KDM Security for Identity-Based Encryption: Constructions and Separations

Yu Chen * Jiang Zhang † Yi Deng ‡ Jinyong Chang §

Abstract

For encryption schemes, key dependent message (KDM) security requires that ciphertexts preserve secrecy even when the encrypt messages may depend on the secret keys. While KDM security has been extensively studied for public-key encryption (PKE), it receives much less attention in the setting of identity-based encryption (IBE). In this work, we focus on the KDM security for IBE. Our results are threefold.

- We give a simple structure-mirroring PKE-to-IBE transformation via indistinguishability obfuscation and puncturable PRFs. This transformation establishes a connection between PKE and IBE in general. Particularly, it provides a liberal interface for transferring the KDM security results (both positive and negative) from PKE to IBE, though in the selective-identity sense.
- On the positive side, we present two constructions that achieve KDM security in the adaptive-identity sense for the first time. One is generically built from identity-based hash proof system (IB-HPS) with homomorphic property, which indicates that the IBE schemes of Gentry (Eurocrypt 2006), Coron (DCC 2009), Chow et al. (CCS 2010) are actually KDM-secure in the single-key setting. The other is built from indistinguishability obfuscation and a new notion named puncturable unique signature, which is bounded KDM-secure in the single-key setting.
- On the negative side, we separate n-circular security (which is a prototypical case of KDM security) from the standard IND-CPA/CCA security for IBE by giving a counterexample based on differing-inputs obfuscation and a new notion named puncturable IBE.

Mathematics Subject Classification: 94A60

Keywords: KDM security, IBE, IB-HPS, Obfuscation

1 Introduction

Secure encryption is arguably the most central subject in cryptography. Starting with semantic (or IND-CPA) security [GM84], secure encryption has developed a series of successively stronger security notions providing secrecy in increasingly adversarial scenarios. Nevertheless, standard security notions (including semantic security and its successive stronger notions) have to assume that the encrypted messages do not directly depend on the secret key, since as observed by the seminal work of Goldwasser and Micali [GM84] semantic security may compromise if the

*State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. Email: yuchen.prc@gmail.com

†State Key Laboratory of Cryptology, Beijing, China. Email: jiangzhang09@gmail.com

‡State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. Email: ydeng.cas@gmail.com

§Peking University, Beijing, China. Email: changjinyong@pku.edu.cn

adversary gets to see encryptions of the secret key. As a result, for a long time encrypting key-dependent messages was considered as a dangerous abuse of an encryption scheme. However, recent research has revealed great importance of secure key-dependent encryption. On the practical side, it admits natural implementation of encrypted storage systems (e.g. BitLocker in Windows operating systems). On the theoretical side, it has surprising connections with other fundamental notions such as obfuscation and encryption with weakly random keys. It also plays a crucial role for designing some high-level cryptographic protocols, such as discouraging delegation of credentials in anonymous credential system [CL01], enabling "bootstrapping" technique in fully homomorphic encryption [Gen09], and realizing symbolic protocols with the framework of axiomatic security [ABHS05].

1.1 Related Work

The formal study of key-dependent message (KDM) security dates back to more than a decade ago. Camenisch and Lysyanskaya [CL01] considered n-circular security, which stipulates semantic security remains in the presence of an encrypted "key circle", where n secret keys is organized in a cycle and each secret key is encrypted under the public key of its left neighbor. Black et al. [BRS02] suggested generalized KDM security, which stipulates semantic security still holds even when the adversary can ask for encryptions of key-dependent messages $m \leftarrow f(sk_1, \ldots, sk_n)$ under pk_i, where $(pk_i, sk_i)_{1 \leq i \leq n}$ are n public/secret key pairs and f is an arbitrary function from permissible dependent function family $\mathcal{F}$. However, the circular-secure and KDM-secure PKE schemes proposed in [CL01, BRS02] are only provably secure in the random oracle model. Since then, a challenging problem is to achieve KDM security without relying on random oracle heuristic.

Several years later, Boneh et al. [BHHO08] made a major progress by giving an elegant KDM-secure PKE scheme w.r.t. affine functions in the standard model under the decisional Diffie-Hellman assumption. After this breakthrough, a large number of results have emerged, both on the positive and negative side.

On the positive side, there are three main lines of research. The first direction focuses on broadening $\mathcal{F}$ from a weak family of functions to a larger one via generic amplification techniques, including [BHHI10, BGK11, App11]. The second direction aims to achieve better efficiency by adopting block-wise encryption, including [ACPS09, MTY11]. The third direction considers KDM security under more powerful attacks, including chosen-ciphertext attack [BDU08, CCS09, Hof13, QLH13, LLJ15, HLL16], key leakage attack [BG10, HKS16], related-key attack [BDH14]. Recently, a new trend is seeking constructions of KDM-secure PKE from general assumptions or systems. Wee [Wee16] presents a framework of KDM-secure PKE schemes via hash proof system with homomorphic property. This elegant framework yields a conceptually simple and unified treatment of the works of Boneh et al. [BHHO08], Brakerski and Goldwasser [BG10] and Brakerski, Goldwasser and Kalai [BGK11] in the single-key setting. Marcedone et al. [MPS16] proposes an ingenious PKE scheme with bounded KDM security from one-way functions and indistinguishability obfuscation ($i\mathcal{O}$).

On the negative side, there are two lines of research. The first direction focuses on the complexity of KDM security. Haitner and Holenstein [HH09] showed that there is no black-box construction of KDM-secure PKE w.r.t. all (unbounded size) circuits. Barak et al. [BHHI10] extended this impossibility result by showing that it is impossible to prove KDM security w.r.t. $\mathcal{F}$ that contains exponentially hard pseudorandom functions, using only black-box access to the query function and the adversary. The second direction considers the separation of n-circular security (a restricted form of KDM security) from standard security notions like IND-CPA/IND-CCA security. For the case of $n = 1$ there are trivial counterexamples via folklore argument, but when $n \geq 2$ the question turns out to be much more challenging. For the case of $n = 2$, Acar et al. [ABBC10] and Cash et al. [CGH12] respectively gave the counterexamples that are IND-CPA secure but not 2-circular secure, based on the SXDH assumption over asymmetric bilinear groups. Later, Bishop et al. [BHW15] obtained more counterexamples for $n = 2$, based on the decision linear assumption and learning with errors (LWE) assumptions. For the more general case of arbitrary n, Koppula et al. [KRW15] derived a counterexample based on the assumption that $i\mathcal{O}$ for arbitrary polynomial sized circuits exists. Concurrently and independently, Marcedone and Orlandi [MO14] gave a similar result under on a stronger assumption of virtual black-box (VBB) obfuscation exists for a certain functionality.[1] Very recently, Koppula and Waters [KW16] and Alamati and Peikert [AP16] contrived the counterexamples based on the plain LWE and ring-LWE assumptions, respectively.

1.2 Motivation

Most existing works on KDM security dealt with the symmetric or public-key settings. Compare to PKE, IBE is generally more difficult to construct due to its richer functionality. To date, there are only two works addressed KDM security in the identity-based setting. Alperin-Sheriff and Peikert [AP12] initiated the study of KDM security for IBE. They considered user-level KDM security, which captures the scenario that the encrypted messages might be functions of users secret keys. They also proposed a KDM-secure IBE scheme w.r.t. affine functions under the LWE assumption. Galindo et al. [GHV12] considered system-level KDM security for IBE, which captures the scenario that the encrypted messages might be functions of the master secret key. They constructed such an IBE scheme w.r.t. affine functions under the rank assumption in bilinear groups, but only provides security against a bounded number of encryption queries.

Prior constructions of [AP12, GHV12] have a common downside: they are only provably secure in the selective-identity sense (the adversary must declare the target identities before seeing the master public key), which is a weaken model in IBE. While KDM security in the selective-identity sense might be sufficient in some restricted situations, KDM security in the adaptive-identity sense is more natural and covers many real world attacks. Thereby, constructing KDM-secure IBE schemes in the adaptive-identity sense is an important open problem (as noted in [GHV12]).

On the other hand, to justify the dedicate pursue of KDM security for IBE, a fundamental problem is whether the standard security notions like IND-CPA/CCA already imply KDM security in the identity-based setting. Relative to few positive results, no such negative results are known.

In summary, as opposite to the extensive study in the PKE setting, the research of KDM security for IBE is largely open, both on the positive and negative side. We are thus motivated to consider the following intriguing questions:

Can we transfer the KDM security results for PKE to IBE in a general manner? How to construct KDM-secure IBE schemes in the adaptive-identity sense? Do standard security notions like IND-CPA/CCA security imply KDM security in the realm of IBE?

2 Our Contributions

Our focus. The fact that there are two types of secret keys in IBE gives rise to two levels of KDM security in the identity-based setting, depending on whether the adversary gets to see encryptions of functions of the master secret keys or users secret keys. In this work, following the choice of [AP12] we choose to focus on user-level KDM security. The first reason is that

[1] Later, the authors of [MO14] refined their counterexample to rely only on $i\mathcal{O}$ following [KRW15].

it is a mirror image of KDM security for PKE in the IBE setting, and thus allows us to carry out comparative study. The second reason is that it enables some important applications such as "bootstrapping" technique for identity-based fully homomorphic encryption [GSW13]. Last but not the least, it is crucial for the management of an IBE system itself (e.g. key revocation, updating and retrieve), as elaborated in [AP12].

Our contributions of this work is threefold. We first give a simple PKE-to-IBE transformation based on $i\mathcal{O}$ and puncturable PRFs. A distinguished property of this transformation is structure-mirroring, which provides us a generic complier to transfer the KDM security results (including both positive constructions and negative counterexamples) from PKE to IBE. The downside is that the obtained results are restricted to the selective-identity sense. We then focus on the constructions of KDM-secure IBE in the adaptive-identity sense. Firstly, we propose a generic construction from identity-based hash proof system with homomorphic property, which is compact and KDM-secure w.r.t. $\mathcal{F}$ defined by the corresponding projective hash. Secondly, we give a concrete construction from $i\mathcal{O}$ and a new notion named puncturable unique signature, which is KDM-secure w.r.t. circuits of a priori bounded size. To the best of our knowledge, they are the first two IBE schemes achieving KDM security in the adaptive-identity sense. On the negative side, we show that in the IBE setting the standard IND-CPA/CCA security does not imply n-circular security by contriving a counterexample based on differing-inputs obfuscation and a new notion called puncturable IBE. In what follows, we give an overview of our results.

2.1 Transfer KDM Security Results for PKE to IBE

Given fruitful results on KDM security in the public-key setting, a promising idea for constructing KDM-secure IBE is to make a KDM-secure PKE identity-based. To do so, we need an efficient id-to-pk hash to map identities to well-formed public keys as well as a master trapdoor to enable the Private Key Generator (PKG) to extract secret keys for any identities. In addition, the master trapdoor should be "puncturable" to admit a reduction to the starting PKE.

