CPA secure, then $X_{\mathsf{sk}[i]}$ is hidden from outsiders. Since $X_{1 \oplus \mathsf{sk}[i]}$ is completely hidden (even against the legitimate receiver), Naive satisfies security against outsiders based on the projection-KDM$^{(1)}$-CPA security of SKE.

Replacing Projection-KDM-CPA Secure SKE with Circular Secure SKE. We now try to realize a secret-key TE scheme based on a circular secure (CIRC$^{(1)}$ secure) SKE scheme. Recall that CIRC$^{(1)}$ security allows us to securely encrypt only a copy of secret key bits. Thus, as the first attempt to avoid encrypting negations of secret key bits, we modify the above construction Naive into the following construction that we call Naive^*.

In Naive^*, when encrypting (X_0, X_1) under an index $i \in [\ell_{\mathsf{sk}}]$, we basically encrypt $X_{\mathsf{sk}[i]}$ in a bit-by-bit manner in the same way as Naive. However, for indices $j \in [\mu]$ of Type 4, we replace the ciphertext of SKE with the special symbol `flip`. When receiving the symbol `flip` instead of the j-th ciphertext, the receiver sets the value of $X_{\mathsf{sk}[i]}[j]$ as $1 \oplus \mathsf{sk}[i]$. This is possible since the receiver has sk and knows the value of $\mathsf{sk}[i]$. Thus, if we modify the construction in this way, the receiver holding sk can obtain the entire bits of $X_{\mathsf{sk}[i]}$ similarly to Naive.

In Naive^*, we now need to generate encryptions of only a copy of $\mathsf{sk}[i]$ and not those of a negation of $\mathsf{sk}[i]$. However, we cannot prove that Naive^* satisfies the two security notions of TE (security against the receiver/outsiders) based on the CIRC$^{(1)}$ security of SKE. For example, considering security against outsiders, X_0 and X_1 are partially leaked to outsiders because of the use of the symbol `flip`. Concretely, outsiders can know that $X_0[j] = 1$ and $X_1[j] = 0$ for the indices j of Type 4. A similar problem lies in the argument on security against the receiver. Concretely, the receiver holding sk can know $X_{1 \oplus \mathsf{sk}[i]}[j]$ for the indices j of Type 4 and either one of Type 1 or 2 depending on the value of $\mathsf{sk}[i]$. The reason why $X_{1 \oplus \mathsf{sk}[i]}[j]$ for the indices j of Type 4 are leaked to the receiver is clear. The reason why those for the indices j of Type 1 or 2 are leaked to the receiver is as follows. For example, when $\mathsf{sk}[i] = 0$, the receiver finds that the value of $X_{1 \oplus \mathsf{sk}[i]}[j]$ is 1 for j of Type 2 from the fact that the decrypted message from the j-th ciphertext is 1 but the symbol `flip` was not sent for this j.

To summarize, if SKE is CIRC$^{(1)}$ secure, the following properties hold for Naive^*: $X_0[j]$ and $X_1[j]$ for the indices j of Type 1, 2, and 3 are hidden but those for the indices j of Type 4 are leaked to outsiders. Also, $X_{1 \oplus \mathsf{sk}[i]}[j]$ for the indices j of Type 3 and either one of Type 1 or 2 are hidden but the remaining parts are leaked to the receiver holding sk.

Transforming into a Full-Fledged Secret-Key TE Scheme. A natural question here is whether the above Naive^* is useful or not. We show that by using a *leakage-resilient* SKE scheme lrSKE, we can transform Naive^* into an ordinary secret-key TE scheme sTE. As we will explain later, the type of leakage-resilience that lrSKE should satisfy is weak, and any IND-CPA secure SKE scheme can be transformed into one satisfying it. Thanks to this transformation, we can realize a secret-key TE scheme based only on a CIRC$^{(1)}$ secure SKE scheme.

The description of sTE is as follows. The secret key sk of sTE is that of Naive^* itself. When encrypting (X_0, X_1) under the index $i \in [\ell_{\mathsf{sk}}]$, we first generate two keys lrk_0 and lrk_1 of lrSKE.

Then, we encrypt X_0 and X_1 into lrct_0 and lrct_1 by using lrSKE with the keys lrk_0 and lrk_1, respectively. Moreover, we encrypt $(\mathsf{lrk}_0, \mathsf{lrk}_1)$ into ct by using Naive^* with the key sk. The resulting ciphertext of sTE is $(\mathsf{lrct}_0, \mathsf{lrct}_1, \mathsf{ct})$. When decrypting this ciphertext, we first obtain $\mathsf{lrk}_{\mathsf{sk}[i]}$ from ct by using Naive^* with the key sk. We then obtain $X_{\mathsf{sk}[i]}$ by decrypting $\mathsf{ct}_{\mathsf{sk}[i]}$ using lrSKE with the key $\mathsf{lrk}_{\mathsf{sk}[i]}$.

We now argue that sTE satisfies (full-fledged) security against the receiver/outsiders. Without loss of generality, we assume that lrk_0 and lrk_1 are uniformly random n-bit strings. We define Type 1, 2, 3, and 4 for indices in $[n]$ as before using lrk_0 and lrk_1 instead of X_0 and X_1. Since lrk_0 and lrk_1 are chosen uniformly at random, these four types appear equally likely. In this case, ct hides expectedly a $1/2$-fraction of bits of $\mathsf{lrk}_{1\oplus\mathsf{sk}[i]}$ against the receiver holding sk. Also, ct hides expectedly a $3/4$-fraction of bits of each of lrk_0 and lrk_1 against outsiders. Thus, if lrSKE is resilient against both forms of secret key leakage, sTE satisfies both security against the receiver and security against outsiders.

Fortunately, the leakage-resilience that lrSKE should satisfy in the above argument is weak. The amount of leakage is (expectedly) only a constant fraction. In addition, more importantly, which bits of the secret key are leaked is determined completely at random from the fact that Type 1, 2, 3, and 4 appear uniformly at random, out of the control of adversaries. Leakage-resilience against such secret key leakage is weak, and we can transform any IND-CPA secure SKE scheme into one satisfying it by using the leftover hash lemma [HILL99, DRS04]. From this fact, sTE can be realized from a CIRC$^{(1)}$ secure SKE scheme.

2.3 Towards the Completeness in the Public-Key Setting

As we mentioned earlier, in the actual technical sections, we deal with the public-key setting. Namely, we prove Theorem 1 in the PKE setting. We finally explain how to prove it with the techniques explained so for.

Single-Key Setting. We first construct a (public-key) TE scheme based on a CIRC$^{(1)}$ secure SKE scheme and an IND-CPA secure PKE scheme both of which are implied by a CIRC$^{(1)}$ secure PKE scheme. This construction is almost the same as that of sTE above except that we use a leakage-resilient PKE scheme instead of a leakage-resilient SKE scheme. By combining this transformation with the previous results [BHHI10, KM19], we can obtain Theorem 1 in the PKE setting for the number of key pairs $n = 1$.

Multi-key Setting. We then move on to the case of multiple key pairs. As mentioned before, for achieving the completeness in this setting, we introduce an extended version of TE that we call conformed TE (CTE). CTE is conformed to construct KDM$^{(n)}$-CCA secure PKE schemes for $n > 1$. Roughly, CTE is TE that satisfies the following two additional properties.

- When generating a public/secret key pair, it additionally generates a trapdoor that enables us to recover both a "0-side" message X_0 and a "1-side" message X_1 from a ciphertext encrypting (X_0, X_1). (Recall that in ordinary TE, the receiver can recover only one of them even having the secret key.)

- A CTE scheme has additional (untargeted and secret-key) encryption/decryption algorithms, and a ciphertext generated by the additional encryption algorithm is indistinguishable even under the existence of the above trapdoor and encryptions of a "key cycle" generated by the additional encryption algorithm. Encryptions of a key cycle are ciphertexts such that the s-th ciphertext is an encryption of the $(s \bmod n) + 1$-th secret key under the s-th secret key when there are n keys. We call this property *special weak circular security.*

We remark that a TE scheme satisfying only the second property is almost the same as augmented TE introduced by Barak et al. [BHHI10] to construct a bounded-KDM$^{(n)}$-CPA secure PKE scheme for $n > 1$. Roughly speaking, when constructing a KDM-CCA secure PKE scheme, the first property mainly plays its role to deal with decryption queries, and the second property plays its role to deal with multiple key pairs. For the details of the formalization of CTE as well as its relation to augmented TE, see Section 6.

We construct a CTE scheme based on a CIRC$^{(n)}$ secure SKE scheme and an IND-CPA secure PKE scheme. Basically, this construction is again an extension of sTE in which a leakage-resilient PKE scheme is used instead of a leakage-resilient SKE scheme. The trapdoor of the construction consists of secret keys of the leakage-resilient PKE scheme. Also, the special weak circular security of it is proved based on the CIRC$^{(n)}$ security of the underlying SKE scheme.

We finish the proof of Theorem 1 in the public-key setting for $n > 1$ by constructing a bounded-KDM$^{(n)}$-CCA secure PKE scheme from the combination of the following four building blocks: (1) a CTE scheme, (2) an IND-CCA secure PKE scheme, (3) a garbling scheme for circuits, and (4) a reusable DV-NIZK argument system for NP languages. As we already explained, by Theorem 1 for $n = 1$ and results by [KM19, LQR$^+$19], an IND-CCA secure PKE scheme and a reusable DV-NIZK argument system can be constructed from the combination of an IND-CPA secure PKE scheme and a CIRC$^{(1)}$ secure SKE scheme. Also, a garbling scheme for circuits can be constructed from a one-way function. Thus, all the building blocks can be based on the combination of an IND-CPA secure PKE scheme and a CIRC$^{(n)}$ secure SKE scheme. This completes the proof of Theorem 1 in the PKE setting for $n > 1$.

Our construction of bounded-KDM-CCA secure PKE in the multi-key setting can be seen as combining the construction ideas from the two existing constructions: the construction of KDM-CPA secure PKE in the multi-key setting based on an augmented TE scheme by Barak et al. [BHHI10], and the construction of KDM-CCA secure PKE in the single key setting based on an IND-CPA secure PKE scheme and a projection-KDM secure SKE scheme by Kitagawa and Matsuda [KM19]. However, a simple combination of each of the techniques from [BHHI10, KM19] as it is is not sufficient. We bridge the gap with the properties of the CTE scheme. For the details, see Section 7.

3 Preliminaries

In this section, we review the basic notation, and the definitions as well as existing results for encryption primitives treated in this paper. We give the formal definitions for ordinary IND-CCA/CPA security, a (reusable) DV-NIZK argument system, and a garbling scheme for circuits in Section A.

3.1 Basic Notation and Notions

For $n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\}$. For strings x and y, "$|x|$" denotes the bit-length of x, "$x[i]$" (with $i \in [|x|]$) denotes the i-th bit of x, and "$(x \overset{?}{=} y)$" is the operation that returns 1 if $x = y$ and 0 otherwise. For a discrete finite set S, "$|S|$" denotes its size, and "$x \xleftarrow{\mathsf{r}} S$" denotes choosing an element x uniformly at random from S. For a (probabilistic) algorithm A, "$y \leftarrow \mathsf{A}(x)$" denotes assigning to y the output of $\mathcal{A}$ on input x, and if we need to specify a randomness r used in A, we write "$y \leftarrow \mathsf{A}(x; r)$". If furthermore $\mathcal{O}$ is a function or an algorithm, then "$\mathsf{A}^{\mathcal{O}}$" means that A has oracle access to $\mathcal{O}$. A function $\epsilon(\lambda) : \mathbb{N} \to [0, 1]$ is said to be *negligible* if $\epsilon(\lambda) = \lambda^{-\omega(1)}$. We write $\epsilon(\lambda) = \mathsf{negl}(\lambda)$ to mean ϵ being negligible. The character "λ" always denotes a security parameter. "PPT" stands for *probabilistic polynomial time.*

For a distribution $\mathcal{X}$, the *min-entropy of* $\mathcal{X}$ is defined by $\mathbf{H}_\infty(\mathcal{X}) := -\log_2(\ \max_x \Pr[\mathcal{X} = x]\ )$. For distributions $\mathcal{X}$ and $\mathcal{Y}$ (forming a joint distribution), the *average min-entropy of* $\mathcal{X}$ *given* $\mathcal{Y}$ is defined by $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y}) := -\log_2(\ \mathbf{E}_{y\leftarrow\mathcal{Y}}[\max_x \Pr[\mathcal{X} = x|\mathcal{Y} = y]\ ]\ )$.

3.2 Public-Key and Secret-Key Encryption

Here, we recall the definitions for public-key and secret-key encryption schemes. We first introduce the definitions for PKE, and then briefly mention how to recover those for SKE.

Syntax of Public-Key Encryption. A PKE scheme PKE consists of the three PPT algorithms $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$:[6]

- KG is the key generation algorithm that takes 1^λ as input, and outputs a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- Enc is the encryption algorithm that takes a public key pk and a message m as input, and outputs a ciphertext ct.
- Dec is the (deterministic) decryption algorithm that takes a public key pk, a secret key sk, and a ciphertext ct as input, and outputs a message m or the invalid symbol $\perp$.

A PKE scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is said to be *correct* if for all $\lambda \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$, and m, we have $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) = \mathsf{m}$.

We refer to a PKE scheme whose message space is 1-bit as a *bit-PKE* scheme.

Simple Key Generation. We say that a PKE scheme has *simple key generation* if its key generation algorithm KG first picks a secret key sk uniformly at random (from some prescribed secret key space) and then computes a public key pk from sk. For PKE with simple key generation, we slightly abuse the notation and simply write $\mathsf{pk} \leftarrow \mathsf{KG}(\mathsf{sk})$ to denote this computation. Any IND-CPA/IND-CCA secure PKE scheme can be viewed as one with simple key generation by just regarding a randomness used in the key generation algorithm as sk.

Weak Noisy-Leakage-Resilience. We will use a PKE scheme that satisfies *weak noisy-leakage-resilience* (against CPA), formalized by Naor and Segev [NS09]. In the weak "noisy" leakage setting, an adversary's leakage function f must be chosen before seeing pk, and must satisfy the condition that the average min-entropy of sk given $f(\mathsf{sk})$ is greater than a predetermined lower bound.

Formally, for a PKE scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, a polynomial $L = L(\lambda)$, and an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, consider the experiment described in Figure 1. In the experiment, $\mathcal{A}$ is required to be *L-noisy-leakage-respecting*, which requires that $L \geq \mathbf{H}_\infty(\mathsf{sk}) - \widetilde{\mathbf{H}}_\infty(\mathsf{sk}|f(\mathsf{sk}), \mathsf{st})$ hold.

Definition 1 (Weak Noisy-Leakage-Resilience) *Let $L = L(\lambda)$ be a polynomial. We say that a PKE scheme* PKE *is* weakly L-noisy-leakage-resilient *if for all PPT L-noisy-leakage-respecting adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, we have* $\mathsf{Adv}^{\mathsf{wlr}}_{\mathsf{PKE},\mathcal{A},L}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathsf{wlr}}_{\mathsf{PKE},\mathcal{A},L}(\lambda) = 1] - 1/2| = \mathsf{negl}(\lambda)$.

```
$\mathsf{Expt}^{\mathsf{wlr}}_{\mathsf{PKE},\mathcal{A},L}(\lambda)$ :
    $(f, \mathsf{st}) \leftarrow \mathcal{A}_0(1^\lambda)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$
    $b \xleftarrow{\mathsf{r}} \{0,1\}$
    $b' \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{Enc}}(\cdot,\cdot)}(\mathsf{pk}, f(\mathsf{sk}), \mathsf{st})$
    Return $(b' \overset{?}{=} b)$.

$\mathcal{O}_{\mathsf{Enc}}(\mathsf{m}_0, \mathsf{m}_1)$ :    // $|\mathsf{m}_0| = |\mathsf{m}_1|$
    Return $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$.
```

Figure 1: The weak noisy-leakage-resilience experiment for PKE. In the experiment, it is required that $L \geq \mathbf{H}_\infty(\mathsf{sk}) - \widetilde{\mathbf{H}}_\infty(\mathsf{sk}|f(\mathsf{sk}), \mathsf{st})$.

Any IND-CPA secure PKE scheme can be straightforwardly converted into a weakly noisy-leakage-resilient one by using the leftover hash lemma [HILL99, DRS04]. In fact, Naor and Segev [NS09] showed this fact for the case of weak "bounded" leakage-resilience (where the output-length of a leakage function is bounded), and it is easy to see that their proof carries over to the case of weak noisy-leakage-resilience. Furthermore, this conversion is fully black-box and preserves the simple key generation property. (It works for SKE as well.) Since we will use this fact in Section 5, for completeness, we provide a formal proof of this fact in Section B.

Lemma 1 *Assume that there exists an IND-CPA secure PKE scheme with simple key generation whose secret key length is $\ell_{\mathsf{sk}} = \ell_{\mathsf{sk}}(\lambda)$. Then, for any polynomials $L = L(\lambda)$ and $\ell'_{\mathsf{sk}} = \ell'_{\mathsf{sk}}(\lambda)$ satisfying $\ell'_{\mathsf{sk}} - (L + \ell_{\mathsf{sk}}) = \omega(\log \lambda)$, there exists a weakly L-noisy-leakage-resilient PKE scheme with simple key generation whose secret key length is ℓ'_{sk}. Furthermore, the construction is fully black-box.*[7]

For example, from an IND-CPA secure PKE scheme with simple key generation with secret key length ℓ_{sk}, for any constant $\beta \in [0, 1)$, we can construct a scheme whose secret key length is ℓ'_{sk} and satisfies weak $(\beta\ell'_{\mathsf{sk}})$-noisy-leakage-resilience by setting the term $\omega(\log \lambda)$ simply as λ and setting $\ell'_{\mathsf{sk}} := \frac{\ell_{\mathsf{sk}}+\lambda}{1-\beta}$.

KDM-CCA/CPA Security. We recall KDM-CCA/CPA security for PKE.

Definition 2 (KDM-CCA/CPA Security) *Let* $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme whose secret key length and message length are ℓ_{sk} and μ, respectively. Let $n = n(\lambda)$ be a polynomial, and $\mathcal{F}$ be a family of functions with domain $(\{0,1\}^{\ell_{\mathsf{sk}}})^n$ and range $\{0,1\}^\mu$. We say that* PKE *is* KDM-CCA secure with respect to $\mathcal{F}$ in the n-key setting ($\mathcal{F}$-KDM$^{(n)}$-CCA secure) *if for all PPT adversaries $\mathcal{A}$, we have* $\mathsf{Adv}^{\mathsf{kdmcca}}_{\mathsf{PKE},\mathcal{F},\mathcal{A},n}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathsf{kdmcca}}_{\mathsf{PKE},\mathcal{F},\mathcal{A},n}(\lambda) = 1] - 1/2| = \mathsf{negl}(\lambda)$, *where the experiment* $\mathsf{Expt}^{\mathsf{kdmcca}}_{\mathsf{PKE},\mathcal{F},\mathcal{A},n}(\lambda)$ *is described in Figure 2.*

KDM-CPA security with respect to $\mathcal{F}$ in the n-key setting ($\mathcal{F}$-KDM$^{(n)}$-CPA security) is defined analogously, except that $\mathcal{A}$ is disallowed to use $\mathcal{O}_{\mathsf{dec}}$.

Function Families for KDM Security. In this paper, the function families for KDM security that we will specifically treat are as follows.

[6] In this paper, we only consider (public-key/secret-key) encryption schemes in which secret keys and messages are bit strings, whose lengths are determined by the security parameter λ.