We observe that the construction by Alperin-Sheriff and Peikert [AP12] is a good exemplification of this idea. Roughly, they first constructed a KDM-secure PKE from lattices. This PKE is of "dual"-style and thus admits an efficient id-to-pk hash. They then transformed it to an IBE by embedding a puncturable master trapdoor via a so called "all-but-d" trapdoor functions. Nevertheless, it seems difficult to generalize their construction since its id-to-pk hash and all-but-d trapdoor functions all heavily rely on the specific algebra of lattices.

The above analysis indicates that the primary technical hurdles to implement the promising idea in a general manner lie in the id-to-pk hash is not always obvious especially when the well-formed public keys are exponentially sparse (as noted in [GPV08], e.g. the Regev's PKE), and the existence of a puncturable master trapdoor is unclear.

Structure-mirroring PKE-to-IBE transformation. We circumvent these hurdles by giving a generic PKE-to-IBE transformation. Let PKE = $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be any PKE, the transformation proceeds as follows: choose a puncturable PRF whose domain is the desired identity space and pick a random PRF key as the master secret key msk; build the master public key mpk as an obfuscation of a circuit that first computes the PRF value of the input identity, then uses the PRF value as the random coins to invoke PKE.KeyGen to obtain a key pair, and finally discards the secret key and only outputs the public key; to extract a secret key for an identity id, the PKG first computes its PRF value at point id using msk, then invokes PKE.KeyGen to recover the corresponding key pair and outputs the secret key; to encrypt a message under an identity id, the sender first derives the corresponding public key by executing

mpk on id (note that mpk is essentially an obfuscated circuit), then runs PKE.$\mathsf{Encrypt}$; the decryption algorithm is same as that of the underlying PKE. We highlight that the heart of this transformation is using a puncturable PRF key as msk and invoking PKE.KeyGen with random coins $\mathsf{PPRF}(msk, id)$ to obtain the corresponding key pair for id. This key mechanism provides us a universal id-to-pk hash as well as an all purpose puncturable master trapdoor, which allow us to derive an IBE from any PKE.

A salient feature of the above transformation is structure-mirroring, which means the structures of secret keys and ciphertexts of the resulting IBE are identical to those of the starting PKE. This feature enables us to translate the KDM security results (including positive constructions and negative counterexamples) for PKE to IBE in a neat and generic way: If the starting PKE is KDM-secure (w.r.t. $\mathcal{F}$ under CPA/CCA attacks), the resulting IBE is also KDM-secure in the same setting. If the starting PKE is n-circular insecure, so is the resulting IBE.

Somewhat surprisingly, the generality of this transformation admits much broader applications beyond KDM security: it immediately lifts a bunch of security results (e.g., CCA security, leakage/tampering resilience) from PKE to IBE.

The shortcoming of this transformation is that it only yields security in the selective-identity sense. Such security loss seems unavoidable due to the use of the punctured programming techniques [SW14]: the reduction has to program the target identities to the target public keys when publishing mpk. We leave the generic PKE-to-IBE transformation that yields security in the adaptive-identity sense as a challenging open problem.

2.2 KDM-secure IBE from Identity-Based Hash Proof System

Recently, Wee [Wee16] presented an elegant framework for constructing KDM-secure PKE from hash proof system (HPS) [CS02] with homomorphic property. Inspired by this result, a tempting idea is to construct KDM-secure IBE from identity-based hash proof system (IB-HPS) introduced by Alwen et al. [ADN+10], which is a counterpart of HPS in the identity-based setting.

Next, we briefly review the generalized notion of IB-HPS, then sketch how to build KDM-secure IBE from IB-HPS satisfying homomorphic property.

Generalized IB-HPS. Let $L \subset X$ be a collection of languages indexed by the identity set I. An IB-HPS for $L \subset X$ consists of four polynomial-time algorithms: (Setup, $\mathsf{Extract}$, Priv, Pub). The Setup algorithm outputs a master key pair (mpk, msk); the $\mathsf{Extract}$ algorithm outputs a secret key sk_{id} for id using msk; the Priv algorithm defines a hash $\Lambda : SK \times X \to \Pi$ where SK is the secret key space and Π is the proof space; the Pub algorithm admits public evaluation of Λ on L. We say $\Lambda_{sk_{id}}$ is smooth if its output distributes uniformly over Π when $x \xleftarrow{\text{R}} X \backslash L_{id}$, and say it is projective if its output is completely determined by id for $x \in L_{id}$. We also require the language membership problem is hard in the identity-based setting: for arbitrarily chosen $id \in I$, the two distributions of $x \xleftarrow{\text{R}} L_{id}$ and $x \xleftarrow{\text{R}} X \backslash L_{id}$ are computationally indistinguishable, even the adversary knows a secret key for any identities (including id).

KDM security from IB-HPS. Starting with a smooth IB-HPS, we can build a CPA secure IBE scheme as below. Let Π be a group under operation "+" and the message space $M = \Pi$.[2] The Setup and $\mathsf{Extract}$ algorithms are exactly the same as that of IB-HPS. To encrypt a message m under an identity id, the sender randomly picks $x \leftarrow L_{id}$ with witness w, computes $\pi \leftarrow \Lambda_{sk_{id}}(x)$ publicly via Pub with w, and sets $c = (x, y = \pi + m)$ as the ciphertext. To decrypt

[2] Generally, one can always assume there exists an efficient invertible encoding $\phi : M \to \Pi$.

a ciphertext $c = (x, y)$, the receiver computes $\pi \leftarrow \Lambda_{sk_{id}}(x)$ privately via Priv with sk_{id}, then outputs $m = (y - \pi)$. The correctness of this construction follows from the projective property of Λ, while the CPA security follows from a simple hybrid. Let id^* be the target identity and m_0, m_1 be the target messages chosen by the adversary and $(x^*, y^* = \Lambda_{sk_{id^*}}(x^*) + m_\beta)$ be the challenge ciphertext generated by the challenger. We first switch the distribution of x^* from $U_{L_{id^*}}$ to $U_{X \setminus L_{id^*}}$ (this change is computationally indistinguishable due to the hardness of the language membership problem), then apply the smoothness of $\Lambda_{sk_{id^*}}$ to hide the message m_β in an information-theoretical way.

Akin to the public-key setting, a primary difficulty of attaining KDM security for IBE is that the simulator has to answer KDM encryption queries whereas typically it does not know the secret keys. Observe that a notable property of the above security reduction (from IBE to IB-HPS) is that the simulator always possesses secret keys for any identity. It seems that with such property one can easily bypass the difficulty: achieving KDM security w.r.t. any computable functions, a.k.a, no PPT adversary can tell a real KDM encryption oracle apart from a zero encryption oracle given only black-box access. However, the intuition is problematic in that the responses to real KDM encryption oracle, i.e., encryptions of key dependent messages $f(sk_{id^*})$ may leak the information of sk_{id^*} in an uncontrollable way, and thus we are unable to directly apply smooth property of $\Lambda_{sk_{id^*}}$ to argue the CPA security anymore.

We tackle this problem following the high-level idea outlined in [MTY11]: find a way to simulate KDM encryptions without knowledges of the secret keys, yet ensure that the simulated KDM encryptions are indistinguishable to the real KDM encryptions as well as the zero encryptions. Naturally, to make such simulation possible, we have to restrict $\mathcal{F}$ to a function family tied to the structure of Λ. Concretely, we implement this idea by extending the techniques due to [Wee16]. Assume Λ additionally satisfies homomorphism, i.e., $\Lambda_{sk_{id}}(x_1 \cdot x_2) = \Lambda_{sk_{id}}(x_1) + \Lambda_{sk_{id}}(x_2)$ for any $id \in I$ and any $x_1, x_2 \in X$, we show that the above IBE construction from IB-HPS is actually KDM-secure w.r.t. $\mathcal{F} = \{f_{u,v} : sk \rightarrow \Lambda_{sk}(u) + v\}_{u \in X, v \in \Pi}$.

The KDM security is established in two steps, as depicted in Figure 5 and 6. Let id^* be the target identity chosen by the adversary. We first exploit the homomorphism of $\Lambda_{sk_{id^*}}$ (coupled with the projective property) and the group structure of X (coupled with the language membership problem) to show that real KDM encryptions are indistinguishable from simulated encryptions without using sk_{id^*}. Then we are safe to apply smoothness of $\Lambda_{sk_{id^*}}$ (coupled with the projective property and the group structures of X and Π) to show simulated encryptions are indistinguishable from zero encryptions. The formal proof is done via a sequence of hybrids that changes query by query. This proves the KDM security in the single-key setting.

Note that the homomorphic requirement on Λ is met by most known instantiations of IB-HPS,[3] therefore this result immediately indicates that the IBE schemes in [Gen06, Cor09, CDRW10] are actually KDM-secure in the single-key setting. When $\mathcal{F}$ induced by Λ consists of affine function family (possibly in the exponent as in [BG10]), we can amplify $\mathcal{F}$ to the class of circuits of a-priori bounded size [BHHI10, App11].

Achieving leakage-resilience simultaneously. Note that IB-HPS is also a powerful tool in constructing leakage-resilient IBE schemes in the bounded-retrieval model. The leakage-resilient IBE from IB-HPS [ADN+10] is almost identical to the IND-CPA construction except that a randomness extractor is applied to the hash proof before using it. More precisely, let $\mathsf{ext} : \Pi \times S \rightarrow K$ be an average-case strong randomness extractor, and the message space $M = K$. To encrypt m under id, the sender randomly picks $x \xleftarrow{\text{R}} L_{id}$ with witness w and a random seed $s \in S$, computes $\pi \leftarrow \Lambda_{sk_{id}}(x)$ publicly via Pub and w, then sets $c = (x, s, y = \mathsf{ext}_s(\pi) + m)$ as ciphertext. It is not hard to see that if ext_s is homomorphic on Π, then $\mathsf{ext}_s \circ \Lambda_{sk_{id}}$ is

[3] If the projective hash in IB-HPS instantiations due to Boneh et al. [BGH07] and Gentry et al. [GPV08] satisfy homomorphic property is still unclear to us.

homomorphic on X. Thereby, this construction simultaneously achieves KDM security w.r.t. $\mathcal{F}' = \{f'_{u,v,s} : sk \rightarrow \mathsf{ext}_s(\Lambda_{sk}(u) + v))\}_{u \in X, v \in \Pi, s \in S}$. Note that as an explicit construction of average-case strong extractor [DORS08], universal hash family usually admits simple algebra structure, and thus naturally satisfies the homomorphic requirement on Π. Particularly, as noted in [Wee16], when $\Lambda_{sk_{id}}$ itself serves as a good extractor, the leakage-resilient construction is already KDM-secure without any modification. This could be viewed as a special case of our generalized explanation by setting ext_s as identity function.

Comparison with [AP12]. Their scheme is KDM-secure w.r.t. affine functions based on the LWE assumption. However, it only offers security in the selective-identity sense.

Our scheme is a generic construction based on IB-HPS, which achieves KDM-security w.r.t. affine-like functions[4] in the adaptive-identity sense. Existing instantiations of homomorphic IB-HPS imply our construction can be based on a variety of number-theoretic assumptions, including the decisional augmented bilinear Diffie-Hellman (BDH) exponent assumption, the decisional square BDH assumption, and the decisional BDH assumption. Last but not the least, by applying appropriate randomness extractor, our construction achieves leakage-resilience simultaneously.

2.3 KDM-secure IBE from Puncturable Unique Signature and $i\mathcal{O}$

KDM security grows stronger when $\mathcal{F}$ is larger. The largest possible $\mathcal{F}$ is the family of circuits of a-priori bounded size, and the corresponding security is refereed to as bounded KDM security [BHHI10].[5] Though we can attain bounded KDM security by applying amplification technique [BHHI10, App11] to our generic construction based on IB-HPS, it is still instructive to seek direct constructions.

Very recently, Marcedone et al. [MPS16] proposed an ingenious bounded KDM-secure PKE scheme from one-way functions (OWF) and $i\mathcal{O}$, which are qualitatively different[6] to the specific assumptions (LWE, DDH, QR or DCR) previously used to achieve bounded KDM security. We refer to their construction as the MPS scheme, and choose it as the starting point of our IBE construction.

Starting point: the MPS scheme. We first briefly review the MPS scheme (in the single-key case), then show how to adapt it into an IBE scheme. Let $F : X \rightarrow Y$ be a family of injective OWFs. The secret key is just a random $x \in X$, while the public key consists of $g \xleftarrow{\text{R}} F$ and $y \leftarrow g(x)$. To encrypt a message m, the ciphertext is an obfuscation of a circuit Enc that hardwires $pk = (g, y)$ and m as constants, and on input sk returns m if $g(sk) = y$ and $\perp$ otherwise. To decrypt a ciphertext, one just runs the ciphertext (which is an obfuscated circuit in nature) on input the secret key.

The proof for KDM security follows the *triple mode proof framework* [MTY11]. More precisely, it proceeds via three games. Game 0 and Game 2 correspond to real KDM encryption and zero encryption respectively, while Game 1 corresponds to the simulated KDM encryption. An important requirement is that the simulation should be done without the knowledge of secret key and the simulated KDM encryption must be indistinguishable from the real KDM encryption in Game 0 and zero encryption in Game 2.

[4] Since our construction is generic, the exact form of $\mathcal{F}$ is decided by the concrete instantiation of IB-HPS.

[5] [BHHI10] also introduces a slighter stronger notion named length-dependent security, in which the circuit size could grow polynomially in the length of their inputs and outputs. In this work, we stick to bounded KDM security for simplicity of exposition.

[6] It is known that black-box construction of collision-resistant hash functions (CRHFs) from OWF and $i\mathcal{O}$ is impossible, and as a result, they are separated from those assumptions that imply the existence of CRHFs.

The authors of [MPS16] achieve this by obfuscating a circuit Sim that hardwires $pk = (g, y)$ and a function f as constants, and on input sk outputs $f(sk)$ if $y = g(sk)$ and $\perp$ otherwise. Since $\mathsf{Enc}_{pk,m=f(sk)}$ and $\mathsf{Sim}_{pk,f}$ are functionally equivalent, one can reduce the indistinguishability between Game 0 and Game 1 to the security of $i\mathcal{O}$. Proving Game 1 $\approx_c$ Game 2 is more involved, because $\mathsf{Sim}_{pk,f}$ and $\mathsf{Enc}_{pk,0^{|m|}}$ may have differing inputs. Here, a stronger form of obfuscation – differing-input obfuscation ($di\mathcal{O}$) [BGI+12, ABG+13, BCP14] is required. Before applying $di\mathcal{O}$, one have to show that no PPT adversary can find a differing input of $\mathsf{Sim}_{pk,f}$ and $\mathsf{Enc}_{pk,0^{|m|}}$. Since the entire simulations of Game 1 and Game 2 do not require the secret key, thus one easily argue this based on the one-wayness of g. In addition, by requiring the underlying OWF to be injective, the above two circuits have at most one differing input. According to [BCP14], $di\mathcal{O}$ for such circuit family is implied by standard $i\mathcal{O}$.

Basic idea for adaption. A straightforward approach to make the MPS scheme identity-based is using our structure-mirroring PKE-to-IBE transformation. However, the resulting IBE only achieves bounded KDM security in the selective-identity sense. A useful observation is that, unlike most encryption schemes, the ciphertext in the MPS scheme is simply an obfuscated circuit which outputs m if its input is a valid secret key corresponding to the public key and outputs $\perp$ otherwise. Such distinguished feature makes the encryption and decryption insensitive to the concrete algebra structures of the secret key and public key. This gives us more flexibility for adaption, and possibly admits dedicated approach rather than the general-purpose PKE-to-IBE transformation.

The crux of the adaption is to introduce a master trapdoor for the MPS scheme. A tempting idea is to replace injective OWFs with injective adaptive trapdoor functions (ATDFs) [KMO10]. More precisely, the master public key is an ATDF g, while the master secret key is its trapdoor td. The identity space is the range Y of g, and a secret key for $id \in Y$ is simply its preimage under g, which is efficiently computable with td. Unfortunately, ATDF does not suffice for the adaption. This is because the security of IBE implies that no PPT adversary is able to find a secret key for any *adversarially chosen* identity even given access to a secret key extraction oracle, while with ATDF it only guarantees that no PPT adversary is able to find a preimage for a *uniformly chosen* image (corresponds to identity) even given access to an inversion oracle. Intuitively, we need a stronger version of ATDFs whose adaptive one-wayness holds for any adversarially chosen image.

We observe that unique signature [GO92, Lys02] can be somewhat viewed as such a "strong" injective ATDF. This leads to the following bounded KDM-secure IBE adapted from the MPS scheme: the PKG generates a key pair for unique signature, output the verification key as mpk and the singing key as msk; a secret key for an identity id is its unique signature signing by msk; to encrypt a message m under an identity id, one outputs an obfuscation of a circuit Enc that hardwires mpk, id and m as constants, and on input sk returns m if sk is valid signature of id and $\perp$ otherwise; to decrypt a ciphertext, one just runs the ciphertext on input the secret key.

Superficially, the security proof can be easily adapted from that for the MPS scheme. In more details, it also proceeds via three games. In Game 0 and Game 2 the simulator answers the encryption queries with real KDM encryption and zero encryption respectively, while in Game 1 the simulator answers the encryption queries with an obfuscation of circuit Sim, which hardwires mpk, id and a function f as constants, and on input sk outputs $f(sk)$ if sk is a valid signature of id and $\perp$ otherwise.

Puncturable unique signature. The devil is in the details. Akin to the proof for the MPS scheme, we have to rely on the security of $di\mathcal{O}$ to prove Game 1 $\approx_c$ Game 2, in that the two circuits $\mathsf{Sim}_{mpk,id,f}$ and $\mathsf{Enc}_{mpk,id,0^{|m|}}$ may have differing inputs. The tricky part is in our context the auxiliary information aux (typically derived from the random coins used to sample the challenging circuits) plays a crucial role when applying $di\mathcal{O}$, which is different from the situation in the MPS scheme. On one hand, aux might not contain the entire random coins used for sampling the two differing-input circuits, since otherwise an adversary may easily find the differing-input. On the other hand, in some applications aux must contain proper secret random coins to admit a reduction from a distinguishing adversary to an algorithm against the security of $di\mathcal{O}$.

We illustrate this subtlety in details via our basic construction. Let id^* be the target identity. If $aux = msk$, then a PPT adversary can easily find a differing-input of $\mathsf{Sim}_{mpk,id^*,f}$ and $\mathsf{Enc}_{mpk,id^*,0^{|m|}}$ by computing $sk_{id^*} \leftarrow \mathsf{Sign}(msk, id^*)$ with msk. From one extreme to the other, if aux contains nothing, there is no way to reduce the indistinguishability of Game 1 and Game 2 to the security of $di\mathcal{O}$, because the simulator is unable to handle the extraction queries made by the distinguishing adversary. We remark that the same issue does not occur in the MPS scheme, because in their setting the adversary does not make queries related to the secret key and thus the simulation for Game 1 and Game 2 could be done without the secret key (in other words, aux could be empty).

We tackle this dilemma by introducing a new notion called puncturable unique signature (PUS). Roughly speaking, a PUS is a unique signature scheme with an additional algorithm $\mathsf{Puncture}$ that on input a signing key sk and a message m^* outputs a succinct punctured signing key $sk(\{m^*\})$, where $sk(\{m^*\})$ can be used to sign any messages other than m^*. Moreover, the signature scheme is still unforgeable on m^* even given this punctured key.

By exploring PUS instead of normal unique signature, we are able to split the secret coins (a.k.a. msk) surgically, i.e., setting $aux = msk(\{id^*\})$. On one hand, given $msk(\{id^*\})$ no PPT adversary can find a differing input of $\mathsf{Sim}_{mpk,id^*,f}$ and $\mathsf{Enc}_{mpk,id^*,0^{|m|}}$ based on the unforgeability of PUS. On the other hand, the indistinguishability of Game 1 and Game 2 can be reduced to the security of $di\mathcal{O}$ because with $msk(\{id^*\})$ the reduction is able handle all legal extraction queries correctly.

By the unique property of PUS, the two circuits have at most one differing input. According to [BCP14], $di\mathcal{O}$ for such circuits are implied by $i\mathcal{O}$. Besides, we note that PUS is implied by injective OWF and $i\mathcal{O}$. This allows us to achieve the bounded KDM security of our IBE scheme based on solely OWF and $i\mathcal{O}$.

2.4 Counterexample of n-Circular Security

One fundamental question is whether KDM security is implied by standard security notions such as IND-CPA (or IND-CCA) in the identity-based setting. If this were true, we would get it for free without considering such notion specifically.

A cursory examination of the problem reveals that the answer is no. As we will sketch in Section 7, one can derive a simple counterexample for 1-circular security. However, akin to the situation in the public-key setting, contriving counterexamples for $n \geq 2$ based on well-studied assumptions becomes significantly more challenging. The primary difficulty somewhat resembles to that identified in [BHW15]: when n identities are thrown into a mix, we need a magic mechanism to enable the identities and ciphertexts to communicate with each other in a way that admits cycle detection but does not compromise semantic security. In public-key setting, prior counterexamples [ABBC10, CGH12, BHW15, KW16, AP16] based on pairing or lattice realize this magic mechanism by introducing extra structures (tie to the algebra of the underlying assumptions) over public keys and ciphertexts.