[7] A fully black-box construction of a primitive Q from another primitive P means that (1) the construction of Q treats an instance of P as an oracle, and (2) the reduction algorithm (for proving the security of the construction of Q) treats the adversary attacking the construction of Q and the instance of P as oracles. (See [RTV04] for the formal treatment.)

```
$\mathsf{Expt}^{\mathsf{kdmcca}}_{\mathsf{PKE},\mathcal{F},\mathcal{A},n}(\lambda)$ :
    $L_{\mathsf{kdm}} \leftarrow \emptyset$
    $\forall s \in [n] : (\mathsf{pk}^s, \mathsf{sk}^s) \leftarrow \mathsf{KG}(1^\lambda)$
    $b \xleftarrow{\mathsf{r}} \{0,1\}$
    $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{kdm}}(\cdot,\cdot,\cdot),\mathcal{O}_{\mathsf{dec}}(\cdot,\cdot)}((\mathsf{pk}^s)_{s\in[n]})$
    Return $(b' \overset{?}{=} b)$.

$\mathcal{O}_{\mathsf{kdm}}(\alpha, f_0, f_1)$ :    // $\alpha \in [n]$, $f_0, f_1 \in \mathcal{F}$
    $\mathsf{m} \leftarrow f_b((\mathsf{sk}^s)_{s\in[n]})$
    $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}^\alpha, \mathsf{m})$
    $L_{\mathsf{kdm}} \leftarrow L_{\mathsf{kdm}} \cup \{(\alpha, \mathsf{ct})\}$
    Return $\mathsf{ct}$.

$\mathcal{O}_{\mathsf{dec}}(\alpha, \mathsf{ct})$ :              // $\alpha \in [n]$
    If $(\alpha, \mathsf{ct}) \in L_{\mathsf{kdm}}$ then return $\bot$.
    Return $\mathsf{Dec}(\mathsf{pk}^\alpha, \mathsf{sk}^\alpha, \mathsf{ct})$.
```

Figure 2: The KDM-CCA experiment for PKE.

```
$\mathsf{Expt}^{\mathsf{circ}}_{\mathsf{PKE},\mathcal{A},n}(\lambda)$ :
    $\forall s \in [n] : (\mathsf{pk}^s, \mathsf{sk}^s) \leftarrow \mathsf{KG}(1^\lambda)$
    $b \xleftarrow{\mathsf{r}} \{0,1\}$
    $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{circ}}(\cdot,\cdot)}((\mathsf{pk}^s)_{s\in[n]})$
    Return $(b' \overset{?}{=} b)$.

$\mathcal{O}_{\mathsf{circ}}(\alpha, \mathtt{cmd})$ :    // $\alpha \in [n]$, $\mathtt{cmd} \in ([n] \times [\ell_{\mathsf{sk}}]) \cup \{\mathtt{zero}, \mathtt{one}\}$
    $\mathsf{m}_1 \leftarrow \begin{cases} \mathsf{sk}^\beta[i] & \text{if } \mathtt{cmd} = (\beta, i) \in [n] \times [\ell_{\mathsf{sk}}] \\ 0 & \text{if } \mathtt{cmd} = \mathtt{zero} \\ 1 & \text{if } \mathtt{cmd} = \mathtt{one} \end{cases}$
    $\mathsf{m}_0 \leftarrow 0$
    Return $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}^\alpha, \mathsf{m}_b)$
```

Figure 3: The circular security experiment for bit-PKE.

- $\mathcal{P}$ *(Projection functions):* A function is said to be a projection function if each of its output bits depends on at most a single bit of its input. We denote by $\mathcal{P}$ the family of projection functions.

- $\mathcal{B}_{\mathsf{size}}$ *(Circuits of a-priori bounded size* size*):* We denote by $\mathcal{B}_{\mathsf{size}}$, where $\mathsf{size} = \mathsf{size}(\lambda)$ is a polynomial, the function family each of whose members can be described by a circuit of size size.

Circular Security. In this paper, we also treat circular security (against CPA), which we consider for bit-encryption schemes. Although it is a special case of KDM security, it is convenient for us to introduce a separate definition in the form we use in this paper.

Definition 3 (Circular Security for Bit-PKE) *Let $n = n(\lambda)$ be a polynomial. Let* $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a bit-PKE scheme with the secret key length ℓ_{sk}. We say that* PKE *is* circular secure in the n-key setting (CIRC$^{(n)}$ secure) *if for all PPT adversaries $\mathcal{A}$, we have* $\mathsf{Adv}^{\mathsf{circ}}_{\mathsf{PKE},\mathcal{A},n}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathsf{circ}}_{\mathsf{PKE},\mathcal{A},n}(\lambda) = 1] - 1/2| = \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Expt}^{\mathsf{circ}}_{\mathsf{PKE},\mathcal{A},n}(\lambda)$ *is described in Figure 3.*

Our definition here follows the definition called "circular security with respect to indistinguishability of oracles" formalized by Rothblum [Rot13], with a slight modification to the interface of the oracle: In addition to capturing the multi-key setting, the circular-encryption oracle $\mathcal{O}_{\mathsf{circ}}$ in our definition accepts the special commands "zero" and "one" (returning an encryption of 0 and that of 1, respectively, in the case $b = 1$) to explicitly capture ordinary IND-CPA security. This is for convenience and clarity: A bit-encryption scheme satisfies our definition if and only if it simultaneously satisfies the original definition in [Rot13] (without the augmentation of the oracle interface) and IND-CPA security.

Secret-Key Encryption. An SKE scheme SKE consists of the three PPT algorithms $(\mathsf{K}, \mathsf{E}, \mathsf{D})$:

- K is the key generation algorithm that takes 1^λ as input, and outputs a secret key sk.

- E is the encryption algorithm that takes a secret key sk and a message m as input, and outputs a ciphertext ct.

- D is the (deterministic) decryption algorithm that takes a secret key sk and a ciphertext ct as input, and outputs a message m or the invalid symbol $\perp$.

An SKE scheme $\mathsf{SKE} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$ is said to be *correct* if for all $\lambda \in \mathbb{N}$, $\mathsf{sk} \leftarrow \mathsf{K}(1^\lambda)$ and m, we have $\mathsf{D}(\mathsf{sk}, \mathsf{E}(\mathsf{sk}, \mathsf{m})) = \mathsf{m}$.

We refer to an SKE scheme whose message space is 1-bit as a *bit-SKE* scheme.

Weak noisy-leakage-resilience, KDM security, and circular security for (bit-)SKE are defined analogously to those defined for (bit-)PKE, with the following natural adaptions in the security experiments:

- All of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$, $\mathsf{Enc}(\mathsf{pk}, \cdot)$, and $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, \cdot)$ in the experiments for PKE are replaced with $\mathsf{sk} \leftarrow \mathsf{K}(1^\lambda)$, $\mathsf{E}(\mathsf{sk}, \cdot)$, and $\mathsf{D}(\mathsf{sk}, \cdot)$ in the experiments for SKE, respectively. We do the same treatment for those with the superscripts $s, \alpha \in [n]$.

- All the public keys pk and pk^s ($s \in [n]$) given as input to an adversary in the experiments for PKE are replaced with 1^λ in the experiments for SKE.

Results from [KMT19, KM19]. We recall the results on IND-CCA/KDM-CCA secure PKE from [KMT19, KM19], which we will use in Section 5.

Theorem 2 **([KMT19])** *If there exist an IND-CPA secure PKE scheme and a $\mathcal{P}$-KDM$^{(1)}$-CPA secure SKE scheme, then there exists an IND-CCA secure PKE scheme. Furthermore, the construction is fully black-box.*

Theorem 3 **([KM19])** *If there exist an IND-CPA secure PKE scheme and a $\mathcal{P}$-KDM$^{(1)}$-CPA secure SKE scheme, then for any polynomial $\mathsf{size} = \mathsf{size}(\lambda)$, there exists a $\mathcal{B}_{\mathsf{size}}$-KDM$^{(1)}$-CCA secure PKE scheme.*

We note that [KM19] also showed a construction of a multi-key-KDM-CCA secure PKE scheme by additionally assuming (passive) *RKA-KDM security* with respect to projection functions for the underlying SKE scheme. We do not formally recall it here since it is not known if it follows from the multi-key version of ordinary $\mathcal{P}$-KDM security and our result in Section 7 improves it in terms of the strength of assumptions.

3.3 Targeted Encryption

Here, we recall targeted encryption (TE) [BHHI10].

A TE scheme TE consists of the three PPT algorithms $(\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec})$:

- TKG is the key generation algorithm that takes 1^λ as input, and outputs a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$, where $|\mathsf{sk}| =: \ell_{\mathsf{sk}}$.

- TEnc is the encryption algorithm that takes a public key pk, an index $i \in [\ell_{\mathsf{sk}}]$, a bit $v \in \{0,1\}$, and a message m as input, and outputs a ciphertext ct.

- TDec is the (deterministic) decryption algorithm that takes a public key pk, a secret key $\mathsf{sk} \in \{0,1\}^{\ell_{\mathsf{sk}}}$, an index $i \in [\ell_{\mathsf{sk}}]$, and a ciphertext ct as input, and outputs a message m or the invalid symbol $\perp$.

As the correctness for a TE scheme, we require that for all $\lambda \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{TKG}(1^\lambda)$, $i \in [\ell_{\mathsf{sk}}]$, and m, we have $\mathsf{TDec}(\mathsf{pk}, \mathsf{sk}, i, \mathsf{TEnc}(\mathsf{pk}, i, \mathsf{sk}[i], \mathsf{m})) = \mathsf{m}$.

Barak et al. [BHHI10] defined two kinds of security notions for TE: *security against the receiver* and *security against outsiders*. We recall them here.

```
$\mathsf{Expt}^{\mathsf{receiver}}_{\mathsf{TE},\mathcal{A}}(\lambda)$ :
    $(i^* \in [\ell_{\mathsf{sk}}], \mathsf{st}) \leftarrow \mathcal{A}_0(1^\lambda)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{TKG}(1^\lambda)$
    $b \xleftarrow{\mathsf{r}} \{0,1\}$
    $b' \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{TEnc}}(\cdot,\cdot)}(\mathsf{pk}, \mathsf{sk}, \mathsf{st})$
    Return $(b' \overset{?}{=} b)$.
- - - - - - - - - - - - - - - - - - - - - - - - -
$\mathcal{O}_{\mathsf{TEnc}}(\mathsf{m}_0, \mathsf{m}_1)$ :    // $|\mathsf{m}_0| = |\mathsf{m}_1|$
    $\mathsf{ct} \leftarrow \mathsf{TEnc}(\mathsf{pk}, i^*, 1 \oplus \mathsf{sk}[i^*], \mathsf{m}_b)$
    Return $\mathsf{ct}$.
```

```
$\mathsf{Expt}^{\mathsf{outsider}}_{\mathsf{TE},\mathcal{A}}(\lambda)$ :
    $(i^* \in [\ell_{\mathsf{sk}}], v^* \in \{0,1\}, \mathsf{st}) \leftarrow \mathcal{A}_0(1^\lambda)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{TKG}(1^\lambda)$
    $b \xleftarrow{\mathsf{r}} \{0,1\}$
    $b' \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{TEnc}}(\cdot,\cdot)}(\mathsf{pk}, \mathsf{st})$
    Return $(b' \overset{?}{=} b)$.
- - - - - - - - - - - - - - - - - - - - - - - - -
$\mathcal{O}_{\mathsf{TEnc}}(\mathsf{m}_0, \mathsf{m}_1)$ :    // $|\mathsf{m}_0| = |\mathsf{m}_1|$
    $\mathsf{ct} \leftarrow \mathsf{TEnc}(\mathsf{pk}, i^*, v^*, \mathsf{m}_b)$
    Return $\mathsf{ct}$.
```

Figure 4: The experiments for TE: Security against the receiver (left) and security against outsiders (right).

Security against the Receiver. As the name suggests, this is a security notion against a receiver who holds a secret key. More specifically, this security notion ensures that for every $i \in [\ell_{\mathsf{sk}}]$, if a message is encrypted under the position $(i, 1 \oplus \mathsf{sk}[i])$, its information does not leak to the receiver of the ciphertext who holds a secret key sk. For convenience, we introduce the multi-challenge version of this security notion, which can be shown to be equivalent to the single-challenge version defined in [BHHI10] via a query-wise hybrid argument.

Formally, for a TE scheme $\mathsf{TE} = (\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec})$ and an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, consider the experiment $\mathsf{Expt}^{\mathsf{receiver}}_{\mathsf{TE},\mathcal{A}}(\lambda)$ described in Figure 4 (left). We emphasize again that since this security is considered against a receiver, an adversary is given a secret key sk as input.[8]

Definition 4 (Security against the Receiver) *We say that a TE scheme* TE *satisfies* security against the receiver *if for all PPT adversaries* $\mathcal{A}$*, we have* $\mathsf{Adv}^{\mathsf{receiver}}_{\mathsf{TE},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathsf{receiver}}_{\mathsf{TE},\mathcal{A}}(\lambda) = 1] - 1/2| = \mathsf{negl}(\lambda)$.

Security against Outsiders. This security notion simply ensures that ciphertexts generated under any pair $(i, v) \in [\ell_{\mathsf{sk}}] \times \{0,1\}$ do not leak the information of encrypted messages. Again, we introduce the multi-challenge version for this security notion, which is equivalent to the single-challenge version formalized in [BHHI10].

Formally, for a TE scheme $\mathsf{TE} = (\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec})$ and an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, consider the experiment $\mathsf{Expt}^{\mathsf{outsider}}_{\mathsf{TE},\mathcal{A}}(\lambda)$ described in Figure 4 (right).

Definition 5 (Security against Outsiders) *We say that a TE scheme* TE *satisfies* security against outsiders *if for all PPT adversaries* $\mathcal{A}$*, we have* $\mathsf{Adv}^{\mathsf{outsider}}_{\mathsf{TE},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathsf{outsider}}_{\mathsf{TE},\mathcal{A}}(\lambda) = 1] - 1/2| = \mathsf{negl}(\lambda)$.

Result from [BHHI10]. Barak et al. [BHHI10] showed the following result, which we will use in Section 5.

Theorem 4 ([BHHI10]) *If there exists a TE scheme satisfying security against the receiver and security against outsiders, then for any polynomial* $\mathsf{size} = \mathsf{size}(\lambda)$*, there exists a* $\mathcal{B}_{\mathsf{size}}$*-KDM*$^{(1)}$*-CPA secure PKE scheme. Furthermore, there is a fully black-box construction of a* $\mathcal{P}$*-KDM*$^{(1)}$*-CPA secure PKE scheme from a TE scheme satisfying the two security notions.*

[8] The original definition by Barak et al. [BHHI10] considered statistical security (i.e. security against computationally unbounded adversaries), but it was remarked there that computational security suffices for their construction of KDM-CPA secure PKE.

We remark that the result on the fully black-box construction can be extended to any function family such that a canonical description of a circuit computing any function in the family can be learned and reconstructed (with overwhelming probabiltiy) by just making polynomially many oracle queries to the function. (This is because in the security proof in [BHHI10], what is garbled is a function queried as a KDM-encryption query.) We only state it for $\mathcal{P}$-KDM security since it is sufficient for our purpose.

We also remark that [BHHI10] also showed that their construction achieves KDM-CPA security in the multi-key setting by additionally assuming that the underlying TE scheme is an *augmented TE* scheme satisfying circular security in the multi-key setting. We do not recall this result and the formal definition of augmented TE since we do not use them directly. In Section 6, we introduce conformed TE, which is also an extension of TE in a similar manner to augmented TE but has several differences. For the details, see the explanation there.

4 Targeted Encryption from Circular Security and Leakage-Resilience

In this section, as our main technical result, we show how to construct a TE scheme from the combination of a circular secure bit-SKE scheme (in the single-key setting) and a weakly noisy-leakage-resilient PKE scheme.

Construction. Our construction uses the following building blocks:

- Let $\mathsf{SKE} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$ be a CIRC$^{(1)}$ secure bit-SKE scheme with the secret-key length ℓ_{k} for some polynomial $\ell_{\mathsf{k}} = \ell_{\mathsf{k}}(\lambda)$. We assume that there exists a special symbol $\mathtt{flip}$ that is perfectly distinguishable from possible outputs of E.

- Let $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a weakly L-noisy-leakage-resilient PKE scheme with simple key generation whose secret-key length is ℓ_{sk} for some polynomial $\ell_{\mathsf{sk}} = \ell_{\mathsf{sk}}(\lambda)$. We assume $L = 0.6\ell_{\mathsf{sk}}$.

Using these buliding blocks, we construct a TE scheme $\mathsf{TE} = (\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec})$, whose secret key length is ℓ_{k}, as described in Figure 5.

Correctness. The correctness of TE follows from that of the building blocks SKE and PKE. Specifically, since $\mathsf{TEnc}(\mathsf{PK}, i, \mathsf{SK}[i] = \mathsf{k}[i], \mathsf{m})$ just computes $\mathsf{Enc}(\mathsf{pk}_{i,\mathsf{k}[i]}, \mathsf{m})$ and $\mathsf{TDec}(\mathsf{PK}, \mathsf{SK}, i, \mathsf{ct})$ computes $\mathsf{Dec}(\mathsf{pk}_{i,\mathsf{k}[i]}, \mathsf{sk}', \mathsf{ct})$ in its last step, it suffices to see that sk' computed in TDec always equals to $\mathsf{sk}_{i,\mathsf{k}[i]}$ for any $i \in [\ell_{\mathsf{k}}]$. Indeed, for every $j \in [\ell_{\mathsf{sk}}]$, we have

- If $(\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) = (1, 0)$, then note that this case implies $\mathsf{sk}_{i,\mathsf{k}[i]}[j] = 1 \oplus \mathsf{k}[i]$. On the other hand, $\mathsf{e}_{i,j} = \mathtt{flip}$ holds by the design of TKG. Hence, TDec sets $\mathsf{sk}'[j] \leftarrow 1 \oplus \mathsf{k}[i] = \mathsf{sk}_{i,\mathsf{k}[i]}[j]$.

- Otherwise (i.e. $(\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) \neq (1, 0)$), $\mathsf{e}_{i,j}$ is just an encryption of $\mathsf{sk}_{i,\mathsf{k}[i]}[j]$. Thus, TDec decrypts it as $\mathsf{sk}'[j] = \mathsf{D}(\mathsf{k}, \mathsf{e}_{i,j}) = \mathsf{sk}_{i,\mathsf{k}[i]}[j]$.

Hence, we have $\mathsf{sk}'[j] = \mathsf{sk}_{i,\mathsf{k}[i]}[j]$ for every $j \in [\ell_{\mathsf{sk}}]$, namely, $\mathsf{sk}' = \mathsf{sk}_{i,\mathsf{k}[i]}$ holds. Thus, TE satisfies correctness.

```
$\mathsf{TKG}(1^\lambda)$ :
  $\mathsf{k} \leftarrow \mathsf{K}(1^\lambda)$
  $\forall i \in [\ell_{\mathsf{k}}]$:
    $\forall v \in \{0,1\}$:  $\mathsf{sk}_{i,v} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{sk}}}$;  $\mathsf{pk}_{i,v} \leftarrow \mathsf{KG}(\mathsf{sk}_{i,v})$
    $\forall j \in [\ell_{\mathsf{sk}}]$:
      $\mathsf{e}_{i,j} \leftarrow \begin{cases} \mathtt{flip} & \text{if } (\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) = (1,0) \\ \mathsf{E}(\mathsf{k}, \mathsf{sk}_{i,\mathsf{k}[i]}[j]) & \text{otherwise} \end{cases}$
  $\mathsf{PK} \leftarrow (\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}, \mathsf{e}_{i,1}, \dots, \mathsf{e}_{i,\ell_{\mathsf{sk}}})_{i \in [\ell_{\mathsf{k}}]}$;  $\mathsf{SK} \leftarrow \mathsf{k}$
  Return $(\mathsf{PK}, \mathsf{SK})$.

$\mathsf{TEnc}(\mathsf{PK}, i, v, \mathsf{m})$ :
  $(\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}, \mathsf{e}_{i,1}, \dots, \mathsf{e}_{i,\ell_{\mathsf{sk}}})_{i \in [\ell_{\mathsf{k}}]} \leftarrow \mathsf{PK}$
  Return $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}_{i,v}, \mathsf{m})$.

$\mathsf{TDec}(\mathsf{PK}, \mathsf{SK} = \mathsf{k}, i, \mathsf{ct})$ :
  $(\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}, \mathsf{e}_{i,1}, \dots, \mathsf{e}_{i,\ell_{\mathsf{sk}}})_{i \in [\ell_{\mathsf{k}}]} \leftarrow \mathsf{PK}$
  $\forall j \in [\ell_{\mathsf{sk}}]$:
    $\mathsf{sk}'[j] \leftarrow \begin{cases} 1 \oplus \mathsf{k}[i] & \text{if } \mathsf{e}_{i,j} = \mathtt{flip} \\ \mathsf{D}(\mathsf{k}, \mathsf{e}_{i,j}) & \text{otherwise} \end{cases}$
  Return $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{pk}_{i,\mathsf{k}[i]}, \mathsf{sk}', \mathsf{ct})$.
```

Figure 5: The construction of a TE scheme TE from a circular secure bit-SKE scheme SKE and a weakly noisy-leakage-resilient PKE scheme PKE.