One may be tempted to extend this line of works to the IBE setting. Unfortunately, two technical hurdles rule out this possibility. Firstly, in IBE identities are self contained and thus it seems impossible to expose extra structures on them.[7] Secondly, in IBE the target identities are adaptively chosen by the adversary. This stands in sharp contrast to the PKE setting where the target public keys are chosen by the challenger, and thus intuitively requires the magic mechanism could be executed "on the fly".

We then turn our attention to $i\mathcal{O}$, which had demonstrated its power in deriving counterexamples in public-key setting.

Review of counterexamples from $i\mathcal{O}$ in the PKE setting. Koppula et al. [KRW15] and Marcedone and Orlandi [MO14] gave two counterexamples for arbitrary n using $i\mathcal{O}$. In a nutshell, their idea is to publish an obfuscation of a circuit called $\mathsf{CycleTest}$ along with each normal IND-CPA encryption, which hardwires the message m as the secret key, takes as inputs public keys $(pk_1, \ldots, pk_n)$ and ciphertexts $(c_1, \ldots, c_n)$, and detects if they form an encryption circle of length n. To prove the modified encryption is still IND-CPA secure, the crux is to argue the circuit $\mathsf{CycleTest}$ does not compromise the CPA security. For this purpose, another circuit $\mathsf{CycleReject}$ which always outputs $\perp$ is introduced. Clearly, $\mathsf{CycleReject}$ does not leak any information, and thus the desired IND-CPA security follows provided that $i\mathcal{O}(\mathsf{CycleTest})$ and $i\mathcal{O}(\mathsf{CycleReject})$ are computationally indistinguishable. In combination with $i\mathcal{O}$, their key idea is to introduce *valid/invalid public keys* such that the two types public keys are computationally indistinguishable on themselves, but are discernible given the associated secret keys. Accordingly, the circuit $\mathsf{CycleTest}$ will check whether its input public keys are valid and output $\perp$ if not.

The overall security is established by the following three hybrids: Hyb_1 uses valid public keys and attaches $i\mathcal{O}(\mathsf{CycleTest})$ along with each encryption; Hyb_2 switches to invalid public keys and the rest are same to Hyb_1; Hyb_3 replaces $i\mathcal{O}(\mathsf{CycleTest})$ with $i\mathcal{O}(\mathsf{CycleReject})$. Eventually, Hyb_1 and Hyb_2 are indistinguishable based on the indistinguishability of valid and invalid public keys, while Hyb_2 and Hyb_3 are indistinguishable based on the security of $i\mathcal{O}$. Thereby, the modified encryption scheme is still IND-CPA secure but n-circular insecure.

Initial attempts. As noted in [MO14, KRW15], the *valid/invalid public keys* switching mechanism lies at the heart of their counterexamples. One might be tempted to adapt their counterexamples to the identity-based setting. However, it does not work due to the fundamental difference between PKE and IBE, as we elaborate below.

The first attempt is to introduce *valid/invalid identity* in an analogous manner. But, this is impossible because identities are always self-recognizable in identity-based setting and thus there is no concept of validity for identities.

The second attempt is to introduce *valid/invalid master public keys*. To establish IND-CPA security in combination with $i\mathcal{O}$, on one hand we need to stipulate invalid master public keys are discernible given secret keys for any identity, whereas on the other hand the hybrid using valid master public key and another hybrid using invalid one must be indistinguishable. This is also impossible since an adversary against IBE can obtain secret keys for any identity other than the target one and thus can easily tell these two hybrids apart.

The above analysis indicates that we have to find a new way to work with obfuscation, without relying on valid/invalid switching technique.

Our approach. We choose an arbitrary IND-CPA secure IBE scheme which satisfies a mild property named *checkable secret key* (which we will formally define in Definition 3.5) as the starting point of our counterexample. Our basic idea is still to publish an obfuscation of a circuit $\mathsf{CycleTest}$ along with each encryption of a message m under some identity id. $\mathsf{CycleTest}$ hardwires

[7] Though arguably we can do this indirectly via our structure-mirroring PKE-to-IBE transformation, it only yields results in the selective-identity sense.

m and id as constants, takes as inputs identities $(id_1, \ldots, id_n)$ and ciphertexts $(c_1, \ldots, c_n)$, sets m as the secret key for id_2 and then attempts to decrypt circularly.

As opposed to the design of checking validity of public key in [MO14, KRW15], during decryption process $\mathsf{CycleTest}$ checks whether each intermediate result is a valid secret key for the corresponding identity as defined. Finally, it outputs "1" if all intermediate results pass the check and "$\perp$" otherwise. To show the modified encryption scheme remains IND-CPA secure, we also introduce a circuit $\mathsf{CycleReject}$ which always returns $\perp$, and wish to show the original game (using obfuscation of $\mathsf{CycleTest}$) and the final game (using obfuscation of $\mathsf{CycleReject}$) are computationally indistinguishable. However, as we analyzed before, valid/invalid switching technique does not extend to identity-based setting. As a consequence, it is unlikely to create an intermediate game in which $\mathsf{CycleTest}$ always returns $\perp$, and thus $i\mathcal{O}$ does not suffice to ensure the original game and the final game are computationally indistinguishable since $\mathsf{CycleTest}$ and $\mathsf{CycleReject}$ are not functionally equivalent.

Differing-Input obfuscation. To overcome this problem, we have to resort to $di\mathcal{O}$. In our context, a prerequisite to utilize $di\mathcal{O}$ is to show that no PPT adversary can find a differing input of $\mathsf{CycleTest}$ and $\mathsf{CycleReject}$. To this end, we further modify $\mathsf{CycleTest}$, making it output the secret key for id_1 rather than a single bit "1" when inputs indeed form an encryption circle. It is easy to see that with this design, if a PPT adversary can find a differing input, the reduction immediately obtains a secret key of id_1, and thus completely breaks the assumed security of the starting IBE scheme. Now we are able to show the obfuscations of $\mathsf{CycleTest}$ and $\mathsf{CycleReject}$ are computationally indistinguishable based on the security of $di\mathcal{O}$, and thus the desired IND-CPA security follows since $\mathsf{CycleReject}$ reveals nothing. We highlight that here we use $di\mathcal{O}$ in a novel way: prior works [ABG+13, BCP14, BST14] directly use the differing inputs to yield contradiction, while we use the output of differing-inputs.

Puncturable IBE. Similar to the status in our second positive construction, here we need to manipulate aux carefully when employing $di\mathcal{O}$. Let id^* be the target identity. If we set $aux = msk$, then a PPT adversary can easily find a differing-input of $\mathsf{CycleTest}$ and $\mathsf{CycleReject}$ by generating an encryption circle $(c_1, \ldots, c_n)$ with respect to $(id^*, id_2, \ldots, id_n)$, where $id_2, \ldots, id_n$ could be arbitrary distinct identities. If aux contains nothing, there is no way to reduce the indistinguishability of the original game using $di\mathcal{O}(\mathsf{CycleTest})$ and the final game using $di\mathcal{O}(\mathsf{CycleReject})$ to the security of $di\mathcal{O}$, because the simulator is unable to handle the extraction queries made by the distinguishing adversary.

We resolve this problem by introducing a new notion of puncturable IBE (PIBE). Roughly speaking, a PIBE is an IBE with an additional algorithm $\mathsf{Puncture}$ that on input msk and an identity id^* outputs a succinct punctured master secret key $msk(\{id^*\})$, where $msk(\{id^*\})$ can be used to extract secret keys for any identities other than id^*. We show that PIBE can be generically constructed from hierarchical IBE. By choosing a PIBE as the starting point of our counterexample, we are able to split the secret coins (a.k.a. msk) surgically, i.e., setting $aux = msk(\{id^*\})$. This allows us finally prove the IND-CPA security of our counterexample based on the security of $di\mathcal{O}$.

In addition, we extend the framework for counterexamples [BHW15] to the IBE setting, which might be of independent interest. Via this framework, we can easily augment the above counterexample to a new one that separates n-circular security from IND-CCA security.

Remark 2.1. Garg et al. [GGHW14] showed that existence of $di\mathcal{O}$ with respect to general auxiliary inputs contradicts a certain "special purpose" obfuscation conjecture. However, this conjecture is not implied by $di\mathcal{O}$. Bellare et al. [BSW16] showed that if sub-exponentially secure OWF exists, then sub-exponentially secure $di\mathcal{O}$ for TMs with unbounded inputs does not exist. Given the results [ABG+13, BCP14] that $di\mathcal{O}$ for circuits and SNARKs [BCCT12, BCC+14] imply $di\mathcal{O}$ for TMs with unbounded inputs, if SNARKs exist then their negative result extends to $di\mathcal{O}$ for circuits. However, their primary negative result only rules out sub-exponentially secure $di\mathcal{O}$ for TMs with unbounded inputs, based on sub-exponential hardness assumption. Besides, Gentry and Wichs [GW11] showed that SNARGs (and thus also SNARKs) cannot be reduced to *any falsifiable cryptographic assumptions* [Nao03] in a black-box manner. So far, the existence of polynomially-secure $di\mathcal{O}$ for polynomial sized circuits (which we used in this work) does not contradict to any standard assumption.

We are also aware of that two variants of $di\mathcal{O}$ evade the aforementioned implausible results. One is $di\mathcal{O}$ for circuits that differ on only polynomially-many inputs proposed by Boyle et al. [BCP14], which is implied by $i\mathcal{O}$. The other one is public-coin $di\mathcal{O}$ proposed by Ishai et al. [IPS15], which stipulates that only public coins can be used to sample the challenging circuits. However, we can use them in the place of $di\mathcal{O}$ in our counterexample sketched as above. Firstly, the fact that $\mathsf{CycleTest}_{id^*,m}$ and $\mathsf{CycleReject}$ having super-polynomial differing-inputs excludes the first choice. Secondly, with public-coin $di\mathcal{O}$ it is impossible to reduce the hardness of finding differing-inputs to the security of IBE, which is a secret-coin notion.

Interpreting our result. We view our result as a first step toward showing that standard security notions for IBE do not imply circular security. Although one may complain that the evidence is not strong due to the use of $di\mathcal{O}$, it do give us some elementary understanding of circular security and its challenges in the IBE setting. We left the counterexamples from well-studied assumptions as a challenging open problem.