Security. We now show that TE satisfies the two security notions for TE.

Theorem 5 *If* PKE *is weakly* $(0.6\ell_{\mathsf{sk}})$*-noisy-leakage-resilient, then* TE *satisfies security against the receiver.*

Proof of Theorem 5. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be any PPT adversary that attacks the security against the receiver of TE. We show that for $\mathcal{A}$, there exists a PPT $(0.6\ell_{\mathsf{sk}})$-noisy-leakage-respecting adversary $\mathcal{B}$ such that $\mathsf{Adv}^{\mathsf{receiver}}_{\mathsf{TE},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{wlr}}_{\mathsf{PKE},\mathcal{B},0.6\ell_{\mathsf{sk}}}(\lambda)$, which implies the theorem. The description of $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$ is as follows.

$\mathcal{B}_0(1^\lambda)$**:** $\mathcal{B}_0$ first runs $(i^*, \mathsf{st}) \leftarrow \mathcal{A}_0(1^\lambda)$. Next, $\mathcal{B}_0$ computes $\mathsf{k} \leftarrow \mathsf{K}(1^\lambda)$, and picks $\mathsf{sk}_{i^*,\mathsf{k}[i^*]} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{sk}}}$. Let $P := \{j \in [\ell_{\mathsf{sk}}] \mid \mathsf{sk}_{i^*,\mathsf{k}[i^*]}[j] = 1 \oplus \mathsf{k}[i^*]\}$ and $\ell := |P|$, and suppose P is $\{p_1, \dots, p_\ell\}$ such that $1 \leq p_1 < \cdots < p_\ell \leq \ell_{\mathsf{sk}}$. $\mathcal{B}_0$ defines the leakage function $f_P : \{0,1\}^{\ell_{\mathsf{sk}}} \to \{0,1\}^\ell$ by

$$f_P(\mathsf{z}) := (\mathsf{z}[p_1], \dots, \mathsf{z}[p_\ell]) \in \{0,1\}^\ell.$$

Then, $\mathcal{B}_0$ sets $\mathsf{st}_{\mathcal{B}}$ as all the information known to $\mathcal{B}_0$, and terminates with output $(f_P, \mathsf{st}_{\mathcal{B}})$.

$\mathcal{B}_1^{\mathcal{O}_{\mathsf{Enc}}(\cdot,\cdot)}(\mathsf{pk}', f_P(\mathsf{sk}') = (\mathsf{sk}'[p_1], \dots, \mathsf{sk}'[p_\ell]) \in \{0,1\}^\ell, \mathsf{st}_{\mathcal{B}})$**:** (where $(\mathsf{pk}', \mathsf{sk}')$ denotes the key pair generated in $\mathcal{B}$'s experiment) $\mathcal{B}_1$ first computes $\mathsf{pk}_{i^*,\mathsf{k}[i^*]} \leftarrow \mathsf{KG}(\mathsf{sk}_{i^*,\mathsf{k}[i^*]})$, and regards pk' as $\mathsf{pk}_{i^*,1\oplus\mathsf{k}[i^*]}$ (correspondingly, implicitly regards sk' as $\mathsf{sk}_{i^*,1\oplus\mathsf{k}[i^*]} \in \{0,1\}^{\ell_{\mathsf{sk}}}$). Then, for every $j \in [\ell_{\mathsf{sk}}]$, $\mathcal{B}_1$ generates $\mathsf{e}_{i^*,j}$ by

$$\mathsf{e}_{i^*,j} \leftarrow \begin{cases} \mathtt{flip} & \text{if } j \in P \wedge \mathsf{sk}'[j] = \mathsf{k}[i^*] \\ \mathsf{E}(\mathsf{k}, \mathsf{sk}_{i^*,\mathsf{k}[i^*]}[j]) & \text{otherwise} \end{cases}.$$

Note that by the definition of P, we have $\mathsf{sk}_{i^*,\mathsf{k}[i^*]}[j] = 1 \oplus \mathsf{k}[i^*]$ if and only if $j \in P$. Furthermore, by the definition of the leakage function $f_P(\cdot)$, we have $\mathsf{sk}'[j] = \mathsf{sk}_{i^*,1\oplus\mathsf{k}[i^*]}[j]$ for all $j \in P$. Hence, we have

$$\begin{aligned} j \in P \wedge \mathsf{sk}'[j] = \mathsf{k}[i^*] \quad &\Longleftrightarrow \quad (\mathsf{sk}_{i^*,\mathsf{k}[i^*]}[j], \mathsf{sk}_{i^*,1\oplus\mathsf{k}[i^*]}[j]) = (1 \oplus \mathsf{k}[i^*], \mathsf{k}[i^*]) \\ &\Longleftrightarrow \quad (\mathsf{sk}_{i^*,0}[j], \mathsf{sk}_{i^*,1}[j]) = (1,0). \end{aligned}$$

Hence, the generation of $\mathsf{e}_{i^*,j}$ is in fact exactly the same as in $\mathsf{Expt}^{\mathsf{receiver}}_{\mathsf{TE},\mathcal{A}}(\lambda)$.

Then, $\mathcal{B}_1$ generates the remaining components in $\mathsf{PK} = (\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}, \mathsf{e}_{i,1}, \dots, \mathsf{e}_{i,\ell_{\mathsf{sk}}})_{i \in [\ell_{\mathsf{k}}]}$ (i.e. the components for the positions $i \in [\ell_{\mathsf{k}}] \setminus \{i^*\}$) by itself exactly as $\mathsf{TKG}(1^\lambda)$ does.

Now, $\mathcal{B}_1$ runs $\mathcal{A}_1(\mathsf{PK}, \mathsf{SK} = \mathsf{k}, \mathsf{st})$. When $\mathcal{A}_1$ submits an encryption query $(\mathsf{m}_0, \mathsf{m}_1)$, $\mathcal{B}_1$ just forwards it to its own encryption oracle $\mathcal{O}_{\mathsf{Enc}}(\cdot,\cdot)$, and returns whatever returned from the oracle to $\mathcal{A}_1$.

When $\mathcal{A}_1$ terminates with output b', $\mathcal{B}_1$ terminates with output b'.

The above completes the description of $\mathcal{B}$. As mentioned above, $\mathcal{B}$ generates the key pair $(\mathsf{PK}, \mathsf{SK})$ with exactly the same distribution as that in the actual experiment for security against the receiver. Since $\mathcal{B}$ embeds its instance pk' to the position $(i^*, 1 \oplus \mathsf{k}[i^*])$, it is straightforward to see that $\mathcal{B}$ perfectly simulates the security experiment for $\mathcal{A}$ so that $\mathcal{A}$'s the challenge bit is that of $\mathcal{B}$'s, and thus $\mathcal{B}$'s advantage is exactly the same as that of $\mathcal{A}$'s.

It remains to confirm that $\mathcal{B}$ is a $(0.6\ell_{\mathsf{sk}})$-noisy-leakage-respecting adversary, namely, $0.6\ell_{\mathsf{sk}} \geq \mathbf{H}_\infty(\mathsf{sk}') - \widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f_P(\mathsf{sk}'), \mathsf{st}_{\mathcal{B}}) = \ell_{\mathsf{sk}} - \widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f_P(\mathsf{sk}'), \mathsf{st}_{\mathcal{B}})$ or equivalently $2^{-\widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f_P(\mathsf{sk}'),\mathsf{st}_{\mathcal{B}})} \leq 2^{-0.4\ell_{\mathsf{sk}}}$ holds. To see this, firstly note that $\mathsf{st}_{\mathcal{B}}$ output by $\mathcal{B}_0$ is independent of the choise of $\mathsf{sk}' \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{sk}}}$, and thus we have $\widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f_P(\mathsf{sk}'), \mathsf{st}_{\mathcal{B}}) = \widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f_P(\mathsf{sk}'))$. Thus, it is sufficient to show $2^{-\widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f_P(\mathsf{sk}'))} \leq 2^{-0.4\ell_{\mathsf{sk}}}$. Next, notice that P is distributed uniformly over $2^{[\ell_{\mathsf{sk}}]}$ (i.e. all the subsets of $[\ell_{\mathsf{sk}}]$), since P is determined by the random choice of $\mathsf{sk}_{i^*,\mathsf{k}[i^*]} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{sk}}}$. Thus, we have

$$2^{-\widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f_P(\mathsf{sk}'))} = \mathop{\mathbf{E}}_{P \xleftarrow{\mathsf{r}} 2^{[\ell_{\mathsf{sk}}]},\ y \xleftarrow{\mathsf{r}} \{0,1\}^{|P|}} \left[\max_{x^*} \Pr_{\mathsf{sk}' \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{sk}}}} [\mathsf{sk}' = x^* | f_P(\mathsf{sk}') = y] \right]$$

$$= \mathop{\mathbf{E}}_{P \xleftarrow{\mathsf{r}} 2^{[\ell_{\mathsf{sk}}]}} \left[2^{-\ell_{\mathsf{sk}} + |P|} \right] = 2^{-2\ell_{\mathsf{sk}}} \cdot \sum_{P' \subseteq [\ell_{\mathsf{sk}}]} 2^{|P'|} = 2^{-2\ell_{\mathsf{sk}}} \cdot \sum_{k=0}^{\ell_{\mathsf{sk}}} \binom{\ell_{\mathsf{sk}}}{k} \cdot 2^k$$

$$\overset{(*)}{=} 2^{-2\ell_{\mathsf{sk}}} \cdot 3^{\ell_{\mathsf{sk}}} = 2^{-(2 - \log_2 3)\ell_{\mathsf{sk}}} \overset{(\dagger)}{<} 2^{-0.4\ell_{\mathsf{sk}}},$$

where the equality $(*)$ uses $\sum_{k=0}^{n} \binom{n}{k} x^k = (1+x)^n$, and the inequality $(\dagger)$ uses $\log_2 3 < 1.6$. Hence, $\mathcal{B}$ is $(0.6\ell_{\mathsf{sk}})$-noisy-leakage-respecting. □ (**Theorem 5**)

Theorem 6 *If* SKE *is CIRC*$^{(1)}$ *secure and* PKE *is* $(0.6\ell_{\mathsf{sk}})$*-noisy-leakage-resilient, then* TE *satisfies security against outsiders.*

Proof of Theorem 6. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be any PPT adversary that attacks the security against outsiders of TE. We show that there exist PPT adversaries $\mathcal{B}_{\mathsf{c}}$ and $\mathcal{B}_{\mathsf{w}}$ (where the latter is $(0.6\ell_{\mathsf{sk}})$-noisy-leakage-respecting) satisfying

$$\mathsf{Adv}^{\mathsf{outsider}}_{\mathsf{TE},\mathcal{A}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathsf{circ}}_{\mathsf{SKE},\mathcal{B}_{\mathsf{c}},1}(\lambda) + \mathsf{Adv}^{\mathsf{wlr}}_{\mathsf{PKE},\mathcal{B}_{\mathsf{w}},0.6\ell_{\mathsf{sk}}}(\lambda), \qquad (1)$$

which implies the theorem.

To this end, we consider the following two games Game 1 and Game 2.

Game 1: This is the experiment for security against outsiders $\mathsf{Expt}^{\mathsf{outsider}}_{\mathsf{TE},\mathcal{A}}(\lambda)$.

Game 2: Same as Game 1, except that every invocation of $\mathsf{E}(\mathsf{k},\cdot)$ during the generation of PK is replaced with $\mathsf{E}(\mathsf{k}, 0)$.

For $t \in \{1,2\}$, let $\mathtt{SUC}_t$ be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game t. By the definitions of the games and events and the triangle inequality, we have

$$\mathsf{Adv}^{\mathsf{outsider}}_{\mathsf{TE},\mathcal{A}}(\lambda) = 2 \cdot \left|\Pr[\mathtt{SUC}_1] - \frac{1}{2}\right| \le 2 \cdot \left|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]\right| + 2 \cdot \left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right|. \qquad (2)$$

In the following, we show how the terms appearing in Equation 2 are bounded.

Lemma 2 *There exists a PPT adversary* $\mathcal{B}_{\mathsf{c}}$ *such that* $\mathsf{Adv}^{\mathsf{circ}}_{\mathsf{SKE},\mathcal{B}_{\mathsf{c}},1}(\lambda) = |\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]|$.

Proof of Lemma 2. The description of $\mathcal{B}_{\mathsf{c}}$ is as follows. Below, k and β denote the secret key and the challenge bit, respectively, chosen in $\mathcal{B}_{\mathsf{c}}$'s experiment. Furthermore, since there is only a single key in the experiment of $\mathcal{B}_{\mathsf{c}}$, we simplify the interface of the circular-encryption oracle $\mathcal{O}_{\mathsf{circ}}$ to take just $\mathtt{cmd} \in [\ell_{\mathsf{k}}] \cup \{\mathtt{zero}, \mathtt{one}\}$ as input.

$\mathcal{B}_{\mathsf{c}}^{\mathcal{O}_{\mathsf{circ}}(\cdot)}(1^\lambda)$**:** $\mathcal{B}_{\mathsf{c}}$ first runs $(i^*, v^*, \mathsf{st}) \leftarrow \mathcal{A}_0(1^\lambda)$. Next, for every $i \in [\ell_{\mathsf{k}}]$, $\mathcal{B}_{\mathsf{c}}$ does the following:

1. For both $v \in \{0,1\}$, pick $\mathsf{sk}_{i,v} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{sk}}}$ and compute $\mathsf{pk}_{i,v} \leftarrow \mathsf{KG}(\mathsf{sk}_{i,v})$.
2. For the positions $j \in [\ell_{\mathsf{sk}}]$ for which $(\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) = (1,0)$ holds, set $\mathsf{e}_{i,j} \leftarrow \mathtt{flip}$.
3. For the remaining positions $j \in [\ell_{\mathsf{sk}}]$ with $(\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) \ne (1,0)$, set

$$\mathtt{cmd}_j \leftarrow \begin{cases} \mathtt{zero} & \text{if } (\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) = (0,0) \\ \mathtt{one} & \text{if } (\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) = (1,1) \\ i & \text{if } (\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) = (0,1) \end{cases},$$

submit $\mathtt{cmd}_j$ to $\mathcal{B}_{\mathsf{c}}$'s oracle $\mathcal{O}_{\mathsf{circ}}(\cdot)$, and receive $\mathsf{e}_{i,j}$ as the answer from $\mathcal{O}_{\mathsf{circ}}$.
Note that if $(\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) = (0,1)$ then $\mathsf{sk}_{i,\mathsf{k}[i]}[j] = \mathsf{k}[i]$ holds, and the latter is trivially true for the cases $(\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) \in \{(0,0),(1,1)\}$. Thus, $\mathcal{O}_{\mathsf{circ}}$ computes $\mathsf{e}_{i,j}$ as follows:

$$\mathsf{e}_{i,j} \leftarrow \begin{cases} \mathsf{E}(\mathsf{k}, \mathsf{sk}_{i,\mathsf{k}[i]}[j]) & \text{if } \beta = 1 \\ \mathsf{E}(\mathsf{k}, 0) & \text{if } \beta = 0 \end{cases}.$$

Therefore, if $\beta = 1$ (resp. $\beta = 0$), then $\mathsf{e}_{i,j}$ for every $j \in [\ell_{\mathsf{sk}}]$ is computed exactly as in Game 1 (resp. Game 2).

Then, $\mathcal{B}_{\mathsf{c}}$ sets $\mathsf{PK} \leftarrow (\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}, \mathsf{e}_{i,1}, \dots, \mathsf{e}_{i,\ell_{\mathsf{sk}}})_{i \in [\ell_{\mathsf{k}}]}$, picks $b \xleftarrow{\mathsf{r}} \{0,1\}$, and runs $\mathcal{A}_1(\mathsf{PK}, \mathsf{st})$.
$\mathcal{B}_{\mathsf{c}}$ answers $\mathcal{A}_1$'s encryption queries $(\mathsf{m}_0, \mathsf{m}_1)$ by returning $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}_{i^*,v^*}, \mathsf{m}_b)$.

When $\mathcal{A}_1$ terminates with output b', $\mathcal{B}_{\mathsf{c}}$ terminates with output $\beta' \leftarrow (b' \overset{?}{=} b)$.

The above completes the description of $\mathcal{B}_{\mathsf{c}}$. It is straightforward to see that if $\beta = 1$ (resp. $\beta = 0$), then $\mathcal{B}_{\mathsf{c}}$ simulates Game 1 (resp. Game 2) perfectly for $\mathcal{A}$. Since $\mathcal{B}_{\mathsf{c}}$ outputs $\beta' = 1$ if and only if $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs), we have

$$\mathsf{Adv}^{\mathsf{circ}}_{\mathsf{SKE},\mathcal{B}_{\mathsf{c}},1}(\lambda) = \left|\Pr[\beta' = 1 | \beta = 1] - \Pr[\beta' = 1 | \beta = 0]\right| = \left|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]\right|.$$

□ (**Lemma 2**)

Lemma 3 *There exists a PPT* $(0.6\ell_{\mathsf{sk}})$*-noisy-leakage-respecting adversary* $\mathcal{B}_{\mathsf{w}}$ *such that* $\mathsf{Adv}^{\mathsf{wlr}}_{\mathsf{PKE},\mathcal{B}_{\mathsf{w}},0.6\ell_{\mathsf{sk}}}(\lambda) = 2 \cdot |\Pr[\mathtt{SUC}_2] - 1/2]$.

Proof Sketch of Lemma 3. The reduction algorithm $\mathcal{B}_{\mathsf{w}}$ for the proof of this lemma proceeds very similarly to $\mathcal{B}$ used in the proof of Theorem 5, with the following differences:

- $\mathcal{B}_{\mathsf{w}}$ embeds its instance pk' into the position (i^*, v^*) output by $\mathcal{A}_0$ (rather than $(i^*, 1 \oplus \mathsf{k}[i^*])$), which means that $(\mathsf{pk}', \mathsf{sk}')$ now corresponds to $(\mathsf{pk}_{i^*,v^*}, \mathsf{sk}_{i^*,v^*})$; $\mathcal{B}_{\mathsf{w}}$ generates the key pair of the opposite position, namely $(\mathsf{pk}_{i^*,1\oplus v^*}, \mathsf{sk}_{i^*,1\oplus v^*})$ by itself.
- $\mathcal{B}_{\mathsf{w}}$ defines the set P by $P := \{j \in [\ell_{\mathsf{sk}}] | \mathsf{sk}_{i^*,1\oplus v^*}[j] = v^*\}$, and uses it to define the leakage function $f_P(\cdot)$ exactly $\mathcal{B}$ in the proof of Theorem 5 does. Note that since we have the correspondence $\mathsf{sk}' = \mathsf{sk}_{i^*,v^*}$, the leakage $f_P(\mathsf{sk}')$ is $(\mathsf{sk}_{i^*,v^*}[j])_{j\in P}$.
- For every $j \in [\ell_{\mathsf{sk}}]$, $\mathcal{B}_{\mathsf{w}}$ generates $\mathsf{e}_{i^*,j}$ by
$$\mathsf{e}_{i^*,j} \leftarrow \begin{cases} \mathtt{flip} & \text{if } j \in P \wedge \mathsf{sk}'[j] = 1 \oplus v^* \\ \mathsf{E}(\mathsf{k}, 0) & \text{otherwise} \end{cases}.$$
Then, by the definition of P and the correspondence $\mathsf{sk}' = \mathsf{sk}_{i^*,v^*}$, we have
$$\begin{aligned} j \in P \wedge \mathsf{sk}'[j] = 1 \oplus v^* &\iff (\mathsf{sk}_{i^*,1\oplus v^*}[j], \mathsf{sk}_{i^*,v^*}[j]) = (v^*, 1 \oplus v^*) \\ &\iff (\mathsf{sk}_{i^*,0}[j], \mathsf{sk}_{i^*,1}[j]) = (1, 0). \end{aligned}$$
Thus, $\mathsf{e}_{i^*,j}$ is generated exactly as in Game 2.

Then, it is straightforward to see that $\mathcal{B}_{\mathsf{w}}$ is $(0.6\ell_{\mathsf{sk}})$-noisy-leakage-respecting and simulates Game 2 perfectly for $\mathcal{A}$, and its advantage in attacking the weak noisy-leakage-resilience of PKE is exactly $2 \cdot |\Pr[\mathtt{SUC}_2] - 1/2|$. □ (**Lemma 3**)

Combining Lemmas 2 and 3 with Equation 2, we can conclude that there exist PPT adversaries $\mathcal{B}_{\mathsf{c}}$ and $\mathcal{B}_{\mathsf{w}}$ satisfying Equation 1. □ (**Theorem 6**)

5 Implications of Our TE Scheme

In this section, we explain the implications of our TE scheme in Section 4.

Completeness of Circular Security for KDM Security in the Single-Key Setting. Note that our construction of TE is a fully black-box construction from the building blocks. Moreover, by appropriately setting parameters, we can construct a PKE scheme with simple key generation whose secret key length is ℓ_{sk} and that satisfies weak $(0.6\ell_{\mathsf{sk}})$-noisy-leakage-resilience, based on any IND-CPA secure PKE scheme via Lemma 1. Hence, the following theorem follows from the combination of Theorems 4, 5, and 6, and Lemma 1.

Theorem 7 *If there exist an IND-CPA secure PKE scheme and a CIRC$^{(1)}$ secure bit-SKE scheme, then for any polynomial* $\mathsf{size} = \mathsf{size}(\lambda)$*, there exists a $\mathcal{B}_{\mathsf{size}}$-KDM$^{(1)}$-CPA secure PKE scheme. Furthermore, there exists a fully black-box construction of a $\mathcal{P}$-KDM$^{(1)}$-CPA secure PKE scheme from an IND-CPA secure PKE scheme and a CIRC$^{(1)}$ secure bit-SKE scheme.*

Combining Theorem 7 with Theorem 3, we obtain the following completeness theorem for KDM security in the single-key setting. This improves the results of [App11] and [KM19] in terms of assumptions.

Theorem 8 *If there exists an IND-CPA secure PKE scheme and a CIRC$^{(1)}$ secure bit-SKE scheme, then for any polynomial* $\mathsf{size} = \mathsf{size}(\lambda)$*, there exists a $\mathcal{B}_{\mathsf{size}}$-KDM$^{(1)}$-CCA secure PKE scheme.*

In Section 7, we will show that a similar completeness theorem for KDM security in the multi-key setting can be established. For the result, we will rely on the results on IND-CCA secure PKE and a reusable DV-NIZK argument system[9] for NP languages stated below.

Additional Results on IND-CCA PKE and DV-NIZK. As stated in Theorem 7, a $\mathcal{P}$-KDM$^{(1)}$-CPA secure PKE scheme can be constructed from an IND-CPA secure PKE and a CIRC$^{(1)}$ secure bit-SKE scheme in a fully black-box manner. Hence, combined with Theorem 2, we obtain the following result on IND-CCA secure PKE, which improves the results of [KMT19] and [HK15] in terms of assumptions.

Theorem 9 *There exists a fully black-box construction of an IND-CCA secure PKE scheme from an IND-CPA secure PKE scheme and a CIRC$^{(1)}$ secure bit-SKE scheme.*

Finally, combining Theorem 7 with Theorem 15, we also obtain the following result on a reusable DV-NIZK argument system, which improves the results of [KM19] and [LQR$^+$19] in terms of assumptions.

Theorem 10 *If there exists an IND-CPA secure PKE scheme and a CIRC$^{(1)}$ secure bit-SKE scheme, then there exists a reusable DV-NIZK argument system for all NP languages.*

6 Conformed Targeted Encryption

In this section, we introduce an encryption primitive that we call *conformed targeted encryption (CTE)*. This is an extension of an ordinary TE, and has some similar flavor to *augmented TE* formalized by Barak et al. [BHHI10]. Our definitional choice of CTE is made so that (1) it can be achieved from the combination of an IND-CPA secure PKE scheme and a circular secure bit-SKE scheme, and (2) it is sufficient as a building block for constructing a KDM-CCA secure PKE scheme in the multi-key setting.

In Section 6.1, we give the definitions for CTE and explain its difference with augmented TE formalized by Barak et al.. In Section 6.2, we show how our TE scheme presented in Section 4 can be extended to be a CTE scheme satisfying all the requirements.

6.1 Definitions

Syntax and Correctness. A *conformed targeted encryption* (CTE) scheme TE consists of the six algorithms $(\mathsf{CKG}, \mathsf{CEnc}, \mathsf{CDec}, \widehat{\mathsf{CDec}}, \mathsf{CSEnc}, \mathsf{CSDec})$:

- CKG, CEnc, and CDec are defined similarly to the key generation, encryption, and decryption algorithms of a TE scheme, respectively, except that in addition to a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$, CKG also outputs a trapdoor td. This process is written as $(\mathsf{pk}, \mathsf{sk}, \mathsf{td}) \leftarrow \mathsf{CKG}(1^\lambda)$.

- $\widehat{\mathsf{CDec}}$ is the trapdoor-decryption algorithm that takes td, an index $i \in [\ell_{\mathsf{sk}}]$, a bit $v \in \{0,1\}$, and a ciphertext ct (supposedly generated by CEnc) as input, and outputs a message m.

- CSEnc and CSDec are the additional *secret-key* encryption and decryption algorithms, respectively, where they use a secret key sk generated by CKG. We denote $\widetilde{\mathsf{ct}}$ to indicate that it is a ciphertext generated by CSEnc.

[9] The formal definitions for IND-CCA security, and those for a reusable DV-NIZK argument system as well as Theorem 15 (which recalls the result from [KM19, LQR$^+$19]) are given in Sections A.1 and A.2, respectively.

As the correctness for a CTE scheme, we require that for all $\lambda \in \mathbb{N}$ and $(\mathsf{pk}, \mathsf{sk}, \mathsf{td}) \leftarrow \mathsf{CKG}(1^\lambda)$, the following conditions are satisfied:

1. $\mathsf{CDec}(\mathsf{pk}, \mathsf{sk}, i, \mathsf{CEnc}(\mathsf{pk}, i, \mathsf{sk}[i], \mathsf{m})) = \mathsf{m}$ holds for all $i \in [\ell_{\mathsf{sk}}]$ and m.

2. $\widehat{\mathsf{CDec}}(\mathsf{td}, i, v, \mathsf{CEnc}(\mathsf{pk}, i, v, \mathsf{m})) = \mathsf{m}$ holds for all $(i, v) \in [\ell_{\mathsf{sk}}] \times \{0, 1\}$ and m.

3. $\mathsf{CDec}(\mathsf{pk}, \mathsf{sk}, i, \mathsf{ct}) = \widehat{\mathsf{CDec}}(\mathsf{td}, i, \mathsf{sk}[i], \mathsf{ct})$ holds for all $i \in [\ell_{\mathsf{sk}}]$ and ct (not necessarily in the support of CEnc).

4. $\mathsf{CSDec}(\mathsf{sk}, \mathsf{CSEnc}(\mathsf{sk}, \mathsf{m})) = \mathsf{m}$ holds for all m.

Note that the first condition of correctness ensures that $(\mathsf{CKG}, \mathsf{CEnc}, \mathsf{CDec})$ constitutes a TE scheme when td in the output of CKG is discarded. We also remark that the third condition of correctness is required to hold for all values of ct not necessarily in the support of CEnc. Looking ahead, this property plays an important role in our construction of KDM-CCA secure PKE in Section 7.

Security Definitions for CTE. For a CTE scheme, we require two security notions: *security against the receiver* and *special weak circular security (in the multi-key setting)*.[10] The former is defined in exactly the same way as that for TE, except that we just discard and ignore the trapdoor td generated from CKG. Thus, we omit its formal description.

The latter security notion, special weak circular security, requires that the additional secret-key encryption/decryption algorithms $(\mathsf{CSEnc}, \mathsf{CSDec})$ satisfy a weak form of circular security in the multi-key setting. Specifically, in the n-key setting, we require that messages encrypted by CSEnc be hidden even in the presence of public keys $\{\mathsf{pk}^s\}_{s \in [n]}$, trapdoors $\{\mathsf{td}^s\}_{s \in [n]}$, and *encryptions of a "key cycle"* $\{\mathsf{CSEnc}(\mathsf{sk}^s, \mathsf{sk}^{(s \bmod n)+1})\}_{s \in [n]}$. We call it *weak* since except for giving $\{(\mathsf{pk}^s, \mathsf{td}^s)\}_{s \in [n]}$ to an adversary, our definition is the same as the definition of weak circular security formalized by Cash, Green, and Hohenberger [CGH12].

Formally, let $n = n(\lambda)$ be a polynomial. For a CTE scheme $(\mathsf{CKG}, \mathsf{CEnc}, \mathsf{CDec}, \widehat{\mathsf{CDec}}, \mathsf{CSEnc}, \mathsf{CSDec})$, n, and an adversary $\mathcal{A}$, consider the experiment $\mathsf{Expt}^{\mathsf{sp-wcirc}}_{\mathsf{CTE}, \mathcal{A}, n}(\lambda)$ described in Figure 6. Note that in the experiment, $\mathcal{O}_{\mathsf{CSEnc}}$ is an ordinary (challenge) encryption oracle. Thus, except for the encryptions of a key cycle $\{\mathsf{CSEnc}(\mathsf{sk}^s, \mathsf{sk}^{(s \bmod n)+1})\}_{s \in [n]}$, $\mathcal{A}$ is *not* allowed to directly obtain encryptions of key-dependent messages.

Definition 6 (Special Weak Circular Security) *Let $n = n(\lambda)$ be a polynomial. We say that a CTE scheme* CTE *satisfies* special weak circular security in the n-key setting (special weak CIRC$^{(n)}$ security) *if for all PPT adversaries $\mathcal{A}$, we have* $\mathsf{Adv}^{\mathsf{sp-wcirc}}_{\mathsf{CTE}, \mathcal{A}, n}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathsf{sp-wcirc}}_{\mathsf{CTE}, \mathcal{A}, n}(\lambda) = 1] - 1/2| = \mathsf{negl}(\lambda)$.

Relation to Augmented TE. As mentioned earlier, Barak et al. [BHHI10] introduced the notion of *augmented TE*, and used it to construct a $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CPA-secure PKE scheme for any polynomials $n = n(\lambda)$ and $\mathsf{size} = \mathsf{size}(\lambda)$. An augmented TE scheme is a TE scheme with the additional *public-key* encryption/decryption algorithms, for which Barak et al. assumed circular security in the n-key setting. (Their definition requires that encryptions of a key cycle of length n are indistinguishable from encryptions of some fixed messages.)

We observe that their security proof goes through even if (1) the additional encryption/decryption algorithms are of secret-key, and (2) we only require *weak* circular security in the n-key setting

[10] We can also consider security against outsiders for CTE. However, we do not formalize it since we need not use it in our construction of KDM-CCA secure PKE.

```
$\mathsf{Expt}^{\mathsf{sp-wcirc}}_{\mathsf{CTE},\mathcal{A},n}(\lambda)$ :
    $\forall s \in [n] : (\mathsf{pk}^s, \mathsf{sk}^s, \mathsf{td}^s) \leftarrow \mathsf{CKG}(1^\lambda)$
    $(\widetilde{\mathsf{ct}}^s)_{s\in[n]} \leftarrow \mathsf{EncCycle}((\mathsf{sk}^s)_{s\in[n]})$
    $b \xleftarrow{\mathsf{r}} \{0,1\}$
    $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{CSEnc}}(\cdot,\cdot,\cdot)}((\mathsf{pk}^s, \mathsf{td}^s, \widetilde{\mathsf{ct}}^s)_{s\in[n]})$
    Return $(b' \overset{?}{=} b)$.
```

```
$\mathsf{EncCycle}((\mathsf{sk}^s)_{s\in[n]})$ :
    $\forall s \in [n] : \widetilde{\mathsf{ct}}^s \leftarrow \mathsf{CSEnc}(\mathsf{sk}^s, \mathsf{sk}^{(s \bmod n)+1})$
    Return $(\widetilde{\mathsf{ct}}^s)_{s\in[n]}$.

$\mathcal{O}_{\mathsf{CSEnc}}(\alpha, \mathsf{m}_0, \mathsf{m}_1)$ :    // $\alpha \in [n]$, $|\mathsf{m}_0| = |\mathsf{m}_1|$
    $\widetilde{\mathsf{ct}} \leftarrow \mathsf{CSEnc}(\mathsf{sk}^\alpha, \mathsf{m}_b)$
    Return $\widetilde{\mathsf{ct}}$.
```

Figure 6: The experiment for defining special weak circular security for a CTE scheme.

[CGH12], which requires that IND-CPA security holds in the presence of encryptions of a key cycle of length n.

Our formalization for CTE is based on these observations, but CTE has an additional syntactical extension involving a trapdoor generated in the key generation algorithm, together with the additional correctness requirements. This plays an important role in the security proof for our $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA secure PKE scheme presented in Section 7. We also remark that we do not require CTE to satisfy security against outsiders, while it is necessary for augmented TE used in the construction of KDM-CPA secure PKE in [BHHI10]. Our construction of KDM-CCA secure PKE does not require security against outsiders for the underlying CTE scheme because of the other building blocks. (See Section 7.)

6.2 Construction

Let $n = n(\lambda)$ be a polynomial for which we would like our CTE scheme CTE to satisfy special weak CIRC$^{(n)}$ security. Let $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and $\mathsf{SKE} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$ be PKE and SKE schemes as in Section 4, respectively, where we now require SKE to be CIRC$^{(n)}$ secure.

Our construction of a CTE scheme $\mathsf{CTE} = (\mathsf{CKG}, \mathsf{CEnc}, \mathsf{CDec}, \widehat{\mathsf{CDec}}, \mathsf{CSEnc}, \mathsf{CSDec})$ based on PKE and SKE, is a simple extension of our TE scheme $\mathsf{TE} = (\mathsf{TKG}, \mathsf{TEnc}, \mathsf{TDec})$ presented in Section 4. Specifically, each algorithm of CTE operates as follows:

- CKG computes a public/secret key pair $(\mathsf{PK}, \mathsf{SK})$ in exactly the same way as TKG, and additionally outputs $\mathsf{td} := (\mathsf{pk}_{i,v}, \mathsf{sk}_{i,v})_{i\in[\ell_{\mathsf{k}}], v\in\{0,1\}}$ as a trapdoor.
- CEnc and CDec are exactly TEnc and TDec, respectively.
- $\widehat{\mathsf{CDec}}(\mathsf{td}, i, v, \mathsf{ct}) := \mathsf{Dec}(\mathsf{pk}_{i,v}, \mathsf{sk}_{i,v}, \mathsf{ct})$.
- CSEnc and CSDec use E and D to encrypt/decrypt a message/ciphertext in a bit-wise fashion. More specifically, $\mathsf{CSEnc}(\mathsf{SK} = \mathsf{k}, \mathsf{m} \in \{0,1\}^\mu)$ outputs $\widetilde{\mathsf{ct}} = (\widetilde{\mathsf{ct}}_t)_{t\in[\mu]}$, where $\widetilde{\mathsf{ct}}_t \leftarrow \mathsf{E}(\mathsf{k}, \mathsf{m}[t])$ for each $t \in [\mu]$; $\mathsf{CDec}(\mathsf{SK} = \mathsf{k}, \widetilde{\mathsf{ct}} = (\widetilde{\mathsf{ct}}_t)_{t\in[\mu]})$ computes $\mathsf{m}[t] \leftarrow \mathsf{D}(\mathsf{k}, \widetilde{\mathsf{ct}}_t)$ for each $t \in [\mu]$, and outputs m.

See Figure 5 for the figure version of our construction CTE.

Correctness. The first condition of correctness is exactly the same as the correctness for TE. The third condition of correctness holds because sk' computed in $\mathsf{CDec}(\mathsf{PK}, \mathsf{SK} = \mathsf{k}, i, \cdot)$ is $\mathsf{sk}_{i,\mathsf{k}[i]}$ as we saw for the correctness of TE. The second and fourth conditions of correctness are trivially satisfied because of the correctness of PKE and SKE, respectively.

Security. The following theorems guarantee that CTE satisfies the two kinds of security notions for CTE. We omit the proof of Theorem 11 since it is exactly the same as that of Theorem 5.

```
$\mathsf{CKG}(1^\lambda)$ :
  $\mathsf{k} \leftarrow \mathsf{K}(1^\lambda)$
  $\forall i \in [\ell_{\mathsf{k}}]$:
    $\forall v \in \{0,1\}$:  $\mathsf{sk}_{i,v} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{sk}}}$;  $\mathsf{pk}_{i,v} \leftarrow \mathsf{KG}(\mathsf{sk}_{i,v})$
    $\forall j \in [\ell_{\mathsf{sk}}]$:
      $\mathsf{e}_{i,j} \leftarrow \begin{cases} \mathtt{flip} & \text{if } (\mathsf{sk}_{i,0}[j], \mathsf{sk}_{i,1}[j]) = (1,0) \\ \mathsf{E}(\mathsf{k}, \mathsf{sk}_{i,\mathsf{k}[i]}[j]) & \text{otherwise} \end{cases}$
  $\mathsf{PK} \leftarrow (\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}, \mathsf{e}_{i,1}, \ldots, \mathsf{e}_{i,\ell_{\mathsf{sk}}})_{i \in [\ell_{\mathsf{k}}]}$;   $\mathsf{SK} \leftarrow \mathsf{k}$;   $\mathsf{td} \leftarrow (\mathsf{sk}_{i,v}, \mathsf{pk}_{i,v})_{i \in [\ell_{\mathsf{k}}], v \in \{0,1\}}$
  Return $(\mathsf{PK}, \mathsf{SK}, \mathsf{td})$.
```

```
$\mathsf{CEnc}(\mathsf{PK}, i, v, \mathsf{m})$ :
  $(\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}, \mathsf{e}_{i,1}, \ldots, \mathsf{e}_{i,\ell_{\mathsf{sk}}})_{i \in [\ell_{\mathsf{k}}]} \leftarrow \mathsf{PK}$
  Return $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}_{i,v}, \mathsf{m})$.
```

```
$\widetilde{\mathsf{CDec}}(\mathsf{td}, i, v, \mathsf{ct})$ :
  $(\mathsf{pk}_{i,v}, \mathsf{sk}_{i,v})_{i \in [\ell_{\mathsf{k}}], v \in \{0,1\}} \leftarrow \mathsf{td}$
  Return $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{pk}_{i,v}, \mathsf{sk}_{i,v}, \mathsf{ct})$.
```

```
$\mathsf{CDec}(\mathsf{PK}, \mathsf{SK} = \mathsf{k}, i, \mathsf{ct})$ :
  $(\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1}, \mathsf{e}_{i,1}, \ldots, \mathsf{e}_{i,\ell_{\mathsf{sk}}})_{i \in [\ell_{\mathsf{k}}]} \leftarrow \mathsf{PK}$
  $\forall j \in [\ell_{\mathsf{sk}}]$:
    $\mathsf{sk}'[j] \leftarrow \begin{cases} 1 \oplus \mathsf{k}[i] & \text{if } \mathsf{e}_{i,j} = \mathtt{flip} \\ \mathsf{D}(\mathsf{k}, \mathsf{e}_{i,j}) & \text{otherwise} \end{cases}$
  Return $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{pk}_{i,\mathsf{k}[i]}, \mathsf{sk}', \mathsf{ct})$.
```

```
$\mathsf{CSEnc}(\mathsf{SK} = \mathsf{k}, \mathsf{m})$ :
  $\forall t \in [|\mathsf{m}|] : \widetilde{\mathsf{ct}}_t \leftarrow \mathsf{E}(\mathsf{k}, \mathsf{m}[t])$
  Return $\widetilde{\mathsf{ct}} = (\widetilde{\mathsf{ct}}_t)_{t \in [|\mathsf{m}|]}$.
```

```
$\mathsf{CSDec}(\mathsf{SK} = \mathsf{k}, \widetilde{\mathsf{ct}})$ :
  Parse $\widetilde{\mathsf{ct}}$ as $(\widetilde{\mathsf{ct}}_t)_{t \in [\mu]}$ for some $\mu \in \mathbb{N}$
    where each $\widetilde{\mathsf{ct}}_t$ is a ciphertext of $\mathsf{SKE}$.
  $\forall t \in [\mu] : \mathsf{m}[t] \leftarrow \mathsf{D}(\mathsf{k}, \widetilde{\mathsf{ct}}_t)$
  Return $\mathsf{m}$.
```

Figure 7: The construction of a CTE scheme CTE from a circular secure bit-SKE scheme SKE and a weakly noisy-leakage-resilient PKE scheme PKE.