3 Preliminaries

3.1 Basic Notations

For a set X, we use $x \xleftarrow{\text{R}} X$ to denote the operation of sampling x uniformly at random from X, and use $|X|$ to denote its size. We use U_X to denote the uniform distribution over X. For a positive integer d, we use $[d]$ to denote the set $\{1, \ldots, d\}$. We denote $\lambda \in \mathbb{N}$ as the security parameter. We say that a quantity is negligible, written $\mathsf{negl}(\lambda)$, if it vanishes faster than the inverse of any polynomial in λ. A probabilistic polynomial time (PPT) algorithm is a randomized algorithm that runs in time $\mathsf{poly}(\lambda)$. If $\mathcal{A}$ is a randomized algorithm, we write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n; r)$ to indicate that $\mathcal{A}$ outputs z on inputs $(x_1, \ldots, x_n)$ and random coins r. For notational clarity we usually omit r and write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n)$.

Let $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ denote two ensembles of random variables indexed by λ. We say that X and Y are statistically indistinguishable, written $X \approx_s Y$, if the statistical distance between X_λ and Y_λ is negligible in λ. We say that X and Y are computationally indistinguishable, written $X \approx_c Y$, if the advantage of any PPT algorithm in distinguishing X_λ and Y_λ is $\mathsf{negl}(\lambda)$.

3.2 Indistinguishability/Differing-Input Obfuscation for Circuits

We recall the notion of *indistinguishability obfuscation* for circuits from Garg et al. [GGH⁺13]. First, we define the notion of equivalent sampler.

Definition 3.1 (Equivalent Sampler for Circuits)**.** An efficient non-uniform sampling algorithm Sample is called an *equivalent sampler* for a circuit family $\mathcal{C}_\lambda$ if there exists a negligible function α such that the following holds:

$$\Pr[\forall x, C_0(x) = C_1(x) : (C_0, C_1, aux) \leftarrow \mathsf{Sample}(\lambda)] > 1 - \alpha(\lambda)$$

Definition 3.2 (Indistinguishability Obfuscator ($i\mathcal{O}$))**.** A uniform PPT machine $i\mathcal{O}$ is called an indistinguishability obfuscator for a circuit class $\{\mathcal{C}_\lambda\}$ if the following conditions are satisfied:

- (Preserving Functionality) For all security parameter $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all inputs $x \in \{0,1\}^*$, we have that:

$$\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\lambda, C)] = 1$$

- (Indistinguishability of Obfuscation) For any PPT adversaries $\mathcal{S}$, $\mathcal{D}$, if $\mathcal{S}$ constitutes an equivalent sampler w.r.t. a negligible function α, then we have:

$$|\Pr[\mathcal{D}(aux, i\mathcal{O}(\lambda, C_0)) = 1] - \Pr[\mathcal{D}(aux, i\mathcal{O}(\lambda, C_1)) = 1]| \leq \alpha(\lambda)$$

  where $(C_0, C_1, aux) \leftarrow \mathcal{S}(\lambda)$.

[GGH+13] showed how $i\mathcal{O}$ can be constructed for the circuit class $P/poly$. Next, we recall the notion of *differing-input obfuscation* from Ananth et al. [ABG+13], which is also equivalent to that of Boyel et al. [BCP14]. First, we define the notion of a differing-inputs sampler.

Definition 3.3 (Differing-Inputs Sampler for Circuits)**.** An efficient non-uniform sampling algorithm Sample is called a *differing-inputs sampler* for a circuit family $\mathcal{C}_\lambda$ if for all PPT adversary $\mathcal{A}$, we have that:

$$\Pr[C_0(x) \neq C_1(x) : (C_0, C_1, aux) \leftarrow \mathsf{Sample}(\lambda), x \leftarrow \mathcal{A}(C_0, C_1, aux)] \leq \alpha(\lambda)$$

Definition 3.4 (Differing-Inputs Obfuscator for Circuits)**.** A uniform PPT machine $di\mathcal{O}$ is called a *differing-inputs obfuscator* for a circuit family $\{C_\lambda\}$ if it satisfies the following conditions:

- (Preserving Functionality) For all security parameter $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all inputs $x \in \{0,1\}^*$, we have that:

$$\Pr[C'(x) = C(x) : C' \leftarrow di\mathcal{O}(\lambda, C)] = 1$$

- (Indistinguishability of Obfuscation) For any PPT adversaries $\mathcal{S}$, $\mathcal{D}$, if $\mathcal{S}$ constitutes a differing-inputs sampler w.r.t. a negligible function α, we have:

$$|\Pr[\mathcal{D}(aux, di\mathcal{O}(\lambda, C_0)) = 1] - \Pr[\mathcal{D}(aux, di\mathcal{O}(\lambda, C_1)) = 1]| \leq \alpha(\lambda),$$

  where $(C_0, C_1, aux) \leftarrow \mathcal{S}(\lambda)$.

Lemma 3.1 ([BCP14])**.** *For the circuit class $P/poly$, $i\mathcal{O}$ implies $di\mathcal{O}$ for circuits differing on at most polynomially-many inputs.*

3.3 Punctureable Pseudorandom Functions

Puncturable PRFs (PPRFs) [SW14] is a simplest type of constrained PRFs (CPRFs) [KPTZ13, BW13, BGI14]. In PPRFs, constrained key can be derived for any polynomial size subset T of domain X, and such a constrained key allows evaluation on all elements $x \in X \backslash T$. Formally, a puncturable PRF $\mathsf{F} : K \times X \to Y$ is given by three polynomial time algorithms as below:

- $\mathsf{KeyGen}(\lambda)$: on input a security parameter λ, output a random secret key $k \xleftarrow{\text{R}} K$.
- $\mathsf{Puncture}(k, T)$: on input a secret key $k \in K$ and a polynomial size subset $T \subset X$, output a punctured key $k(T)$.

- $\mathsf{Eval}(k(T), x)$: on input a punctured key $k(T)$ and an element $x \in X$, output $\mathsf{F}(k, x)$ if $x \notin T$ and a special reject symbol $\perp$ otherwise.

Security. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against PPRFs and define its advantage in the following experiment:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\beta = \beta' : \begin{array}{l} k \leftarrow \mathsf{KeyGen}(\lambda); \\ (state, T = \{x_1^*, \ldots, x_n^*\}) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{eval}}(\cdot)}(\lambda); \\ k(T) \leftarrow \mathsf{Puncture}(k, T); \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}, \{y_{i,0}^* \leftarrow \mathsf{F}(k, x_i^*), y_{i,1}^* \xleftarrow{\mathrm{R}} Y\}_{1 \le i \le n}; \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{eval}}(\cdot)}(state, k(T), \{y_{i,\beta}^*\}_{1 \le i \le n}); \end{array}\right] - \frac{1}{2}.$$

Here $\mathcal{O}_{\mathsf{eval}}(\cdot)$ is an evaluation oracle that on input x returns $y \leftarrow \mathsf{F}(k, x)$. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is not allowed to query $\mathcal{O}_{\mathsf{eval}}(\cdot)$ with $x_i^* \in T$. A puncturable PRF is said to be pseudorandom if for any PPT adversary, its advantage defined as above is negligible in λ. A weaker notion named selective pseudorandomness for PPRF can be defined via a similar experiment by asking $\mathcal{A}_1$ to commit (T, x^*) at the very beginning.

3.4 Identity-Based Encryption

Definition 3.5 (Identity-Based Encryption)**.** An identity-based encryption scheme [BF03] consists of four algorithms as follows.

- $\mathsf{Setup}(\lambda)$: on input a security parameter λ, output a master public key mpk and a master secret key msk.[8]
- $\mathsf{Extract}(msk, id)$: on input msk and an identity $id \in I$, output a secret key sk_{id} for id.
- $\mathsf{Encrypt}(mpk, id, m)$: on input mpk and an identity $id \in I$ and a message $m \in M$, output a ciphertext c.
- $\mathsf{Decrypt}(sk_{id}, c)$: on input a secret key sk_{id} and a ciphertext $c \in C$, output a message $m \in M$ or a special reject symbol $\perp$ indicating c is invalid.

Perfect correctness. For all $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$, all $id \xleftarrow{\mathrm{R}} I$, all $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$, all $m \xleftarrow{\mathrm{R}} M$ and all $c \leftarrow \mathsf{Encrypt}(mpk, id, m)$, it holds that $\mathsf{Decrypt}(sk_{id}, c) = m$.

Checkable secret key. We say a secret key sk is valid for id if sk is honestly generated by $\mathsf{Extract}(msk, id)$. Moreover, we say an IBE scheme satisfies "checkable secret key" property if there exists an efficient deterministic algorithm $\mathsf{CheckSK}$ that can check if a given secret key sk is valid for id. It is easy to verify that most existing pairing-based IBE schemes [BF03, BB04, Wat05, Gen06, Wat09] and lattice-based IBE schemes [GPV08, ABB10] satisfy such property.

IND-CPA Security. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against IBE and define its advantage in the following experiment:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\beta = \beta' : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\lambda); \\ (state, id^*, m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{ext}}(\cdot)}(mpk); \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(mpk, id^*, m_\beta); \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{ext}}(\cdot)}(state, c^*); \end{array}\right] - \frac{1}{2}.$$

[8] We assume mpk includes the descriptions of identity space I, message space M, and ciphertexts space C. mpk will be used as an input for algorithms $\mathsf{Extract}$ and $\mathsf{Decrypt}$, and is omitted when the context is clear.

Here $\mathcal{O}_{\mathsf{ext}}(\cdot)$ is an extraction oracle that on input $id \in I$ returns $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$. Note that $\mathcal{O}_{\mathsf{ext}}(\cdot)$ returns the same sk_{id} for repeated extraction queries on id. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is not allowed to query $\mathcal{O}_{\mathsf{ext}}(\cdot)$ with id^*. An IBE scheme is said to be IND-CPA secure if for any PPT adversary $\mathcal{A}$, its advantage defined as above is negligible in λ. The IND-CCA security for IBE can be defined similarly by giving the adversary access to an additional decryption oracle $\mathcal{O}_{\mathsf{dec}}(\cdot,\cdot)$ that on input $\langle id, c\rangle$ returns $m \leftarrow \mathsf{Decrypt}(sk_{id}, c)$. A natural constraint is that $\mathcal{A}_2$ is not allowed to query $\mathcal{O}_{\mathsf{dec}}(\cdot,\cdot)$ with (id^*, c^*).

A weaken security notion for IBE is selective-identity IND-CPA/IND-CCA security, where the adversary has to commit the target identity id^* before seeing mpk.

3.5 Key-Dependent Message Security for IBE

The following definition is adapted from [AP12]. We use slightly different but actually equivalent notation, however.

Let $\mathcal{F} \subset \{f : SK^{d\le n} \to M\}$, where SK is the secret key space and M is the message space and n is the maximum number of users. We use $|m|$ to represent the length of each message in M. We define KDM security w.r.t. $\mathcal{F}$ ($\mathcal{F}$-KDM security for short) for IBE as below.