Theorem 11 *If* PKE *is weakly* $(0.6\ell_{\mathsf{sk}})$*-noisy-leakage-resilient, then* CTE *satisfies security against the receiver.*

Theorem 12 *Let* $n = n(\lambda)$ *be a polynomial. If* SKE *is CIRC*$^{(n)}$ *secure, then* CTE *satisfies special weak CIRC*$^{(n)}$ *security.*

Proof Sketch of Theorem 12. This is straightforward to see by noting that CSEnc directly uses E to encrypt a given message in a bit-wise fashion, and the trapdoor td consists only of key pairs of the underlying PKE scheme PKE and thus is independent of a secret key $\mathsf{SK} = \mathsf{k}$.

More specifically, for $s \in [n]$, let $\mathsf{SK}^s = \mathsf{k}^s$ denote the s-th secret key. Then, consider a modified security experiment, which proceeds similarly to the experiment for the special weak CIRC$^{(n)}$ security of CTE, except that for every $s \in [n]$, all invocations of $\mathsf{E}(\mathsf{k}^s, \cdot)$ (which include those during the execution of $\mathsf{EncCycle}((\mathsf{SK}^s = \mathsf{k}^s)_{s \in [n]})$, those during the execution of $(\mathsf{PK}^s, \mathsf{SK}^s = \mathsf{k}^s, \mathsf{td}^s) \leftarrow \mathsf{CKG}(1^\lambda)$, and those for encryption queries from an adversary) are replaced with $\mathsf{E}(\mathsf{k}^s, 0)$. Note that this modified experiment is independent of the challenge bit b, and thus any adversary has zero advantage. Furthermore, by the CIRC$^{(n)}$ security of SKE, for any PPT adversary, its advantage in the original special weak CIRC$^{(n)}$ security experiment is negligibly close to that in the modified experiment. □ (**Theorem 12**)

7 KDM-CCA Security in the Multi-key Setting

In this section, we show the completeness of circular security in the multi-key setting. Specifically, we show the following theorem:

Theorem 13 *Let* $n = n(\lambda)$ *be a polynomial. Assume that there exist an IND-CPA secure PKE scheme and a CIRC*$^{(n)}$ *secure bit-SKE scheme. Then, for any polynomial* $\mathsf{size} = \mathsf{size}(\lambda)$*, there exists a* $\mathcal{B}_{\mathsf{size}}$*-KDM*$^{(n)}$*-CCA secure PKE scheme.*

Note that this result improves the result by Kitagawa and Matsuda [KM19] (recalled as Theorem 3) in terms of the strength of assumptions and the number of keys.

As explained earlier, we will show the above theorem by constructing a $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA secure PKE scheme from the building blocks that are all implied by an IND-CPA secure PKE scheme and a CIRC$^{(n)}$ secure bit-SKE scheme. Our construction can be seen as combining the construction ideas from the bounded-KDM$^{(n)}$-CPA secure PKE scheme from an augmented TE scheme by Barak et al. [BHHI10] and the bounded-KDM$^{(1)}$-CCA secure PKE scheme from an IND-CPA secure PKE scheme and a projection-KDM$^{(1)}$-CPA secure SKE scheme by Kitagawa and Matsuda [KM19]. The latter construction in fact uses an IND-CCA secure PKE scheme, a garbling scheme, and a reusable DV-NIZK argument system as additional building blocks, which are implied by the assumption used in [KM19]. Construction-wise, roughly speaking, our construction is obtained by replacing the underlying IND-CPA secure scheme of the Kitagawa-Matsuda construction with a CTE scheme.

Construction. To construct a $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA secure PKE scheme, we use the following building blocks all of which are implied by the combination of an IND-CPA secure PKE scheme and a CIRC$^{(n)}$ secure SKE scheme:

- Let $\mathsf{CTE} = (\mathsf{CKG}, \mathsf{CEnc}, \mathsf{CDec}, \widehat{\mathsf{CDec}}, \mathsf{CSEnc}, \mathsf{CSDec})$ be a CTE scheme whose secret key length is ℓ_{sk}. Let $\ell_{\widetilde{\mathsf{e}}}$ denote the length of a ciphertext when encrypting a message of length ℓ_{sk} by using CSEnc. We denote the randomness space of CEnc by $\mathcal{R}$.

- Let $\mathsf{PKE}_{\mathsf{cca}} = (\mathsf{KG}_{\mathsf{cca}}, \mathsf{Enc}_{\mathsf{cca}}, \mathsf{Dec}_{\mathsf{cca}})$ be an IND-CCA secure PKE scheme.

- Let $\mathsf{GC} = (\mathsf{Garble}, \mathsf{Eval}, \mathsf{Sim})$ be a garbling scheme for circuits.

- Let $\mathsf{DVNIZK} = (\mathsf{DVKG}, \mathsf{P}, \mathsf{V})$ be a reusable DV-NIZK argument system for the following NP language L:[11]

$$L = \left\{ \left(\mathsf{pk}, (\mathsf{ct}_{i,v})_{i \in [\ell_{\mathsf{sk}}], v \in \{0,1\}}\right) \;\middle|\; \exists (\mathsf{lab}_i, \mathsf{r}_{i,0}, \mathsf{r}_{i,1})_{i \in [\ell_{\mathsf{sk}}]} \text{ s.t. } \begin{array}{l} \forall (i,v) \in [\ell_{\mathsf{sk}}] \times \{0,1\} : \\ \mathsf{ct}_{i,v} = \mathsf{CEnc}(\mathsf{pk}, i, v, \mathsf{lab}_i; \mathsf{r}_{i,v}) \end{array} \right\}.$$

Let $\mu = \mu(\lambda)$ be a polynomial that denotes the length of messages to be encrypted by our constructed PKE scheme. Let $n = n(\lambda)$ and $\mathsf{size} = \mathsf{size}(\lambda) \geq \max\{n \cdot \ell_{\mathsf{sk}}, \mu\}$ be polynomials for which we wish to achieve $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA security. Finally, let $\mathsf{pad} = O(n \cdot (|\mathsf{CSDec}| + \ell_{\widetilde{\mathsf{e}}}) + \mathsf{size}) \geq \mathsf{size}$ be the size parameter for the underlying garbling scheme (which is the size of a circuit that will be specified in the security proof), where $|\mathsf{CSDec}|$ denotes the size of the circuit computing CSDec.

Using these ingredients, we construct our proposed PKE scheme $\mathsf{PKE}_{\mathsf{kdm}} = (\mathsf{KG}_{\mathsf{kdm}}, \mathsf{Enc}_{\mathsf{kdm}}, \mathsf{Dec}_{\mathsf{kdm}})$ whose message space is $\{0,1\}^\mu$ as described in Figure 8.

Correctness. The correctness of $\mathsf{PKE}_{\mathsf{kdm}}$ follows from that of the building blocks. Specifically, let $(\mathsf{PK}, \mathsf{SK}) = ((\mathsf{pk}, \mathsf{pk}_{\mathsf{cca}}, \mathsf{pk}_{\mathsf{dv}}, \widetilde{\mathsf{ct}}), \mathsf{sk})$ be a key pair output by $\mathsf{KG}_{\mathsf{kdm}}$, let $\mathsf{m} \in \{0,1\}^\mu$ be any message, and let $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{kdm}}(\mathsf{PK}, \mathsf{m})$ be an honestly generated ciphertext. Due to the correctness of CTE, $\mathsf{PKE}_{\mathsf{cca}}$, and DVNIZK, each decryption/verification done in the execution of $\mathsf{Dec}_{\mathsf{kdm}}(\mathsf{PK}, \mathsf{SK}, \mathsf{CT})$ never fails, and just before the final step of $\mathsf{Dec}_{\mathsf{kdm}}$, the decryptor can recover a garbled circuit $\widetilde{\mathsf{Q}}$ and the labels $(\mathsf{lab}_i)_i$, which is generated as $(\widetilde{\mathsf{Q}}, (\mathsf{lab}_i)_i) \leftarrow \mathsf{Sim}(1^\lambda, \mathsf{pad}, \mathsf{m})$. Then, by the correctness of GC, we have $\mathsf{Eval}(\widetilde{\mathsf{Q}}, (\mathsf{lab}_i)_i) = \mathsf{m}$.

[11] Intuitively, a statement $(\mathsf{pk}, (\mathsf{ct}_{i,v})_{i \in [\ell_{\mathsf{sk}}], v \in \{0,1\}})$ of the language L constitutes a $(\ell_{\mathsf{sk}} \times 2)$-matrix of ciphertexts such that the pair $(\mathsf{ct}_{i,0}, \mathsf{ct}_{i,1})$ in the i-th row encrypt the same plaintext lab_i for each $i \in [\ell_{\mathsf{sk}}]$.

```
$\mathsf{KG_{kdm}}(1^\lambda)$ :
    $(\mathsf{pk}, \mathsf{sk}, \mathsf{td}) \leftarrow \mathsf{CKG}(1^\lambda)$
    $(\mathsf{pk_{cca}}, \mathsf{sk_{cca}}) \leftarrow \mathsf{KG_{cca}}(1^\lambda)$
    $(\mathsf{pk_{dv}}, \mathsf{sk_{dv}}) \leftarrow \mathsf{DVKG}(1^\lambda)$
    $\widetilde{\mathsf{ct}} \leftarrow \mathsf{CSEnc}(\mathsf{sk}, (\mathsf{sk_{cca}}, \mathsf{sk_{dv}}))$
    $\mathsf{PK} \leftarrow (\mathsf{pk}, \mathsf{pk_{cca}}, \mathsf{pk_{dv}}, \widetilde{\mathsf{ct}})$;   $\mathsf{SK} \leftarrow \mathsf{sk}$
    Return $(\mathsf{PK}, \mathsf{SK})$.
```

```
$\mathsf{Dec_{kdm}}(\mathsf{PK}, \mathsf{SK} = \mathsf{sk}, \mathsf{CT})$ :  $^{(\star)}$
    $(\mathsf{pk}, \mathsf{pk_{cca}}, \mathsf{pk_{dv}}, \widetilde{\mathsf{ct}}) \leftarrow \mathsf{PK}$
    $(\mathsf{sk_{cca}}, \mathsf{sk_{dv}}) \leftarrow \mathsf{CSDec}(\mathsf{sk}, \widetilde{\mathsf{ct}})$
    $(\widetilde{\mathsf{Q}}, (\mathsf{ct}_{i,v})_{i,v}, \pi) \leftarrow \mathsf{Dec_{cca}}(\mathsf{pk_{cca}}, \mathsf{sk_{cca}}, \mathsf{CT})$
    $x \leftarrow (\mathsf{pk}, (\mathsf{ct}_{i,v})_{i,v})$
    If $\mathsf{V}(\mathsf{sk_{dv}}, x, \pi) = \mathsf{reject}$ then return $\bot$.
    $\forall i \in [\ell_{\mathsf{sk}}] : \mathsf{lab}_i \leftarrow \mathsf{CDec}(\mathsf{pk}, \mathsf{sk}, i, \mathsf{ct}_{i,\mathsf{sk}[i]})$
    Return $\mathsf{m} \leftarrow \mathsf{Eval}(\widetilde{\mathsf{Q}}, (\mathsf{lab}_i)_i)$.
```

```
$\mathsf{Enc_{kdm}}(\mathsf{PK}, \mathsf{m})$ :
    $(\mathsf{pk}, \mathsf{pk_{cca}}, \mathsf{pk_{dv}}, \widetilde{\mathsf{ct}}) \leftarrow \mathsf{PK}$
    $(\widetilde{\mathsf{Q}}, (\mathsf{lab}_i)_i) \leftarrow \mathsf{Sim}(1^\lambda, \mathsf{pad}, \mathsf{m})$
    $\forall (i, v) \in [\ell_{\mathsf{sk}}] \times \{0,1\}$ :
        $\mathsf{r}_{i,v} \xleftarrow{\mathsf{r}} \mathcal{R}$
        $\mathsf{ct}_{i,v} \leftarrow \mathsf{CEnc}(\mathsf{pk}, i, v, \mathsf{lab}_i; \mathsf{r}_{i,v})$
    $x \leftarrow (\mathsf{pk}, (\mathsf{ct}_{i,v})_{i,v})$
    $w \leftarrow (\mathsf{lab}_i, \mathsf{r}_{i,0}, \mathsf{r}_{i,1})_i$
    $\pi \leftarrow \mathsf{P}(\mathsf{pk_{dv}}, x, w)$
    $\mathsf{CT} \leftarrow \mathsf{Enc_{cca}}(\mathsf{pk_{cca}}, (\widetilde{\mathsf{Q}}, (\mathsf{ct}_{i,v})_{i,v}, \pi))$
    Return $\mathsf{CT}$.
```

Figure 8: The construction of a $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA secure PKE scheme $\mathsf{PKE_{kdm}}$ from a CTE scheme CTE, an IND-CCA secure PKE scheme $\mathsf{PKE_{cca}}$, a garbling scheme for circuits GC, and a reusable DV-NIZK argument system DVNIZK. The notations like $(X_{i,v})_{i,v}$ and $(X_i)_i$ are abbreviations for $(X_{i,v})_{i \in [\ell_{\mathsf{sk}}], v \in \{0,1\}}$ and $(X_i)_{i \in [\ell_{\mathsf{sk}}]}$, respectively. $^{(\star)}$ If CSDec, CDec, or $\mathsf{Dec_{cca}}$ returns $\bot$, then $\mathsf{Dec_{kdm}}$ returns $\bot$ and terminate.

Security. The following theorem guarantees the $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA security of $\mathsf{PKE_{kdm}}$. Combined with Theorems 9, 10, 11, and 12, it implies Theorem 13.

Theorem 14 *Let $n = n(\lambda)$, $\mu = \mu(\lambda)$, and $\mathsf{size} = \mathsf{size}(\lambda) \geq \max\{n \cdot \ell_{\mathsf{sk}}, \mu\}$ be any polynomials. Also, let $\mathsf{pad} = O(n \cdot (|\mathsf{CSDec}| + \ell_{\widetilde{\mathsf{e}}}) + \mathsf{size}) \geq \mathsf{size}$ (which is the size of the circuit described in Figure 9), where $|\mathsf{CSDec}|$ denotes the size of the circuit computing CSDec. Assume that CTE satisfies security against the receiver and special weak CIRC$^{(n)}$ security, $\mathsf{PKE_{cca}}$ is IND-CCA secure, GC is a secure garbling scheme, and DVNIZK is a reusable DV-NIZK argument system (satisfying soundness and zero-knowledge) for the NP language L. Then, $\mathsf{PKE_{kdm}}$ is $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA secure.*

Overview of the Proof. Before showing the formal proof, we give its overview. The proof uses a sequence of games argument. The first game is the original $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA experiment regarding $\mathsf{PKE_{kdm}}$. Let $\mathcal{A}$ be a PPT adversary, and for $s \in [n]$, let $(\mathsf{PK}^s = (\mathsf{pk}^s, \mathsf{pk}^s_{\mathsf{cca}}, \mathsf{pk}^s_{\mathsf{dv}}, \widetilde{\mathsf{ct}}^s), \mathsf{SK}^s = \mathsf{sk}^s)$ denote the s-th public/secret key pair.

We first invoke the zero-knowledge of DVNIZK to change the security game so that the simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ is used to generate each $(\mathsf{pk}^s_{\mathsf{dv}}, \mathsf{sk}^s_{\mathsf{dv}})$ at key generation, and generate π in the response to KDM-encryption queries.

Next, we deal with the KDM-encryption queries (α, f_0, f_1), and make the behavior of the KDM-encryption oracle (essentially) independent of the secret keys $\{\mathsf{sk}^s\}_{s \in [n]}$. If there existed only a single key pair $(\mathsf{PK}, \mathsf{SK} = \mathsf{sk})$, then we could change the generation of the CTE-ciphertexts $(\mathsf{ct}_{i,v})_{i,v}$ in the KDM-encryption oracle so that we garble the KDM function f_b by $(\widetilde{\mathsf{Q}}, (\mathsf{lab}_{i,v})_{i,v}) \leftarrow \mathsf{Garble}(1^\lambda, f_b)$ and then encrypt $\mathsf{lab}_{i,v}$ by $\mathsf{ct}_{i,v} \leftarrow \mathsf{CEnc}(\mathsf{pk}^s, i, v, \mathsf{lab}_{i,v})$ for every $(i, v) \in [\ell_{\mathsf{sk}}] \times \{0,1\}$. Since $\mathsf{Eval}(\widetilde{\mathsf{Q}}, (\mathsf{lab}_{i,\mathsf{sk}[i]})_{i \in [\ell_{\mathsf{sk}}]}) = f_b(\mathsf{sk})$, this can go unnoticed by $\mathcal{A}$ due to the security of GC and the security against the receiver of CTE, and the behavior of the resulting KDM-encryption oracle becomes independent of the secret key sk. However, we cannot take this rather simple approach in the multi-key setting, since the KDM-function f_b here is a function that takes all keys $\{\mathsf{sk}^s\}_{s \in [n]}$ as input, while we need to garble a circuit that takes a single key sk^α as input. Here, we rely on the clever technique of Barak et al. [BHHI10] to transform the KDM function f_b to a circuit Q so that $\mathsf{Q}(\mathsf{sk}^\alpha) = f_b((\mathsf{sk}^s)_{s\in|n|})$ holds, by using encryptions of the key cycle $\{\widetilde{\mathsf{e}}^s = \mathsf{CSEnc}(\mathsf{sk}^s, \mathsf{sk}^{(s \bmod n)+1})\}_{s\in[n]}$. Specifically, Q has α, f_b, and $\{\widetilde{\mathsf{e}}^s\}_{s\in[n]}$ hardwired, and it on input sk^α decrypts the encryptions of the key cycle one-by-one to recover all keys $\{\mathsf{sk}^s\}_{s\in[n]}$ and then outputs $f_b((\mathsf{sk}^s)_{s\in[n]})$. Then, we can garble Q instead of garbling f_b directly, and the argument goes similarly to the above. This change necessitates that the subsequent games generate the encryptions of the key cycle.

Then, we deal with the decryption queries (α, CT), and make the behavior of the decryption oracle independent of the secret keys $\{\mathsf{sk}^s\}_{s\in[n]}$. To achieve this, notice that the only essential part that we need to use the secret key sk^α in the decryption procedure is the step of executing $\mathsf{lab}_i \leftarrow \mathsf{CDec}(\mathsf{pk}^\alpha, \mathsf{sk}^\alpha, \mathsf{ct}_{i,\mathsf{sk}[i]})$ for every $i \in [\ell_{\mathsf{sk}}]$. To eliminate the dependency on sk^α in this step, in the next game we replace the above step with $\mathsf{lab}_i \leftarrow \widehat{\mathsf{CDec}}(\mathsf{td}^\alpha, i, \mathsf{sk}^\alpha[i], \mathsf{ct}_{i,\mathsf{sk}^\alpha[i]})$ for every $i \in [\ell_{\mathsf{sk}}]$. This makes no change in the behavior of the decryption oracle due to the third condition of the correctness of CTE. Next, we further change this step to always decrypt the "0-side" ciphertext $\mathsf{ct}_{i,0}$ as $\mathsf{lab}_i \leftarrow \widehat{\mathsf{CDec}}(\mathsf{td}^\alpha, i, 0, \mathsf{ct}_{i,0})$ for every $i \in [\ell_{\mathsf{sk}}]$. Now the behavior of the decryption oracle becomes independent of the secret keys $\{\mathsf{sk}^s\}_{s\in[n]}$. The behavior of the decryption oracle could differ between the change only if $\widehat{\mathsf{CDec}}(\mathsf{td}^\alpha, i^*, 0, \mathsf{ct}_{i^*,0}) \neq \widehat{\mathsf{CDec}}(\mathsf{td}^\alpha, i^*, 1, \mathsf{ct}_{i^*,1})$ holds for some $i^* \in [\ell_{\mathsf{sk}}]$ and yet the proof π recovered from CT is valid. Let us call such a query a bad decryption query. If $\mathcal{A}$ does not make a bad decryption query, this change of the behavior of the decryption oracle cannot be noticed by $\mathcal{A}$. Similarly to [KM19], we bound the probability of a bad query occurring to be negligible using a deferred analysis technique and postpone to bound it in a later (in fact the final) game, together with the second correctness condition of CTE. See the formal proof for this argument.

Now, since the behavior of the KDM-encryption and decryption oracles become independent of the secret keys $\{\mathsf{sk}^s\}_{s\in[n]}$, the remaining steps in which we use the secret keys are to generate $\{\widetilde{\mathsf{ct}}^s\}_{s\in[n]}$ in public keys, and to generate the encryptions of the key cycle $\{\widetilde{\mathsf{e}}^s\}_{s\in[n]}$. Then, we can rely on the special weak CIRC$^{(n)}$ security of CTE to ensure that $\widetilde{\mathsf{ct}}^s$ is indistinguishable from an encryption of a garbage that contains no information on $(\mathsf{sk}^s_{\mathsf{cca}}, \mathsf{sk}^s_{\mathsf{dv}})$ in the presence of the trapdoors $\{\mathsf{td}^s\}_{s\in[n]}$ and the encryptions of the key cycle $\{\widetilde{\mathsf{e}}^s\}_{s\in[n]}$. Finally, we invoke the IND-CCA security of $\mathsf{PKE}_{\mathsf{cca}}$ to conclude that $\mathcal{A}$'s advantage in the final game is negligible.

Proof of Theorem 14. Let $n = n(\lambda)$ be an arbitrary polynomial that denotes the number of key pairs. Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA security of $\mathsf{PKE}_{\mathsf{kdm}}$. For simplicity and without loss of generality, we assume that $\mathcal{A}$ does not make a decryption query (α, CT) such that $(\alpha, \mathsf{CT}) \in L_{\mathsf{kdm}}$. We proceed the proof via a sequence of games argument using nine games. For every $t \in [9]$, let $\mathtt{SUC}_t$ be the event that $\mathcal{A}$ succeeds in guessing the challenge bit b in Game t. The final game (Game 9) is used only to bound the probability of a bad event introduced later.

Game 1: This is the original $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA game regarding $\mathsf{PKE}_{\mathsf{kdm}}$. Thus, we have $\mathsf{Adv}^{\mathsf{kdmcca}}_{\mathsf{PKE}_{\mathsf{kdm}},\mathcal{B}_{\mathsf{size}},\mathcal{A},n}(\lambda) = 2 \cdot |\Pr[\mathtt{SUC}_1] - 1/2|$.

The detailed description of the game is as follows:

Generate a key pair $(\mathsf{PK}^s, \mathsf{SK}^s)$ of $\mathsf{PKE}_{\mathsf{kdm}}$ for every $s \in [n]$ as follows:

1. Compute $(\mathsf{pk}^s, \mathsf{sk}^s, \mathsf{td}^s) \leftarrow \mathsf{CKG}(1^\lambda)$.
2. Compute $(\mathsf{pk}^s_{\mathsf{cca}}, \mathsf{sk}^s_{\mathsf{cca}}) \leftarrow \mathsf{KG}_{\mathsf{cca}}(1^\lambda)$.
3. Compute $(\mathsf{pk}^s_{\mathsf{dv}}, \mathsf{sk}^s_{\mathsf{dv}}) \leftarrow \mathsf{DVKG}(1^\lambda)$.
4. Compute $\widetilde{\mathsf{ct}}^s \leftarrow \mathsf{CSEnc}(\mathsf{sk}^s, (\mathsf{sk}^s_{\mathsf{cca}}, \mathsf{sk}^s_{\mathsf{dv}}))$.

5. Set $\mathsf{PK}^s := (\mathsf{pk}^s, \mathsf{pk}^s_{\mathsf{cca}}, \mathsf{pk}^s_{\mathsf{dv}}, \widetilde{\mathsf{ct}}^s)$ and $\mathsf{SK}^s := \mathsf{sk}^s$.

Then, choose the challenge bit $b \xleftarrow{\mathsf{r}} \{0,1\}$, generate an empty list L_{kdm}, and run $\mathcal{A}((\mathsf{PK}^s)_{s\in[n]})$. From here on, $\mathcal{A}$ may start making KDM-encryption and decryption queries.

- The KDM-encryption oracle responds to $\mathcal{A}$'s query $(\alpha, f_0, f_1) \in [n] \times (\mathcal{B}_{\mathsf{size}})^2$ as follows:
  1. Compute $(\widetilde{\mathsf{Q}}, (\mathsf{lab}_i)_i) \leftarrow \mathsf{Sim}(1^\lambda, \mathsf{pad}, f_b((\mathsf{sk}^s)_{s\in[n]}))$.
  2. For every $i \in [\ell_{\mathsf{sk}}]$ and $v \in \{0,1\}$, pick $\mathsf{r}_{i,v} \xleftarrow{\mathsf{r}} \mathcal{R}$ and compute $\mathsf{ct}_{i,v} \leftarrow \mathsf{CEnc}(\mathsf{pk}^\alpha_{i,v}, i, v, \mathsf{lab}_i; \mathsf{r}_{i,v})$.
  3. Set $x := (\mathsf{pk}^\alpha, (\mathsf{ct}_{i,v})_{i,v})$ and $w := (\mathsf{lab}_i, \mathsf{r}_{i,0}, \mathsf{r}_{i,1})_i$, and compute $\pi \leftarrow \mathsf{P}(\mathsf{pk}^\alpha_{\mathsf{dv}}, x, w)$.
  4. Return $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}(\mathsf{pk}^\alpha_{\mathsf{cca}}, (\widetilde{\mathsf{Q}}, (\mathsf{ct}_{i,v})_{i,v}, \pi))$ to $\mathcal{A}$ and add (α, CT) to the list L_{kdm}.
- The decryption oracle responds to $\mathcal{A}$'s query $(\alpha \in [n], \mathsf{CT})$ as follows:
  1. Compute $(\widetilde{\mathsf{Q}}, (\mathsf{ct}_{i,v})_{i,v}, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{pk}^\alpha_{\mathsf{cca}}, \mathsf{sk}^\alpha_{\mathsf{cca}}, \mathsf{CT})$.
  2. If $\mathsf{V}(\mathsf{sk}^\alpha_{\mathsf{dv}}, (\mathsf{pk}^\alpha, (\mathsf{ct}_{i,v})_{i,v}), \pi) = \mathsf{reject}$, then return $\perp$ to $\mathcal{A}$.
  3. For every $i \in [\ell_{\mathsf{sk}}]$, compute $\mathsf{lab}_i \leftarrow \mathsf{CDec}(\mathsf{pk}^\alpha, \mathsf{sk}^\alpha, i, \mathsf{ct}_{i,\mathsf{sk}^\alpha[i]})$.
  4. Return $\mathsf{m} \leftarrow \mathsf{Eval}(\widetilde{\mathsf{Q}}, (\mathsf{lab}_i)_i)$ to $\mathcal{A}$.

  Note that the above procedure is not exactly the same as $\mathsf{Dec}_{\mathsf{kdm}}(\mathsf{PK}^\alpha, \mathsf{SK}^\alpha = \mathsf{sk}^\alpha, \mathsf{CT})$, because the computations of $\mathsf{CSDec}(\mathsf{sk}^\alpha, \widetilde{\mathsf{ct}}^\alpha)$ for retrieving $(\mathsf{sk}^\alpha_{\mathsf{cca}}, \mathsf{sk}^\alpha_{\mathsf{dv}})$ is omitted. However, the answer to a decryption query computed by the above procedure is exactly the same as that computed by $\mathsf{Dec}_{\mathsf{kdm}}$. Therefore, it does not affect $\mathcal{A}$'s view. Looking ahead, the trapdoors $\{\mathsf{td}^s\}_{s\in[n]}$ will be used from Game 5.

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

Game 2: Same as Game 1, except that the simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ for the zero-knowledge property of DVNIZK is used for generating $(\mathsf{pk}^s_{\mathsf{dv}}, \mathsf{sk}^s_{\mathsf{dv}})$ for every $s \in [n]$ and a proof π in a ciphertext in response to KDM-encryption queries, instead of using DVKG and P.

More precisely, we make the following two changes from the previous game:

(1) When generating PK^s and SK^s, $(\mathsf{pk}^s_{\mathsf{dv}}, \mathsf{sk}^s_{\mathsf{dv}}, \mathsf{td}^s_{\mathsf{dv}}) \leftarrow \mathcal{S}_1(1^\lambda)$ is executed instead of $(\mathsf{pk}^s_{\mathsf{dv}}, \mathsf{sk}^s_{\mathsf{dv}}) \leftarrow \mathsf{DVKG}(1^\lambda)$.

(2) When responding to $\mathcal{A}$'s KDM-encryption query (α, f_0, f_1), the KDM-encryption oracle computes $\pi \leftarrow \mathcal{S}_2(\mathsf{td}^\alpha_{\mathsf{dv}}, x)$ instead of $\pi \leftarrow \mathsf{P}(\mathsf{pk}^\alpha_{\mathsf{dv}}, x, w)$, where $x = (\mathsf{pk}^\alpha, (\mathsf{ct}_{i,v})_{i,v})$ and $w = (\mathsf{lab}_i, \mathsf{r}_{i,0}, \mathsf{r}_{i,1})_i$.

By applying the zero-knowledge of DVNIZK with a hybrid argument over key indices $[n]$, we have $|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]| = \mathsf{negl}(\lambda)$.

Game 3: Same as Game 2, except that for answering to KDM-encryption queries from $\mathcal{A}$, the KDM-encryption oracle uses a garbled circuit $\widetilde{\mathsf{Q}}$ and the labels $(\mathsf{lab}_i)_{i\in[\ell_{\mathsf{sk}}]}$ generated by garbling a circuit into which encryptions of the "key cycle" is hardwired, instead of using the simulator Sim of GC.

More specifically, we make the following two changes from the previous game:

Circuit $\mathsf{Q}[\alpha, f, (\widetilde{\mathsf{e}}^s)_{s\in[n]}](\cdot)$:

Hardwired: An index $\alpha \in [n]$, a circuit $f : \{0,1\}^{n\cdot\ell_{\mathsf{sk}}} \to \{0,1\}^{\mu}$, and ciphertexts $(\widetilde{\mathsf{e}}^s)_{s\in[n]}$.

Input: A string $z \in \{0,1\}^{\ell_{\mathsf{sk}}}$.

```
1. Set $z^{\alpha} \leftarrow z$.
2. For $k = \alpha, \alpha+1, \ldots, \alpha+n-2$ (where set $k \leftarrow k-n$ if $k > n$), compute $z^{(k \bmod n)+1} \leftarrow \mathsf{CSDec}(z^k, \widetilde{\mathsf{e}}^k)$.
3. Return $\mathsf{m} \leftarrow f((z^s)_{s\in[n]})$.
```

Figure 9: Description of the circuit Q.

(1) Just after generating key pairs $\{(\mathsf{PK}^s, \mathsf{SK}^s)\}_{s\in[n]}$, encryptions of the "key cycle" are generated by using the additional encryption algorithm CSEnc, namely by computing $\widetilde{\mathsf{e}}^s \leftarrow \mathsf{CSEnc}(\mathsf{sk}^s, \mathsf{sk}^{(s \bmod n)+1})$ for every $s \in [n]$.

(2) When responding to a KDM-encryption query from $\mathcal{A}$, the KDM-encryption oracle generates a garbled circuit by garbling the circuit Q shown in Figure 9.

More precisely, when $\mathcal{A}$ makes a KDM-encryption query (α, f_0, f_1), the KDM-encryption oracle computes $(\widetilde{\mathsf{Q}}, (\mathsf{lab}_{i,v})_{i,v}) \leftarrow \mathsf{Garble}(1^{\lambda}, \mathsf{Q}[\alpha, f_b, (\widetilde{\mathsf{e}}^s)_{s\in[n]}])$. Moreover, for every $i \in [\ell_{\mathsf{sk}}]$ and $v \in \{0,1\}$, the oracle computes $\mathsf{ct}_{i,v} \leftarrow \mathsf{CEnc}(\mathsf{pk}^{\alpha}, i, v, \mathsf{lab}_{i,\mathsf{sk}^{\alpha}[i]})$.[12]

Note that for every $s \in [n]$, $\widetilde{\mathsf{e}}^s$ encrypts $\mathsf{sk}^{(s \bmod n)+1}$ with the key sk^s, and thus $\mathsf{CSDec}(\mathsf{sk}^s, \widetilde{\mathsf{e}}^s) = \mathsf{sk}^{(s \bmod n)+1}$ holds. Thus, we have

$$\mathsf{Q}[\alpha, f_b, (\widetilde{\mathsf{e}}^s)_{s\in[n]}](\mathsf{sk}^{\alpha}) = f_b((\mathsf{sk}^s)_{s\in[n]}).$$

Therefore, by setting pad as the size of the circuit Q which is $O(n \cdot (|\mathsf{CSDec}| + \ell_{\widetilde{\mathsf{e}}}) + \mathsf{size})$, from the security of GC with a hybrid argument over $\mathcal{A}$'s KDM-encryption queries, we have $|\Pr[\mathtt{SUC}_2] - \Pr[\mathtt{SUC}_3]| = \mathsf{negl}(\lambda)$.

Game 4**:** Same as Game 3, except that when responding to a KDM-encryption query (α, f_0, f_1), the KDM-encryption oracle computes $\mathsf{ct}_{i,1\oplus\mathsf{sk}^{\alpha}[i]} \leftarrow \mathsf{CEnc}(\mathsf{pk}^{\alpha}, i, 1 \oplus \mathsf{sk}^{\alpha}[i], \mathsf{lab}_{i,1\oplus\mathsf{sk}^{\alpha}[i]})$ for every $i \in [\ell_{\mathsf{sk}}]$.

By applying the security against the receiver of CTE with a hybrid argument over all positions $[\ell_{\mathsf{sk}}]$ and all key indices $[n]$, we have $|\Pr[\mathtt{SUC}_3] - \Pr[\mathtt{SUC}_4]| = \mathsf{negl}(\lambda)$.

Due to the change made in this game, $\mathsf{ct}_{i,v}$ is now computed as $\mathsf{ct}_{i,v} \leftarrow \mathsf{CEnc}(\mathsf{pk}^{\alpha}, i, v, \mathsf{lab}_{i,v})$ for every $(i,v) \in [\ell_{\mathsf{sk}}] \times \{0,1\}$. This means that except for the use of the encryptions of the key cycle $\{\widetilde{\mathsf{e}}^s\}_{s\in[n]}$, the behavior of the KDM-encryption oracle becomes independent of the secret keys $\{\mathsf{sk}^s\}_{s\in[n]}$. In the next two games, we will make changes that ensure that the secret keys are also not used for responding to decryption queries.

Game 5**:** Same as Game 4, except that when responding to a decryption query under a key index $\alpha \in [n]$, the decryption oracle computes the label lab_i by decrypting $\mathsf{ct}_{i,\mathsf{sk}^{\alpha}[i]}$ using the trapdoor-decryption algorithm $\widehat{\mathsf{CDec}}$, instead of the ordinary decryption algorithm CDec, for every $i \in [\ell_{\mathsf{sk}}]$.

More precisely, for a decryption query (α, CT), the decryption oracle responds as follows. (The change from the previous game is underlined.)

[12] In Game 3, the labels of the opposite positions, $\{\mathsf{lab}_{i,1\oplus\mathsf{sk}^{\alpha}[i]}\}_{i\in[\ell_{\mathsf{sk}}]}$, are not used at all. They will be used in the subsequent games.

1. Compute $(\widetilde{\mathsf{Q}}, (\mathsf{ct}_{i,v})_{i,v}, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{pk}^{\alpha}_{\mathsf{cca}}, \mathsf{sk}^{\alpha}_{\mathsf{cca}}, \mathsf{CT})$.
2. If $\mathsf{V}(\mathsf{sk}^{\alpha}_{\mathsf{dv}}, (\mathsf{pk}^{\alpha}, (\mathsf{ct}_{i,v})_{i,v}), \pi) = \mathsf{reject}$, then return $\bot$ to $\mathcal{A}$.
3. For every $i \in [\ell_{\mathsf{sk}}]$, compute $\underline{\mathsf{lab}_i \leftarrow \widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i, \mathsf{sk}^{\alpha}[i], \mathsf{ct}_{i,\mathsf{sk}^{\alpha}[i]})}$.
4. Return $\mathsf{m} \leftarrow \mathsf{Eval}(\widetilde{\mathsf{Q}}, (\mathsf{lab}_i)_i)$ to $\mathcal{A}$.

Recall that the third condition of the correctness of CTE ensures that $\mathsf{CDec}(\mathsf{pk}^{\alpha}, \mathsf{sk}^{\alpha}, i, \mathsf{ct}) = \widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i, \mathsf{sk}^{\alpha}[i], \mathsf{ct})$ holds for all $i \in [\ell_{\mathsf{sk}}]$ and all ciphertexts ct that are not necessarily in the support of CEnc. Hence, the behavior of the decryption oracle does not change between Games 4 and 5, and we have $\Pr[\mathtt{SUC}_4] = \Pr[\mathtt{SUC}_5]$.

Game 6: Same as Game 5, except that when responding to a decryption query under a key index $\alpha \in [n]$, the decryption oracle computes the label lab_i by decrypting the "0-side" ciphertext $\mathsf{ct}_{i,0}$, instead of the "$(1 \oplus \mathsf{sk}^{\alpha}[i])$-side" ciphertext $\mathsf{ct}_{i,\mathsf{sk}^{\alpha}[i]}$, for every $i \in [\ell_{\mathsf{sk}}]$.

More precisely, for a decryption query (α, CT), the decryption oracle responds as follows. (The change from the previous game is underlined.)

1. Compute $(\widetilde{\mathsf{Q}}, (\mathsf{ct}_{i,v})_{i,v}, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{pk}^{\alpha}_{\mathsf{cca}}, \mathsf{sk}^{\alpha}_{\mathsf{cca}}, \mathsf{CT})$.
2. If $\mathsf{V}(\mathsf{sk}^{\alpha}_{\mathsf{dv}}, (\mathsf{pk}^{\alpha}, (\mathsf{ct}_{i,v})_{i,v}), \pi) = \mathsf{reject}$, then return $\bot$ to $\mathcal{A}$.
3. For every $i \in [\ell_{\mathsf{sk}}]$, compute $\underline{\mathsf{lab}_i \leftarrow \widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i, 0, \mathsf{ct}_{i,0})}$.
4. Return $\mathsf{m} \leftarrow \mathsf{Eval}(\widetilde{\mathsf{Q}}, (\mathsf{lab}_i)_i)$ to $\mathcal{A}$.

By the change made in this game, the secret keys $\{\mathsf{sk}^s\}_{s \in [n]}$ are not needed for responding to decryption queries.

We define the following events in Game $t \in \{5, \dots, 9\}$.

$\mathtt{BDQ}_t$**:** In Game t, $\mathcal{A}$ makes a decryption query (α, CT) that satisfies the following two conditions, where $(\widetilde{\mathsf{Q}}, (\mathsf{ct}_{i,v})_{i,v}, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{pk}^{\alpha}_{\mathsf{cca}}, \mathsf{sk}^{\alpha}_{\mathsf{cca}}, \mathsf{CT})$:

1. $\mathsf{V}(\mathsf{sk}^{\alpha}_{\mathsf{dv}}, (\mathsf{pk}^{\alpha}, (\mathsf{ct}_{i,v})_{i,v}), \pi) = \mathsf{accept}$.
2. $\widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i^*, 0, \mathsf{ct}_{i^*,0}) \neq \widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i^*, 1, \mathsf{ct}_{i^*,1})$ holds for some $i^* \in [\ell_{\mathsf{sk}}]$.