KDM Security. Let $\mathcal{A}$ be an adversary against $\mathcal{F}$-KDM security for IBE and define its advantage as:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\beta = \beta' : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\lambda); \\ \beta \xleftarrow{\text{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ext}}(\cdot), \mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot,\cdot)}(mpk); \end{array}\right] - \frac{1}{2}.$$

Let **id** be a list of target identities which is initially empty. During the course of the game, $\mathcal{A}$ can adaptively add identities to $\mathcal{I}$ and access $\mathcal{O}_{\mathsf{ext}}(\cdot)$ and $\mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot,\cdot)$. Here $\mathcal{O}_{\mathsf{ext}}(\cdot)$ is an extraction oracle that on input an identity $id \in I$ returns a secret key $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$. Note that $\mathcal{O}_{\mathsf{ext}}(\cdot)$ returns the same sk_{id} for repeated extraction queries on id. In order to make the definition meaningful, $\mathcal{A}$ is not allowed to query $\mathcal{O}_{\mathsf{ext}}(\cdot)$ for any identities in **id**. $\mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot,\cdot)$ is an encryption oracle depending on a hidden bit β chosen by $\mathcal{CH}$, which on input $i \in [n]$ and $f : SK^d \to M \in \mathcal{F}$ (here d could be any integer less than the current number of identities in **id**), returns a key-dependent encryption $\mathsf{Encrypt}(mpk, id_i^*, f(sk_1^*, \ldots, sk_d^*))$ where sk_i^* is the secret key for id_i^* if $\beta = 0$ and returns a zero encryption $\mathsf{Encrypt}(mpk, id_i^*, 0^{|m|})$ if $\beta = 1$. An IBE scheme is said to be $\mathcal{F}$-KDM secure if for any PPT adversary $\mathcal{A}$, its advantage defined as above is negligible in λ. The selective-identity $\mathcal{F}$-KDM security for IBE can be defined similarly by requiring the adversary $\mathcal{A}$ to commit **id** before seeing mpk.

In this work, we mainly consider two KDM function families for IBE.

Bounded size circuits. Let $\mathcal{F}_{\mathsf{bound}}$ be the set of all functions $f : SK^{d\le n} \to M$ that can be encoded as circuits of size bounded by a a-priori polynomial $p(\lambda)$. Such $\mathcal{F}$ is the largest ensemble for which it is feasible to achieve KDM security, and the corresponding KDM security is referred to as bounded KDM security [BHHI10].

Affine functions. We assume for simplicity $SK \subseteq M$. If M is a ring, we can define affine class $\mathcal{F}_{\mathsf{aff}} = \{a_1 sk_1 + \cdots + a_n sk_n + c \mid a_i, c \in M\}$. The set of all constant functions $\mathcal{F}_{\mathsf{const}} = \{f_c(sk_1, \ldots, sk_n) = c\}_{c\in M}$ and the set of all selector functions $\mathcal{F}_{\mathsf{selec}} = \{f_j(sk_1, \ldots, sk_n) = sk_j\}_{j\in[n]}$ are two important subsets of $\mathcal{F}_{\mathsf{aff}}$. As observed in [BHHO08], KDM security w.r.t. $\mathcal{F}_{\mathsf{const}}$ is equivalent to semantic security, whereas KDM security w.r.t. $\mathcal{F}_{\mathsf{selec}}$ implies (and is actually stronger than) circular security.

Single-key vs. multiple-keys. Note that $\mathcal{F}$ is parameterized by an integer n, which indicates that the message might be dependent of n secret keys associated with n distinct identities. When $n = 1$, the corresponding notion is KDM security in the single-key setting. When $n > 1$, the corresponding notion is KDM security in the multi-key setting. Though it is desirable to attain KDM security in the multi-key setting, as pointed out by Wee [Wee16], considering KDM security in the single-key setting is still of great importance out of the following reasons: (1) it suffices for some natural applications; (2) it already captures much of the technical difficulty in attaining KDM security; (3) it could serve as a basis for the bootstrapping to KDM-security in the multi-key setting (in a non-black-box way).

4 KDM-secure IBE from KDM-secure PKE and $i\mathcal{O}$

As mentioned before, in contrast to few KDM results for IBE, there are fruitful KDM results for PKE. Thus, a promising idea is to translate the results of PKE to IBE. Observe that IBE can be viewed as an extension of PKE in which identity plays the role of public key and secret keys can be extracted from a master secret key, and thus the KDM security w.r.t. users secret key in identity-based setting is a mirror image of that in public-key setting. If there exists a structure-mirroring transformation from PKE to IBE (i.e., mapping an identity id to a public key pk, using secret key sk for pk as that for id, inheriting the same encryption/decryption algorithms of PKE), then we are able to compile a KDM-secure PKE into a KDM-secure IBE in a generic manner, while almost ignoring technical details of the starting PKE.

Recall that a PKE scheme consists of three polynomial time algorithms (KeyGen, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$), while an IBE scheme consists of four polynomial time algorithms (Setup, $\mathsf{Extract}$, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$). The key idea of the transformation is to map an identity to random coins, then invoke PKE.KeyGen with the obtained random coins to generate its corresponding public key. Such "id-to-pk" procedure must be done publicly without revealing the corresponding secret key, whereas with master secret key one can recover the random coins associated with any identity and then extracts the secret key. The encryption and decryption algorithms are essentially the same as that of the starting PKE. We implement the above idea by employing $i\mathcal{O}$ and puncturable PRF.

Let R be the randomness space of PKE.KeyGen, I be the desired identity space and PPRF be a puncturable PRF that maps I to R. The transformation works as follows:

- $\mathsf{Setup}(\lambda)$: run $k \leftarrow$ PPRF.$\mathsf{KeyGen}(\lambda)$, then create an obfuscation of circuit id-to-pk hash depicted in Figure 1. Finally, output the obfuscated circuit as mpk and k as msk.
- $\mathsf{Extract}(msk, id)$: on input msk and $id \in I$, compute $r \leftarrow$ PPRF.$\mathsf{Eval}(msk, id)$, $(pk, sk) \leftarrow$ PKE.$\mathsf{KeyGen}(\lambda; r)$, output sk as sk_{id} for id.
- $\mathsf{Encrypt}(mpk, id, m)$: run the obfuscated circuit mpk on input id to obtain its corresponding public key pk (write as $pk = mpk(id)$), then output $c \leftarrow$ PKE.$\mathsf{Encrypt}(pk, m)$.
- $\mathsf{Decrypt}(sk_{id}, c)$: output $m \leftarrow$ PKE.$\mathsf{Decrypt}(sk_{id}, c)$.

The correctness of the above IBE construction follows immediately from that of the starting PKE. For the security, we have the following theorem.

Theorem 4.1. *If the starting PKE is $\mathcal{F}$-KDM secure, the puncturable PRF is selective pseudorandom and the $i\mathcal{O}$ is secure, then the above IBE is $\mathcal{F}$-KDM secure in the selective-identity sense.*

Proof. For simplicity of exposition, we first prove this theorem in the single-key setting (there is only one target identity involved). Then we show how to adapt the proof to the multiple-keys setting. The proof proceeds via a sequence of games as below.

id-to-pk hash

Constants: PPRF key k

Input: id

1. Compute $r \leftarrow \text{PPRF}.\mathsf{Eval}(k, id)$, $(pk, sk) \leftarrow \text{PKE}.\mathsf{KeyGen}(r)$, and output pk.

Figure 1: id-to-pk hash takes as input id, and has constant a PPRF key k hardwired. The size of this circuit is padded to be the maximum of itself and id-to-pk hash* as described in Figure 2.

id-to-pk hash*

Constants: punctured PPRF key $k(\{id^*\})$, id^*, pk^*

Input: id

1. If $id = id^*$, output pk^*.
2. Else compute $r \leftarrow \text{PPRF}.\mathsf{Eval}(k(\{id^*\}), id)$, $(pk, sk) \leftarrow \text{PKE}.\mathsf{KeyGen}(r)$, output pk.

Figure 2: id-to-pk hash* takes as input id, and has constants a punctured PPRF key $k(\{id^*\})$ and identity id^* and public key pk^* hardwired.

Game 0 (the real game):

1. $\mathcal{A}$ commits the target identity id^* it will attack at the very beginning.
2. $\mathcal{CH}$ picks a fresh key k for PPRF as msk, creates an obfuscation of circuit id-to-pk hash as mpk, and sends mpk to $\mathcal{A}$. $\mathcal{CH}$ picks a random bit β, computes $pk^* \leftarrow mpk(id^*)$ and $sk_{id^*} \leftarrow \text{IBE}.\mathsf{Extract}(msk, id^*)$.
3. $\mathcal{A}$ then can make extraction and encryption queries, in the order of its choice.
   - Extraction query $\langle id \rangle$: for any $id \neq id^*$, $\mathcal{CH}$ responds with msk.
   - Encryption query $\langle f \rangle$[9]: $\mathcal{CH}$ responds with $c \leftarrow \text{PKE}.\mathsf{Encrypt}(pk^*, f(sk_{id^*}))$ if $\beta = 0$ or $c \leftarrow \text{PKE}.\mathsf{Encrypt}(pk^*, 0^{|m|})$ if $\beta = 1$.
4. Finally, $\mathcal{A}$ outputs a guess β' for β and wins if $\beta' = \beta$.

Game 1 (create an obfuscation of circuit id-to-pk hash* as mpk):

2. $\mathcal{CH}$ picks a fresh key k for PPRF as msk, computes $k(\{id^*\}) \leftarrow \text{PPRF}.\mathsf{Puncture}(k, id^*)$, $r^* \leftarrow \text{PPRF}.\mathsf{Eval}(k, id^*)$, $(pk^*, sk^*) \leftarrow \text{PKE}.\mathsf{KeyGen}(r^*)$, creates an obfuscation of circuit id-to-pk hash* as mpk. $\mathcal{CH}$ sets sk_{id^*} for id^* as sk^* and sends mpk to $\mathcal{A}$.

Game 2 (replace r^* with a uniformly random string over R):

2. $\mathcal{CH}$ picks $r^* \xleftarrow{\text{R}} R$ instead of computing $r^* \leftarrow \text{PPRF}.\mathsf{Eval}(k, id^*)$.

Lemma 4.2. *The advantages of any PPT adversary in Game 0 and Game 1 are negligibly close in λ, given the security of $i\mathcal{O}$.*

Proof. We prove this lemma by giving a reduction to the security of $i\mathcal{O}$. Suppose there is a PPT adversary $\mathcal{A}$ whose advantages in Game 0 and Game 1 are not negligibly close, then we can build an algorithm $\mathcal{B} = (\mathcal{S}, \mathcal{D})$ against the security of $i\mathcal{O}$ by interacting with $\mathcal{A}$ as follows.

[9] In the single-key setting the encryption query is associated with the only one target identity, thus we just refer to it as $\langle f \rangle$ and implicitly assume $n = 1$.

$\mathcal{S}(\lambda)$ behaves as follows: It invokes $\mathcal{A}$ to obtain the target identity id^*, then runs $k \leftarrow$ PPRF.$\mathsf{KeyGen}(\lambda)$, computes $k(\{id^*\}) \leftarrow$ PPRF.$\mathsf{Puncture}(k, id^*)$, $r^* \leftarrow$ PPRF.$\mathsf{Eval}(k, id^*)$, and $(pk^*, sk^*) \leftarrow$ PKE.$\mathsf{KeyGen}(\lambda; r^*)$. It sets $aux = (k, id^*, sk^*)$, then builds C_0 as the circuit id-to-pk hash and C_1 as the circuit id-to-pk hash*.

Before describing $\mathcal{D}$, we observe that by construction, the circuits C_0 and C_1 always behave identically on every input by the correctness of PPRF. With suitable padding, both C_0 and C_1 have the same size. Thus, $\mathcal{S}$ satisfies the conditions needed for invoking the indistinguishability property of $i\mathcal{O}$.

Now, we can describe the algorithm $\mathcal{D}$. Given aux and $i\mathcal{O}(C_b)$ as the challenge, $\mathcal{D}$ continues to interacts with $\mathcal{A}$ with the aim to determine b. To do so, $\mathcal{D}$ sets $mpk = i\mathcal{O}(C_b)$ and $msk = k$, picks a random bit β and sends mpk to $\mathcal{A}$. When $\mathcal{A}$ makes extraction queries $\langle id \rangle$, $\mathcal{D}$ responds normally with $msk = k$. When $\mathcal{A}$ makes encryption queries $\langle f \rangle$, $\mathcal{D}$ returns $c^* \leftarrow$ PKE.$\mathsf{Encrypt}(pk^*, f(sk^*))$ if $\beta = 0$ and $c^* \leftarrow$ PKE.$\mathsf{Encrypt}(pk^*, 0^{|m|})$ otherwise. Finally, $\mathcal{A}$ outputs a guess β' for β. If $\mathcal{A}$ wins, $\mathcal{D}$ outputs 1.

By construction, if $\mathcal{D}$ receives $i\mathcal{O}(C_0)$ (resp. $i\mathcal{O}(C_1)$), then the probability that $\mathcal{D}$ outputs 1 is exactly the probability of $\mathcal{A}$ winning in Game 0 (resp. Game 1). The lemma follows. □

Lemma 4.3. *The advantages of any PPT adversary in Game 1 and Game 2 are negligibly close, given the selective pseudorandomness of puncturable PRF.*

Proof. We prove this lemma by giving a reduction to selective pseudorandomness of PPRF. Suppose there is a PPT adversary $\mathcal{A}$ whose advantages in Game 1 and Game 2 are not negligibly close, then we can build an algorithm $\mathcal{B}$ that breaks the selective pseudorandomness of PPRF by interacting with $\mathcal{A}$ as follows.

$\mathcal{B}$ invokes $\mathcal{A}$ to obtain the target identity id^*, then submits id^* to its own PPRF challenger and receives back a punctured key $k(\{id^*\})$ as well as r^*, where r^* is either the real PPRF value at id^* or a uniformly random string over R. $\mathcal{B}$ then computes $(pk^*, sk^*) \leftarrow$ PKE.$\mathsf{KeyGen}(r^*)$, builds an obfuscation of the circuit id-to-pk hash* from $(k(\{id^*\}, id^*, pk^*)$ as mpk. $\mathcal{B}$ sends mpk to $\mathcal{A}$ and picks a random bit $\beta \in \{0, 1\}$. When $\mathcal{A}$ makes extraction queries $\langle id \rangle$ where $id \neq id^*$, $\mathcal{B}$ responds with $k(\{id^*\})$. When $\mathcal{A}$ makes encryption queries $\langle f \rangle$, $\mathcal{B}$ returns $c^* \leftarrow$ PKE.$\mathsf{Encrypt}(pk^*, f(sk^*))$ if $\beta = 0$ and $c^* \leftarrow$ PKE.$\mathsf{Encrypt}(pk^*, 0^{|m|})$ otherwise. Finally, $\mathcal{A}$ outputs a guess β' for β and wins if $\beta' = \beta$. If $\mathcal{A}$ wins, $\mathcal{B}$ outputs 1.

By the definitions of Game 1 and Game 2 and the correctness of PPRF, if $\mathcal{B}$ receives a real PRF value at id^* (resp. a random value over R), then the probability that $\mathcal{B}$ outputs 1 is exactly the probability of $\mathcal{A}$ winning in Game 1 (resp. Game 2). The lemma follows. □

Lemma 4.4. *The advantage of any PPT adversary in Game 2 is negligible, given the assumed KDM security of starting PKE.*

Proof. We prove this lemma by giving a reduction to the assumed KDM security of PKE. More precisely, suppose there is a PPT adversary $\mathcal{A}$ wins in Game 2 with non-negligible advantage, then we can build an algorithm $\mathcal{B}$ against the KDM security of PKE with the same advantage.

$\mathcal{B}$ receives pk^* from its PKE challenger, where pk^* is honestly generated by PKE.KeyGen under real random coins r^*. $\mathcal{B}$ invokes $\mathcal{A}$ to obtain the target identity id^*, then runs $k \leftarrow$ PPRF.$\mathsf{KeyGen}(\lambda)$, $k(\{id^*\}) \leftarrow$ PPRF.$\mathsf{Puncture}(k, id^*)$, builds the circuit id-to-pk hash* from $(k(\{id^*\}), id^*, pk^*)$ and computes its obfuscation as mpk. $\mathcal{B}$ then sends mpk to $\mathcal{A}$. Clearly, $\mathcal{B}$ can handle all extraction queries for $id \neq id^*$ with $msk = k$. When $\mathcal{A}$ makes encryption queries $\langle f \rangle$, $\mathcal{B}$ submits $\langle f \rangle$ to its own challenger and forwards the reply to $\mathcal{A}$. Finally, when $\mathcal{A}$ outputs its guess β' for β, $\mathcal{B}$ outputs β' to its PKE challenger.

By construction, a PKE encryption under pk^* is also an IBE encryption under id^* for the same underlying message, and thus $\mathcal{B}$ perfectly simulates Game 2. The lemma follows. □

Putting all the above together, the theorem follows immediately. □

Extension to the multiple-keys setting. The above theorem formally proves that if the starting PKE is KDM-secure in the single-key setting, so is the derived IBE. It is easy to verify that this result extends naturally to the multiple-keys setting. The transformation requires no change except that the size of circuit id-to-pk hash is padded to be the maximum of itself and id-to-pk hash* as specified in Figure 3. The formal proof can be easily adapted from that for Theorem 4.1 by setting the punctured set T as $\{id_1^*, \ldots, id_n^*\}$ rather than $\{id^*\}$ when introducing id-to-pk hash* to establish security.

id-to-pk hash*

Constants: punctured PPRF key $k(\{id_1^*, \ldots, id_n^*\})$, $id_1^*, \ldots, id_n^*$, $pk_1^*, \ldots, pk_n^*$

Input: id

1. If $id = id_i^*$, output pk_i^*.
2. Else compute $r \leftarrow \mathrm{PPRF}.\mathsf{Eval}(k(\{id_1^*, \ldots, id_n^*\}), id)$, $(pk, sk) \leftarrow \mathrm{PKE}.\mathsf{KeyGen}(r)$, output pk.

Figure 3: id-to-pk hash* takes as input id, and has constants a punctured PPRF key $k(\{id_1^*, \ldots, id_n^*\})$ and identities $(id_1^*, \ldots, id_n^*)$ and public keys $(pk_1^*, \ldots, pk_n^*)$ hardwired.

On the negative side, we have the following result.

Proposition 4.5. *If the starting PKE is n-circular insecure, then the above IBE is also n-circular insecure, even in the selective-identity sense.*

Proof. The proof for this proposition is in the same spirit of that for the above theorem. We sketch the rough idea as follows: The distribution of crooked public keys (generated using PRF values of identities) are computationally indistinguishable to that of real public keys (generated using true random coins). Therefore, the advantages of a PPT Test algorithm in these two cases are negligibly close. We omit the details here. □

Other applications. Thanks to the promising "structure-mirroring" property, our transformation also maintains many other security notions, such as security against chosen-ciphertext attacks, leakage attacks and tampering attacks. The security proofs can be easily adapted from that for Theorem 4.1 by simulating the decryption (resp. leakage and tampering) oracle of IBE via that of PKE.

A downside of this transformation lies in it only yields security results in the selective-identity sense, which seems intrinsic due to the use of "puncture programming technique" [SW14].

5 KDM-secure IBE from Homomorphic Identity-Based Hash Proof System

In this section, we present a generic construction of KDM-secure IBE from homomorphic IB-HPS. To the best of our knowledge, this is the first IBE scheme that attains KDM security in the adaptive-identity sense and multiple-keys setting.

5.1 Identity-Based Hash Proof System

In what follows, we adapt the notion of identity-based hash proof system (IB-HPS) from [ADN+10], in the context of group-theoretic languages. (In Definition 5.1 below we generalize the notion so that the language depends not only on security parameter λ but also on the identity id.)

Definition 5.1 (Identity-Based Hash Proof System)**.** An IB-HPS for $L \subset X$ consists of the following algorithms:

- $\mathsf{Setup}(\lambda)$: on input a security parameter λ, output master public key mpk and master secret key msk. We assume that mpk specifies a multiplicative group X, an identity space I, a collection of $\mathcal{NP}$ languages $L = \{L_{id}\}_{id \in I}$ defined over X, as well as an additive group Π. We require that X (resp. L_{id} and $X \backslash L_{id}$ for each $id \in I$) are efficiently samplable (w.l.o.g. obey uniform distribution) given mpk (resp. mpk and id), and denote the associate sampling algorithms by $\mathsf{SampAll}$, $\mathsf{SampYes}$ and SampNo respectively. Particularly, we require $\mathsf{SampYes}$ outputs an element $x \in L_{id}$ together with a witness w.
- $\mathsf{Extract}(msk, id)$: on input msk and an identity $id \in I$, output a secret key sk_{id}. This algorithm implicitly defines a binary relation $\mathsf{R} : SK \times I$, where $(sk_{id}, id) \in \mathsf{R}$ iff $\exists r$ s.t. $sk_{id} = \mathsf{Extract}(msk, id; r)$. Clearly, R is a many-to-one relation.
- $\mathsf{Priv}(sk_{id}, x)$: on input a secret key sk_{id} and an element $x \in X$, output $\pi \in \Pi$. This algorithm defines a family of hash functions $\Lambda = \{\Lambda_{sk_{id}} : X \to \Pi\}$ indexed by the set of secret keys for id.
- $\mathsf{Pub}(id, x, w)$: on input an identity $id \in I$ and an element $x \in L_{id}$ together with a witness w, output $\pi \in \Pi$.