We call such a decryption query a *bad decryption query.*

Games 5 and 6 are identical unless $\mathcal{A}$ makes a bad decryption query in the corresponding games. Thus, we have $|\Pr[\mathtt{SUC}_5] - \Pr[\mathtt{SUC}_6]| \leq \Pr[\mathtt{BDQ}_6]$.

Game 7: Same as Game 6, except that for every $s \in [n]$, when generating PK^s, $\widetilde{\mathsf{ct}}^s$ is generated as $\widetilde{\mathsf{ct}}^s \leftarrow \mathsf{CSEnc}(\widetilde{\mathsf{ct}}^s, 0^{\mu'})$, instead of $\widetilde{\mathsf{ct}}^s \leftarrow \mathsf{CSEnc}(\mathsf{sk}^s, (\mathsf{sk}^s_{\mathsf{cca}}, \mathsf{sk}^s_{\mathsf{dv}}))$, where $\mu' = |\mathsf{sk}^s_{\mathsf{cca}}| + |\mathsf{sk}^s_{\mathsf{dv}}|$.

Recall that in Games 6 and 7, the secret keys $\{\mathsf{sk}^s\}_{s \in [n]}$ are used only to generate (1) the ciphertexts $\{\widetilde{\mathsf{ct}}^s\}_{s \in [n]}$ contained in the public keys $\{\mathsf{PK}^s\}_{s \in [n]}$, and (2) the encryptions of the key cycle $\{\widetilde{\mathsf{e}}^s\}_{s \in [n]} = \{\mathsf{CSEnc}(\mathsf{sk}^s, \mathsf{sk}^{(s \bmod n)+1})\}_{s \in [n]}$. Recall also that the special weak CIRC$^{(n)}$ security of CTE ensures the confidentiality of messages encrypted by CSEnc even in the presence of the trapdoors $\{\mathsf{td}^s\}_{s \in [n]}$ and encryptions of the key cycle. A reduction algorithm (attacking the special weak CIRC$^{(n)}$ security of CTE) can use the encryptions of the key cycle $\{\widetilde{\mathsf{e}}^s\}_{s \in [n]}$ for responding to KDM-encryption queries, and trapdoors $\{\mathsf{td}^s\}_{s \in [n]}$ for responding to decryption queries as well as detecting whether a bad query has been submitted. Hence, by the special weak CIRC$^{(n)}$ security of CTE, we have $|\Pr[\mathtt{SUC}_6] - \Pr[\mathtt{SUC}_7]| = \mathsf{negl}(\lambda)$ and $|\Pr[\mathtt{BDQ}_6] - \Pr[\mathtt{BDQ}_7]| = \mathsf{negl}(\lambda)$.

Game 8: Same as Game 7, except that when responding to a KDM-encryption query (α, f_0, f_1), the KDM-encryption oracle computes $\mathsf{CT} \leftarrow \mathsf{Enc}_{\mathsf{cca}}(\mathsf{pk}^{\alpha}_{\mathsf{cca}}, 0^{\mu''})$, where $\mu'' = |\widetilde{\mathsf{Q}}| + 2\ell_{\mathsf{sk}} \cdot |\mathsf{ct}_{i,v}| + |\pi|$.

Recall that in the previous game, we have eliminated the information of $\mathsf{sk}^s_{\mathsf{cca}}$ from $\widetilde{\mathsf{ct}}^s$ for every $s \in [n]$. Thus, we can rely on the IND-CCA security of $\mathsf{PKE}_{\mathsf{cca}}$ at this point, and straightforwardly derive $|\Pr[\mathtt{SUC}_7] - \Pr[\mathtt{SUC}_8]| = \mathsf{negl}(\lambda)$ by applying a key-index-wise hybrid argument. Moreover, a reduction algorithm (attacking the IND-CCA security of $\mathsf{PKE}_{\mathsf{cca}}$) can detect whether $\mathcal{A}$'s decryption query (α, CT) is bad by using td^{α}, $\mathsf{sk}^{\alpha}_{\mathsf{dv}}$, and the reduction algorithm's own decryption queries. Thus, the IND-CCA security of $\mathsf{PKE}_{\mathsf{cca}}$ also implies $|\Pr[\mathtt{BDQ}_7] - \Pr[\mathtt{BDQ}_8]| = \mathsf{negl}(\lambda)$.

We see that in Game 8, the challenge bit b is information-theoretically hidden from $\mathcal{A}$'s view. Thus, we have $\Pr[\mathtt{SUC}_8] = 1/2$.

We need one more game to bound $\Pr[\mathtt{BDQ}_8]$.

Game 9: Same as Game 8, except that for every $s \in [n]$, when generating PK^s, the experiment uses DVKG to generate $(\mathsf{pk}^s_{\mathsf{dv}}, \mathsf{sk}^s_{\mathsf{dv}})$, instead of using $\mathcal{S}_1$. Namely, we undo the change made between Games 1 and 2 for generating $(\mathsf{pk}^s_{\mathsf{dv}}, \mathsf{sk}^s_{\mathsf{dv}})$ for every $s \in [s]$.

By the zero-knowledge of DVNIZK, we have $|\Pr[\mathtt{BDQ}_8] - \Pr[\mathtt{BDQ}_9]| = \mathsf{negl}(\lambda)$.

Finally, we argue that the soundness of DVNIZK implies $\Pr[\mathtt{BDQ}_9] = \mathsf{negl}(\lambda)$. To see this, note that in Game 9, $(\mathsf{pk}^s_{\mathsf{dv}}, \mathsf{sk}^s_{\mathsf{dv}})$ is now generated by DVKG for every $s \in [n]$. Also, suppose $\mathcal{A}$ submits a bad decryption query (α, CT) such that

- (1) $\mathsf{V}(\mathsf{sk}^{\alpha}_{\mathsf{dv}}, (\mathsf{pk}^{\alpha}, (\mathsf{ct}_{i,v})_{i,v}), \pi) = \mathsf{accept}$, and
- (2) $\widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i^*, 0, \mathsf{ct}_{i^*,0}) \neq \widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i^*, 1, \mathsf{ct}_{i^*,1})$ for some $i^* \in [\ell_{\mathsf{sk}}]$,

where $(\widetilde{\mathsf{Q}}, (\mathsf{ct}_{i,v})_{i,v}, \pi) \leftarrow \mathsf{Dec}_{\mathsf{cca}}(\mathsf{pk}^{\alpha}_{\mathsf{cca}}, \mathsf{sk}^{\alpha}_{\mathsf{cca}}, \mathsf{CT})$. Then, notice that the condition (2) implies $(\mathsf{pk}^{\alpha}, (\mathsf{ct}_{i,v})_{i,v}) \notin L$. This can be seen by considering the contrapositive: $(\mathsf{pk}^{\alpha}, (\mathsf{ct}_{i,v})_{i,v}) \in L$ implies that for all $i \in [\ell_{\mathsf{sk}}]$, $\mathsf{ct}_{i,0}$ and $\mathsf{ct}_{i,1}$ are in the support of $\mathsf{CEnc}(\mathsf{pk}^{\alpha}, i, 0, \mathsf{lab})$ and $\mathsf{CEnc}(\mathsf{pk}^{\alpha}, i, 1, \mathsf{lab})$ for some message lab, respectively, and thus $\widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i, 0, \mathsf{ct}_{i,0}) = \widehat{\mathsf{CDec}}(\mathsf{td}^{\alpha}, i, 1, \mathsf{ct}_{i,1})$ holds due to the second condition of the correctness of CTE. Therefore, $x = (\mathsf{pk}^{\alpha}, (\mathsf{ct}_{i,v})_{i,v})$ and π computed from a bad decryption query satisfy the condition of violating the soundness of DVNIZK.

Thus, we can consider the following reduction algorithm $\mathcal{B}$ attacking the soundness of DVNIZK: On input a public proving key $\mathsf{pk}'_{\mathsf{dv}}$, $\mathcal{B}$ first randomly guesses an index α^* for which it expects $\mathcal{A}$ to make a bad decryption query, sets $\mathsf{pk}^{\alpha^*}_{\mathsf{dv}} \leftarrow \mathsf{pk}'_{\mathsf{dv}}$, generates all the key materials in Game 9 except for $(\mathsf{pk}^{\alpha^*}_{\mathsf{dv}}, \mathsf{sk}^{\alpha^*}_{\mathsf{dv}})$, and simulates Game 9 for $\mathcal{A}$. Whenever $\mathcal{A}$ makes a bad decryption query under the position α^*, $\mathcal{B}$ outputs a statement/proof pair violating the soundness of DVNIZK, while if $\mathcal{A}$ does not make a bad decryption query under the position α^* and terminates, $\mathcal{B}$ simply gives up and aborts. Note that in Game 9, to simulate the KDM-encryption oracle, $\mathcal{B}$ just returns an encryption of the all-zero string and thus need not compute a proof π of DVNIZK. Note also that $\mathcal{B}$ is not directly given the secret verification key $\mathsf{sk}^{\alpha^*}_{\mathsf{dv}}$ but is allowed to use the verification oracle, which is sufficient to perfectly simulate the decryption oracle in Game 9. Moreover, $\mathcal{B}$ can detect whether $\mathcal{A}$ has made a bad decryption query under the position α^* by using $\mathsf{sk}^{\alpha^*}_{\mathsf{cca}}$ and td^{α^*} generated by itself, and its own verification oracle. Thus, $\mathcal{B}$'s advantage in breaking the soundness of DVNIZK is at least $1/n$ times the probability that $\mathcal{A}$ makes a bad decryption query in Game 9. Hence, by the soundness of DVNIZK, we have $\Pr[\mathtt{BDQ}_9] = \mathsf{negl}(\lambda)$.

From the above arguments, we have

$$\begin{aligned}
&\frac{1}{2}\cdot \mathsf{Adv}^{\mathsf{kdmcca}}_{\mathsf{PKE}_{\mathsf{kdm}},\mathcal{B}_{\mathsf{size}},\mathcal{A},n}(\lambda) = \left|\Pr[\mathtt{SUC}_1] - \frac{1}{2}\right| \\
&\le \sum_{t\in[7]}\left|\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]\right| + \left|\Pr[\mathtt{SUC}_8] - \frac{1}{2}\right| \\
&\le \sum_{t\in[7]\setminus\{5\}}\left|\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]\right| + \Pr[\mathtt{BDQ}_6] \\
&\le \sum_{t\in[7]\setminus\{5\}}\left|\Pr[\mathtt{SUC}_t] - \Pr[\mathtt{SUC}_{t+1}]\right| + \sum_{t\in\{6,7,8\}}\left|\Pr[\mathtt{BDQ}_t] - \Pr[\mathtt{BDQ}_{t+1}]\right| + \Pr[\mathtt{BDQ}_9] \\
&= \mathsf{negl}(\lambda).
\end{aligned}$$

Since the choice of $\mathcal{A}$ and n was arbitrary, we can conclude that $\mathsf{PKE}_{\mathsf{kdm}}$ is $\mathcal{B}_{\mathsf{size}}$-KDM$^{(n)}$-CCA secure. □ (**Theorem 14**)

Acknowledgement A part of this work was supported by JST CREST Grant Number JP-MJCR19F6 and JSPS KAKENHI Grant Number 19H01109.

References

[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 / June 2010.

[ABHS05] Pedro Adão, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *ESORICS 2005*, volume 3679 of *LNCS*, pages 374–396. Springer, Heidelberg, September 2005.

[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.

[BBS03] Mihir Bellare, Alexandra Boldyreva, and Jessica Staddon. Randomness re-use in multi-recipient encryption schemeas. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 85–99. Springer, Heidelberg, January 2003.

[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 / June 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.

[BKS19] Elette Boyle, Lisa Kohl, and Peter Scholl. Homomorphic secret sharing from lattices without FHE. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 3–33. Springer, Heidelberg, May 2019.

[BPS07] Michael Backes, Birgit Pfitzmann, and Andre Scedrov. Key-dependent message security under active attacks - brsim/uc-soundness of symbolic encryption with key cycles. In *20th IEEE Computer Security Foundations Symposium, CSF 2007, 6-8 July 2007, Venice, Italy*, pages 112–124. IEEE Computer Society, 2007.

[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.

[CCH$^+$19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1082–1090. ACM Press, June 2019.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 91–122. Springer, Heidelberg, April / May 2018.

[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.

[CL02] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, Heidelberg, August 2002.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer, Heidelberg, May 2004.

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

[GI12] Oded Goldreich and Rani Izsak. Monotone circuits: One-way functions versus pseudorandom generators. *Theory of Computing*, 8(1):231–238, 2012.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 528–557. Springer, Heidelberg, April / May 2017.

[GMOR15] Siyao Guo, Tal Malkin, Igor Carboni Oliveira, and Alon Rosen. The power of negations in cryptography. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 36–65. Springer, Heidelberg, March 2015.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HK15] Mohammad Hajiabadi and Bruce M. Kapron. Reproducible circularly-secure bit encryption: Applications and realizations. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 224–243. Springer, Heidelberg, August 2015.

[HK17] Mohammad Hajiabadi and Bruce M. Kapron. Toward fine-grained blackbox separations between semantic and circular-security notions. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 561–591. Springer, Heidelberg, April / May 2017.

[KM19] Fuyuki Kitagawa and Takahiro Matsuda. CPA-to-CCA transformation for KDM security. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 118–148. Springer, Heidelberg, December 2019.

[KMT19] Fuyuki Kitagawa, Takahiro Matsuda, and Keisuke Tanaka. CCA security and trapdoor functions via key-dependent-message security. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 33–64. Springer, Heidelberg, August 2019.

[KRW15] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 378–400. Springer, Heidelberg, March 2015.

[LQR$^+$19] Alex Lombardi, Willy Quach, Ron D. Rothblum, Daniel Wichs, and David J. Wu. New constructions of reusable designated-verifier NIZKs. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 670–700. Springer, Heidelberg, August 2019.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 18–35. Springer, Heidelberg, August 2009.

[Rot13] Ron Rothblum. On the circular security of bit-encryption. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 579–598. Springer, Heidelberg, March 2013.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 1–20. Springer, Heidelberg, February 2004.

[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 Other Definitions

A.1 IND-CCA/CPA Security

Here, we recall the definitions of IND-CCA/CPA security for PKE and SKE schemes. For convenience, we consider the multi-challenge version by default. We start with the definitions

```
$\mathsf{Expt}^{\mathsf{indcca}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ :
    $L_{\mathsf{enc}} \leftarrow \emptyset$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$
    $b \xleftarrow{\mathsf{r}} \{0, 1\}$
    $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{enc}}(\cdot,\cdot), \mathcal{O}_{\mathsf{dec}}(\cdot)}(\mathsf{pk})$
    Return $(b' \overset{?}{=} b)$.

$\mathcal{O}_{\mathsf{enc}}(\mathsf{m}_0, \mathsf{m}_1)$ :    // $|\mathsf{m}_0| = |\mathsf{m}_1|$
    $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$
    $L_{\mathsf{enc}} \leftarrow L_{\mathsf{enc}} \cup \{\mathsf{ct}\}$
    Return $\mathsf{ct}$.

$\mathcal{O}_{\mathsf{dec}}(\mathsf{ct})$ :
    If $\mathsf{ct} \in L_{\mathsf{enc}}$ then return $\perp$.
    Return $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$.
```

Figure 10: The IND-CCA experiment for PKE.

for PKE.

For a PKE scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary $\mathcal{A}$, consider the experiment $\mathsf{Expt}^{\mathsf{indcca}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ described in Figure 10.

Definition 7 (IND-CCA/CPA Security) *We say that a PKE scheme* PKE *is* IND-CCA secure *if for all PPT adversaries $\mathcal{A}$, we have* $\mathsf{Adv}^{\mathsf{indcca}}_{\mathsf{PKE},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[\mathsf{Expt}^{\mathsf{indcca}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1] - 1/2| = \mathsf{negl}(\lambda)$.

IND-CPA security is defined analogously, except that $\mathcal{A}$ is disallowed to submit a decryption query. We denote the IND-CPA advantage of $\mathcal{A}$ and IND-CPA security experiment by $\mathsf{Adv}^{\mathsf{indcpa}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ *and* $\mathsf{Expt}^{\mathsf{indcpa}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$*, respectively.*

IND-CCA/CPA security for SKE is defined similarly to that for PKE, with the differences that (1) $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$, $\mathsf{Enc}(\mathsf{pk}, \cdot)$, and $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, \cdot)$ in the experiments for PKE are replaced with $\mathsf{sk} \leftarrow \mathsf{K}(1^\lambda)$, $\mathsf{E}(\mathsf{sk}, \cdot)$, and $\mathsf{D}(\mathsf{sk}, \cdot)$, respectively, in the experiments for SKE, and (2) pk given as an input to an adversary $\mathcal{A}$ is just replaced with 1^λ.

A.2 Designated-Verifier Non-interactive Zero-Knowledge Arguments

Here, we review the definitions for (reusable) designated-verifier non-interactive zero-knowledge (DV-NIZK) argument systems. We adopt the syntax used in [KM19].

Let L be an NP language associated with the corresponding NP relation R. A DV-NIZK argument system DVNIZK for L consists of the three PPT algorithms $(\mathsf{DVKG}, \mathsf{P}, \mathsf{V})$:

- DVKG is the key generation algorithm that takes 1^λ as input, and outputs a public proving key pk and a secret verification key sk.
- P is the proving algorithm that takes a public proving key pk, a statement x, and a witness w as input, and outputs a proof π.
- V is the (deterministic) verification algorithm that takes a secret verification key sk, a statement x, and a proof π as input, and outputs either accept or reject.

A DV-NIZK argument system $\mathsf{DVNIZK} = (\mathsf{DVKG}, \mathsf{P}, \mathsf{V})$ is correct if for all $\lambda \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{DVKG}(1^\lambda)$, and $(x, w) \in R$, we have $\mathsf{V}(\mathsf{sk}, x, \mathsf{P}(\mathsf{pk}, x, w)) = \mathsf{accept}$.

We require that a DV-NIZK argument system satisfy *(adaptive) soundness* and *(adaptive) zero-knowledge*. As in [KM19, LQR$^+$19], we consider the *reusable* setting, where the security experiment for soundness (resp. zero-knowledge) allows an adversary to make multiple verification (resp. proving) queries. A DV-NIZK argument system satisfying these versions of soundness and zero-knowledge is called *reusable*.

```
$\mathsf{Expt}^{\mathsf{zk-real}}_{\mathsf{DVNIZK},\mathcal{A}}(\lambda)$ :
  $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{DVKG}(1^\lambda)$
  $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{prove}}(\cdot,\cdot)}(\mathsf{pk},\mathsf{sk})$
  Return $b'$.
- - - - - - - - - - - - - - - - - - -
$\mathcal{O}_{\mathsf{prove}}(x,w)$
  If $(x,w) \notin R$ then return $\bot$.
  Return $\pi \leftarrow \mathsf{P}(\mathsf{pk},x,w)$.
```

```
$\mathsf{Expt}^{\mathsf{zk-sim}}_{\mathsf{DVNIZK},\mathcal{S},\mathcal{A}}(\lambda)$ :
  $(\mathsf{pk},\mathsf{sk},\mathsf{td}) \leftarrow \mathcal{S}_1(1^\lambda)$
  $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{prove}}(\cdot,\cdot)}(\mathsf{pk},\mathsf{sk})$
  Return $b'$.
- - - - - - - - - - - - - - - - - - -
$\mathcal{O}_{\mathsf{prove}}(x,w)$
  If $(x,w) \notin R$ then return $\bot$.
  Return $\pi \leftarrow \mathcal{S}_2(\mathsf{td},x)$.
```

Figure 11: The experiments for defining zero-knowledge of a DV-NIZK argument system: The real experiment (left) and the simulated experiment (right).

Soundness. The soundness of a DV-NIZK argument system is defined as follows.

Definition 8 (Soundness) *We say that a DV-NIZK argument system* $\mathsf{DVNIZK} = (\mathsf{DVKG}, \mathsf{P}, \mathsf{V})$ *for a language L satisfies* soundness *if for all PPT adversaries $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathsf{sound}}_{\mathsf{DVNIZK},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{DVKG}(1^\lambda);\\ (x',\pi') \leftarrow \mathcal{A}^{\mathsf{V}(\mathsf{sk},\cdot,\cdot)}(\mathsf{pk})\end{array} : \begin{array}{l}\mathsf{V}(\mathsf{sk},x',\pi') = \mathsf{accept}\\ \wedge\ x' \notin L\end{array}\right] = \mathsf{negl}(\lambda).$$

Zero-Knowledge. As usual, the zero-knowledge property of a DV-NIZK argument system $\mathsf{DVNIZK} = (\mathsf{DVKG}, \mathsf{P}, \mathsf{V})$ is defined by using a *simulator* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ whose syntax is as follows:

- $\mathcal{S}_1$ takes 1^λ as input, and outputs a fake public key pk, a fake secret key sk, and a trapdoor td.
- $\mathcal{S}_2$ takes a trapdoor td and a statement x as input, and outputs a fake proof π.

For DVNIZK (for an NP language L with the corresponding NP relation R), a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, and an adversary $\mathcal{A}$, consider the real and simulated experiments $\mathsf{Expt}^{\mathsf{zk-real}}_{\mathsf{DVNIZK},\mathcal{A}}(\lambda)$ and $\mathsf{Expt}^{\mathsf{zk-sim}}_{\mathsf{DVNIZK},\mathcal{S},\mathcal{A}}(\lambda)$, respectively, defined in Figure 11.

Definition 9 (Zero-Knowledge) *We say that a DV-NIZK argument system* $\mathsf{DVNIZK} = (\mathsf{DVKG}, \mathsf{P}, \mathsf{V})$ *for an NP language L satisfies* zero-knowledge *if there exists a PPT simulator $\mathcal{S}$ such that for all PPT adversaries $\mathcal{A}$, we have* $\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{DVNIZK},\mathcal{S},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Expt}^{\mathsf{zk-real}}_{\mathsf{DVNIZK},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Expt}^{\mathsf{zk-sim}}_{\mathsf{DVNIZK},\mathcal{S},\mathcal{A}}(\lambda) = 1]| = \mathsf{negl}(\lambda)$.

Result from [KM19, LQR+19]. Here we recall the result independently and concurrently achieved by Kitagawa and Matsuda [KM19] and Lombardi et al. [LQR+19], which we will use in Section 5.

Theorem 15 *If there exists an IND-CPA secure PKE scheme and a $\mathcal{P}$-KDM$^{(1)}$-CPA secure SKE scheme, then there exists a reusable DV-NIZK argument system for all NP languages.*

A.3 Garbling

Here, we recall the definitions of a garbling scheme in the form we use in this paper.

Let $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ be a family of circuits, where the input length of each member in $\mathcal{C}_n$ is n. A garbling scheme GC for $\mathcal{C}$ consists of the three PPT algorithms $(\mathsf{Garble}, \mathsf{Eval}, \mathsf{Sim})$.

- Garble is the garbling algorithm that takes 1^λ and (the description of) a circuit $\mathsf{C} \in \mathcal{C}_n$, where $n = n(\lambda)$ is a polynomial. Then, it outputs a garbled circuit $\widetilde{\mathsf{C}}$ and $2n$ labels $(\mathsf{lab}_{i,v})_{i\in[n],v\in\{0,1\}}$.

```
$\mathsf{Expt}^{\mathsf{gc-real}}_{\mathsf{GC},\mathcal{A}}(\lambda)$ :
    $(\mathsf{C}, \mathsf{x} \in \{0,1\}^n, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$
    $(\widetilde{\mathsf{C}}, (\mathsf{lab}_{i,v})_{i\in[n],v\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{C})$
    $b' \leftarrow \mathcal{A}_2(\widetilde{\mathsf{C}}, (\mathsf{lab}_{i,\mathsf{x}[i]})_{i\in[n]}, \mathsf{st})$
    Return $b'$.
```

```
$\mathsf{Expt}^{\mathsf{gc-sim}}_{\mathsf{GC},\mathcal{A}}(\lambda)$ :
    $(\mathsf{C}, \mathsf{x} \in \{0,1\}^n, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$
    $(\widetilde{\mathsf{C}}, (\mathsf{lab}_{i})_{i\in[n]}) \leftarrow \mathsf{Sim}(1^\lambda, |\mathsf{C}|, \mathsf{C}(\mathsf{x}))$
    $b' \leftarrow \mathcal{A}_2(\widetilde{\mathsf{C}}, (\mathsf{lab}_{i})_{i\in[n]}, \mathsf{st})$
    Return $b'$.
```

Figure 12: The security experiments for a garbling scheme: The real experiment (left) and the simulated experiment (right).

- Eval is the (deterministic) evaluation algorithm that takes a garbled circuit $\widetilde{\mathsf{C}}$ and n labels $(\mathsf{lab}_i)_{i\in[n]}$ as input, and outputs an evaluation result y.
- Sim is the simulator algorithm that takes 1^λ, the size parameter size (where $\mathsf{size} = \mathsf{size}(\lambda)$ is a polynomial), and a string y as input, and outputs a simulated garbled circuit $\widetilde{\mathsf{C}}$ and n simulated labels $(\mathsf{lab}_i)_{i\in[n]}$.

For a garbling scheme, we require the following correctness and security properties.

Correctness For all $\lambda, n \in \mathbb{N}$, $\mathsf{x} \in \{0,1\}^n$, and $\mathsf{C} \in \mathcal{C}_n$, we require that the following two conditions hold.[13]

- $\mathsf{Eval}(\widetilde{\mathsf{C}}, (\mathsf{lab}_{i,\mathsf{x}[i]})_{i\in[n]}) = \mathsf{C}(\mathsf{x})$ for all $(\widetilde{\mathsf{C}}, (\mathsf{lab}_{i,v})_{i\in[n],v\in\{0,1\}})$ output by $\mathsf{Garble}(1^\lambda, \mathsf{C})$.
- $\mathsf{Eval}(\widetilde{\mathsf{C}}, (\mathsf{lab}_{i})_{i\in[n]}) = \mathsf{C}(\mathsf{x})$ for all $(\widetilde{\mathsf{C}}, (\mathsf{lab}_{i})_{i\in[n]})$ output by $\mathsf{Sim}(1^\lambda, |\mathsf{C}|, \mathsf{C}(\mathsf{x}))$, where $|\mathsf{C}|$ denotes the size of C.

Security For all PPT adversaries $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{gc}}_{\mathsf{GC},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Expt}^{\mathsf{gc-real}}_{\mathsf{GC},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Expt}^{\mathsf{gc-sim}}_{\mathsf{GC},\mathcal{A}}(\lambda) = 1]| = \mathsf{negl}(\lambda)$, where the experiments $\mathsf{Expt}^{\mathsf{gc-real}}_{\mathsf{GC},\mathcal{A}}(\lambda)$ and $\mathsf{Expt}^{\mathsf{gc-sim}}_{\mathsf{GC},\mathcal{A}}(\lambda)$ are defined as in Figure 12.

We can realize a garbling scheme for all efficiently computable circuits based on a one-way function [Yao86].

B Constructing Weakly Noisy-Leakage-Resilient PKE (Proof of Lemma 1)

In this section, we give a proof of Lemma 1, namely, how to transform an IND-CPA secure PKE scheme with simple key generation into a weakly noisy-leakage-resilient one (with simple key generation). The transformation uses a universal hash function and its security proof uses the leftover hash lemma [HILL99, DRS04].[14] Thus, we first introduce them, and then proceed to the formal proof.

Recall that for distributions $\mathcal{X}$ and $\mathcal{Y}$ defined over the same set, the *statistical distance between* $\mathcal{X}$ *and* $\mathcal{Y}$ is defined by $\mathbf{SD}(\mathcal{X}, \mathcal{Y}) := \frac{1}{2}\sum_z |\Pr[\mathcal{X} = z] - \Pr[\mathcal{Y} = z]|$.

Definition 10 (Universal Hash Family) *A family of hash functions (hash family, for short)* $\mathcal{H} = \{H : D \to R\}$ *is said to be* universal *if for all distinct elements* $x, x' \in D$*, we have*

$$\Pr_{H \xleftarrow{\mathsf{r}} \mathcal{H}}[H(x) = H(x')] \le |R|^{-1}.$$

[13] Requiring correctness for the output of the simulator may be somewhat non-standard. However, it is satisfied by Yao's garbling scheme based on an IND-CPA secure SKE scheme.

[14] Naor and Segev [NS09] showed their result on weak bounded leakage resilience by using a randomness extractor. We use a universal hash family and the leftover hash lemma for concreteness.

$\mathsf{KG}'(\mathsf{sk}' \in \{0,1\}^{\ell'_{\mathsf{sk}}})$: $H \overset{\mathsf{r}}{\leftarrow} \mathcal{H}$ $\mathsf{sk} \leftarrow H(\mathsf{sk}')$ $\mathsf{pk} \leftarrow \mathsf{KG}(\mathsf{sk})$ Return $\mathsf{pk}' \leftarrow (\mathsf{pk}, H)$.	$\mathsf{Enc}'(\mathsf{pk}', \mathsf{m})$: $(\mathsf{pk}, H) \leftarrow \mathsf{pk}'$ $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ Return ct.	$\mathsf{Dec}'(\mathsf{pk}', \mathsf{sk}', \mathsf{ct})$: $(\mathsf{pk}, H) \leftarrow \mathsf{pk}'$ $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{pk}, H(\mathsf{sk}'), \mathsf{ct})$. Return m.

Figure 13: The transformation of an IND-CPA secure PKE scheme with simple key generation into a weakly noisy-leakage-resilient one.

Lemma 4 (Leftover Hash Lemma, Adapted from [DRS04]) *Let $\mathcal{H} = \{H : D \to R\}$ be a universal hash family. Let $(\mathcal{X}, \mathcal{Y})$ be a joint distribution such that the support of $\mathcal{X}$ is contained in D. Define the following two distributions $\mathcal{D}^{\mathtt{real}}_{(\mathcal{X},\mathcal{Y})}$ and $\mathcal{D}^{\mathtt{rand}}_{(\mathcal{X},\mathcal{Y})}$:*

$$\mathcal{D}^{\mathtt{real}}_{(\mathcal{X},\mathcal{Y})} := \left\{ H \overset{\mathsf{r}}{\leftarrow} \mathcal{H};\ (x,y) \leftarrow (\mathcal{X},\mathcal{Y}) : (H, H(x), y) \right\},$$

$$\mathcal{D}^{\mathtt{rand}}_{(\mathcal{X},\mathcal{Y})} := \left\{ H \overset{\mathsf{r}}{\leftarrow} \mathcal{H};\ (x,y) \leftarrow (\mathcal{X},\mathcal{Y});\ r \overset{\mathsf{r}}{\leftarrow} R : (H, r, y) \right\}.$$

Then, it holds that

$$\mathbf{SD}(\mathcal{D}^{\mathtt{real}}_{(\mathcal{X},\mathcal{Y})}, \mathcal{D}^{\mathtt{rand}}_{(\mathcal{X},\mathcal{Y})}) \leq \frac{1}{2}\sqrt{|R| \cdot 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})}}.$$

Proof of Lemma 1. We first show a transformation of a PKE scheme PKE with simple key generation to another scheme PKE' (which also has simple key generation) using a universal hash family H, and then show that if PKE is IND-CPA secure, then PKE' is weakly L-noisy-leakage-resilient. It will be evident that the construction and reduction are black-box.

The transformation uses the following building blocks/parameters:

- Let $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure PKE scheme with simple key generation whose secret key length is $\ell_{\mathsf{sk}} = \ell_{\mathsf{sk}}(\lambda)$.
- Let $L = L(\lambda)$ and $\ell'_{\mathsf{sk}} = \ell'_{\mathsf{sk}}(\lambda)$ be any polynomials satisfying $\ell = \ell(\lambda) := \ell'_{\mathsf{sk}} - (L + \ell_{\mathsf{sk}}) = \omega(\log \lambda)$.
- Let $\mathcal{H} = \{H : \{0,1\}^{\ell'_{\mathsf{sk}}} \to \{0,1\}^{\ell_{\mathsf{sk}}}\}$ be a universal hash family.

Using PKE and $\mathcal{H}$ as building blocks, the transformed PKE scheme with simple key generation $\mathsf{PKE}' = (\mathsf{KG}', \mathsf{Enc}', \mathsf{Dec}')$, whose secret key length is ℓ'_{sk}, is constructed as in Figure 13.

We remark that if we adopt the syntax of a PKE scheme in which there is a setup algorithm that generates a public parameter shared by all users and algorithms, then the universal hash function H can be put into the public parameter, and thus need not be generated for each public key.

We now give the proof of security. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be any PPT L-noisy-leakage-respecting adversary that attacks the weak L-noisy-leakage-resilience of PKE'. We will show that there exists a PPT adversary $\mathcal{B}$ that attacks the IND-CPA security of the underlying PKE scheme PKE so that

$$\mathsf{Adv}^{\mathsf{wlr}}_{\mathsf{PKE}',\mathcal{A},L}(\lambda) \leq \mathsf{Adv}^{\mathsf{indcpa}}_{\mathsf{PKE},\mathcal{B}}(\lambda) + 2^{-\ell/2}. \tag{3}$$

Since $\ell = \omega(\log \lambda)$ and thus $2^{-\ell/2} = \mathsf{negl}(\lambda)$, this inequality implies that if PKE is IND-CPA secure, then PKE' is weakly L-noisy-leakage-resilient.

To this end, consider the following two games:

Game 1: This is the weak L-noisy-leakage-resilience experiment $\mathsf{Expt}^{\mathsf{wlr}}_{\mathsf{PKE}',\mathcal{A},L}(\lambda)$ itself.

Game 2: Same as Game 1, except that at the key generation, the key pair $(\mathsf{pk},\mathsf{sk})$ of the underlying PKE scheme PKE is generated by picking $\mathsf{sk} \in \{0,1\}^{\ell_{\mathsf{sk}}}$ independently of sk' and then computing $\mathsf{pk} \leftarrow \mathsf{KG}(\mathsf{sk})$, instead of using $\mathsf{sk} \leftarrow H(\mathsf{sk}')$.

Note that in this game, the leakage is still computed as $f(\mathsf{sk}')$. Because of the change made in this game, sk becomes independent of sk'.

For $t \in \{1,2\}$, let $\mathtt{SUC}_t$ be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game t. By the definition of the games and the triangle inequality, we have

$$\mathsf{Adv}^{\mathsf{wlr}}_{\mathsf{PKE}',\mathcal{A},L}(\lambda) = 2 \cdot \left|\Pr[\mathtt{SUC}_1] - \frac{1}{2}\right| \le 2 \cdot \left|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]\right| + 2 \cdot \left|\Pr[\mathtt{SUC}_2] - \frac{1}{2}\right|.$$

We thus bound the two terms appearing in the above inequality.

Firstly, we argue $2 \cdot |\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]| \le 2^{-\ell/2}$. For $t \in \{1,2\}$, let $\mathcal{D}_t$ be the distribution of the values $(H, \mathsf{sk}, f(\mathsf{sk}'), \mathsf{st})$ in Game t, where f is the leakage function and st is the state information both output by $\mathcal{A}_0(1^\lambda)$. Note that the only difference between Game 1 and Game 2 is the distribution of this tuple. (More specifically, only in the generation of sk: $\mathsf{sk} = H(\mathsf{sk}')$ in Game 1 and $\mathsf{sk} \xleftarrow{\mathsf{r}} \{0,1\}^{\ell_{\mathsf{sk}}}$ in Game 2.) Hence, $|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]|$ is upper-bounded by $\mathbf{SD}(\mathcal{D}_1, \mathcal{D}_2)$. Now, consider the following distribution $\mathcal{D}'$:

$$\mathcal{D}' := \left\{ (f,\mathsf{st}) \leftarrow \mathcal{A}_0(1^\lambda);\ \mathsf{sk}' \xleftarrow{\mathsf{r}} \{0,1\}^{\ell'_{\mathsf{sk}}} : (\mathsf{sk}', f(\mathsf{sk}'), \mathsf{st}) \right\}.$$

We interpret $\mathcal{D}'$ as the joint distribution $(\mathcal{X}, \mathcal{Y})$ where $\mathcal{X}$ corresponds to sk' and $\mathcal{Y}$ corresponds to $(f(\mathsf{sk}'), \mathsf{st})$. By definition, we have $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y}) = \widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f(\mathsf{sk}'), \mathsf{st})$. Since $\mathcal{A}$ is L-noisy-leakage-respecting, we have $L \ge \mathbf{H}_\infty(\mathsf{sk}') - \widetilde{\mathbf{H}}_\infty(\mathsf{sk}'|f(\mathsf{sk}'), \mathsf{st}) = \ell'_{\mathsf{sk}} - \widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})$, or equivalently $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y}) \ge \ell'_{\mathsf{sk}} - L$. Also, note that $\mathcal{D}_1$ (resp. $\mathcal{D}_2$) can be seen as the distribution $\mathcal{D}^{\mathtt{real}}_{(\mathcal{X},\mathcal{Y})}$ (resp. $\mathcal{D}^{\mathtt{rand}}_{(\mathcal{X},\mathcal{Y})}$) defined in the leftover hash lemma (Lemma 4). Hence, by applying the leftover hash lemma, we have

$$\begin{aligned} 2 \cdot \left|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]\right| &\le 2 \cdot \mathbf{SD}(\mathcal{D}_1, \mathcal{D}_2) = 2 \cdot \mathbf{SD}(\mathcal{D}^{\mathtt{real}}_{\mathcal{X},\mathcal{Y}}, \mathcal{D}^{\mathtt{rand}}_{(\mathcal{X},\mathcal{Y})}) \\ &\le \sqrt{2^{\ell_{\mathsf{sk}}} \cdot 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})}} \le \sqrt{2^{-(\ell'_{\mathsf{sk}} - (\ell_{\mathsf{sk}} + L))}} = 2^{-\ell/2}, \end{aligned}$$

as desired.

Secondly, we show that there exists a PPT adversary $\mathcal{B}$ that attacks the IND-CPA security of the underlying PKE scheme PKE so that $\mathsf{Adv}^{\mathsf{indcpa}}_{\mathsf{PKE},\mathcal{B}}(\lambda) = 2 \cdot |\Pr[\mathtt{SUC}_2] - 1/2|$. To see this, consider the following adversary $\mathcal{B}$:

$\mathcal{B}^{\mathcal{O}_{\mathsf{Enc}}(\cdot,\cdot)}(\mathsf{pk})$**:** $\mathcal{B}$ firstly runs $(f, \mathsf{st}) \leftarrow \mathcal{A}_0(1^\lambda)$. Next, $\mathcal{B}$ picks $H \xleftarrow{\mathsf{r}} \mathcal{H}$ and $\mathsf{sk}' \xleftarrow{\mathsf{r}} \{0,1\}^{\ell'_{\mathsf{sk}}}$, sets $\mathsf{pk}' \leftarrow (\mathsf{pk}, H)$, and runs $\mathcal{A}_1(\mathsf{pk}', f(\mathsf{sk}'), \mathsf{st})$.

For each of $\mathcal{A}_1$'s encryption queries $(\mathsf{m}_0, \mathsf{m}_1)$, $\mathcal{B}$ just forwards the pair $(\mathsf{m}_0, \mathsf{m}_1)$ to its own encryption oracle $\mathcal{O}_{\mathsf{Enc}}(\cdot,\cdot)$, and returns the received result to $\mathcal{A}_1$.

When $\mathcal{A}_1$ terminates with output b', $\mathcal{B}$ outputs b' and terminates.

It is straightforward to see that $\mathcal{B}$ perfectly simulates Game 2 for $\mathcal{A}$ so that $\mathcal{B}$'s challenge bit is that of $\mathcal{A}$'s. Hence, we have $\mathsf{Adv}^{\mathsf{indcpa}}_{\mathsf{PKE},\mathcal{B}}(\lambda) = 2 \cdot |\Pr[\mathtt{SUC}_2] - 1/2|$.

Putting everything together, we have shown that for any PPT L-noisy-leakage-respecting adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ satisfying Equation 3. This means that PKE' is weakly L-noisy-leakage-resilient. □ (**Lemma 1**)