Language membership assumption. Let X be a group fixed by mpk, and $L = \{L_{id}\}_{id \in I}$ be a collection of languages defined over X. The language membership assumption in the identity-based setting roughly states that for any $id \in I$ the uniform distributions over L_{id} and $X \backslash L_{id}$ are computationally indistinguishable. We now formally define it via the following experiment.

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[b = b' : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\lambda); \\ (state, id^*) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{ext}}(\cdot)}(mpk); \\ \beta \xleftarrow{\text{R}} \{0,1\}; \\ x_0^* \leftarrow \mathsf{SampYes}(mpk, id^*); \\ x_1^* \leftarrow \mathsf{SampNo}(mpk, id^*); \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ext}}(\cdot)}(state, x_\beta^*); \end{array} \right] - \frac{1}{2}.$$

Here $\mathcal{O}_{\mathsf{ext}}(\cdot)$ is an oracle that on input $id \in I$ returns a secret key $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$. We require that $\mathcal{O}_{\mathsf{ext}}(\cdot)$ returns the same secret key for repeated extraction queries on the same identity id.[10] Language membership assumption holds if for any PPT adversary, its advantage ϵ_{lmp} defined as above is negligible in λ. We stress that $\mathcal{A}$ is allowed to query $\mathcal{O}_{\mathsf{ext}}(\cdot)$ with any $id \in I$ (include id^*). This strengthening is crucial for attaining KDM security, as we will see shortly.

Remark 5.1. The language membership assumption in the identity-based setting defined as above requires that $U_{L_{id}} \approx_c U_{X \backslash L_{id}}$ for any $id \in I$. In some scenarios, it is useful to consider an alternative assumption, which requires that $U_{L_{id}} \approx_c U_X$ for any $id \in I$. Let $\rho_{id} = |L_{id}|/|X|$ be the density of L_{id}. It is easy to see that when ρ_{id} is negligible for any $id \in I$, the two assumptions are equivalent due to fact that U_X and $U_{X \backslash L_{id}}$ are statistically close.

[10] This restriction is natural yet necessary. If an adversary obtains multiple secret keys for the same identity, according to projective and smooth properties of Λ, it can break the language membership problem with high probability by checking if the hash values evaluated under these different secret keys are same.

In what follows, we define three properties of Λ.

Projection. We say Λ is projective if the action of $\Lambda_{sk_{id}}$ on L_{id} is determined by id, that is, for all $id \in I$ and all sk_{id} s.t. $(sk_{id}, id) \in \mathsf{R}$, and for all $x \in L_{id}$ with witness w, we have: $\Lambda_{sk_{id}}(x) = \mathsf{Pub}(id, x, w)$.

Homomorphic. We say Λ is homomorphic if for all $id \in I$ and all sk_{id} the function $\Lambda_{sk_{id}}$ is a group homomorphism from X to Π, that is, for all $x_1, x_2 \in X$, we have: $\Lambda_{sk_{id}}(x_1 \cdot x_2) = \Lambda_{sk_{id}}(x_1) + \Lambda_{sk_{id}}(x_2)$.

Smoothness. We say Λ is smooth if for all $id \in I$, we have:

$$(mpk, msk, x, \Lambda_{sk_{id}}(x)) \approx_s (mpk, msk, x, \pi)$$

where $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$, $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$, $x \xleftarrow{\text{R}} X \backslash L_{id}$, and $\pi \xleftarrow{\text{R}} \Pi$. The statistical distance is at most ϵ_{smooth}, which is negligible in λ.

Remark 5.2. When ρ_{id} is negligible for any $id \in I$, the smoothness also holds w.r.t. $x \xleftarrow{\text{R}} X$.

5.2 KDM Secure IBE from Homomorphic IB-HPS

Construction. Starting from an IB-HPS whose projective hashing $\Lambda : X \to \Pi$ is smooth and homomorphic, we can derive a KDM-secure IBE scheme with the same identity set and message space $M = \Pi$. The construction is as below.

- The Setup and $\mathsf{Extract}$ algorithms are identical to that of the starting IB-HPS.
- $\mathsf{Encrypt}(mpk, id, m)$: on input mpk and an identity id and a message m, run $(x, w) \leftarrow \mathsf{SampYes}(mpk, id)$, compute $\pi \leftarrow \mathsf{Pub}(id, x, w)$, $y \leftarrow \pi + m$, output $c = (x, y)$.
- $\mathsf{Decrypt}(sk_{id}, c)$: on input sk_{id} and a ciphertext c, parse $c = (x, y)$, compute $\pi \leftarrow \mathsf{Priv}(sk_{id}, x)$, output $m = y - \pi$.

The correctness of the above construction follows readily from the projective property of Λ. For the security, we have the following theorem.

Theorem 5.1. *Suppose the projective hash $\Lambda : X \to \Pi$ is smooth and homomorphic, and the language membership problem is hard. Then the above construction is $\mathcal{F}$-KDM secure where $\mathcal{F} = \{f_{u,v} : sk \to \phi^{-1}(\Lambda_{sk}(u) + v)\}_{u \in X, v \in \Pi}$.*

Proof. For technical convenience, we will stick to the second language membership assumption, i.e., $U_{L_{id}}$ and U_X are computationally indistinguishable for any $id \in I$. We prove the above theorem via a sequence of games. An overview of the security proof is depicted in Figure 4.

Game 0: This game corresponds to the KDM security game that for each encryption query $\langle f \rangle$ to $\mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot)$, $\mathcal{CH}$ always responds with real KDM encryption, i.e., $\mathcal{O}^{0}_{\mathsf{enc}}(f) = \mathsf{Encrypt}(mpk, id^*, f(sk_{id^*}))$.

Game 1: This game corresponds to the KDM security game that for each encryption query $\langle f \rangle$ to $\mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot)$, $\mathcal{CH}$ always responds with simulated encryption $\mathcal{O}_{\mathsf{sim}}(f)$. Suppose f is indexed by (u, v), i.e., $f_{u,v}(sk) = \Lambda_{sk}(u) + v$, $\mathcal{O}_{\mathsf{sim}}(f)$ samples $x^* \xleftarrow{\text{R}} L_{id^*}$ with witness w^*, outputs $(x^* \cdot u^{-1}, \mathsf{Pub}(id^*, x^*, w^*) + v)$.

Game 2: This game corresponds to the KDM security game that for each encryption query $\langle f \rangle$ to $\mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot)$, $\mathcal{CH}$ always responds with zero encryption, i.e., $\mathcal{O}^{1}_{\mathsf{enc}}(f) = \mathsf{Encrypt}(mpk, id^*, 0^{|m|})$.

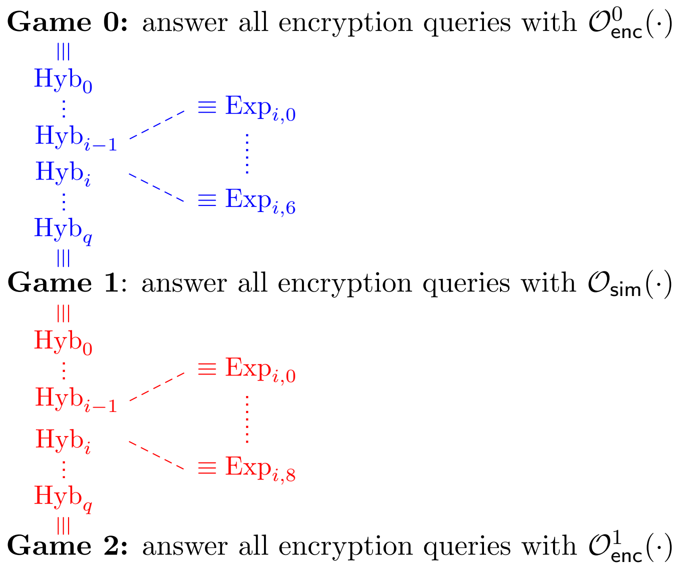

Figure 4: An overview of the security proof.

To establish the desired KDM security, it suffices to show that Game 0 and Game 2 are computationally indistinguishable. To this end, we show both Game 0 and Game 2 are computationally indistinguishable from the intermediate Game 1. Without loss of generality, we assume the maximum number of encryption queries made by the adversary is upper bounded by a polynomial q in λ.

Lemma 5.2. *Game 0 and Game 1 are computationally indistinguishable.*

Proof. We introduce $q+1$ hybrids indexed by $0 \leq i \leq q$ between Game 0 and Game 1, where in Hyb_i the first i encryption queries are answered with $\mathcal{O}_{\mathsf{sim}}(\cdot)$ and the rest encryption queries are answered with $\mathcal{O}^0_{\mathsf{enc}}(\cdot)$. Clearly, Hyb_0 is exactly Game 0 and Hyb_q is exactly Game 1. In what follows, we show that for each $1 \leq i \leq q$, Hyb_{i-1} and Hyb_i are computationally indistinguishable. Note that these two successive hybrids only differ at the response to the i-th encryption query $\langle f_i \rangle$, the crux is to show that:

$$\mathcal{O}^0_{\mathsf{enc}}(f_i) \approx_c \mathcal{O}_{\mathsf{sim}}(f_i)$$

To this end, we further introduce seven experiments (from $\text{Exp}_{i,0}$ to $\text{Exp}_{i,6}$) between each successive Hyb_{i-1} and Hyb_i. In all the seven intermediate experiments, the first $i-1$ encryption queries are answered with $\mathcal{O}_{\mathsf{sim}}(\cdot)$, and the last $q-i$ encryption queries are answered with $\mathcal{O}^0_{\mathsf{enc}}(\cdot)$. They only differ at the response to the i-th encryption query $\langle f_i = f_{u_i,v_i} \rangle$ as highlighted below.

As depicted in Figure 5, we need to show $\text{Exp}_{i,2} \approx_c \text{Exp}_{i,3}$ and $\text{Exp}_{i,4} \approx_c \text{Exp}_{i,5}$ based on the language membership assumption. Recall that a reduction to the language membership problem knows exactly one secret key for any $id \in I$ even including the target identity. This allows us to carry out hybrid arguments between $\text{Exp}_{i,2}, \text{Exp}_{i,3}$ and $\text{Exp}_{i,4}, \text{Exp}_{i,5}$.

$\text{Exp}_{i,0}$ (identical to Hyb_{i-1}): $\mathcal{CH}$ interacts with $\mathcal{A}$ as follows.

1. Run $\mathsf{Setup}(\lambda)$ to generate (mpk, msk), send mpk to $\mathcal{A}$.
2. On extraction query $\langle id \rangle$, return $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$.
3. $\mathcal{A}$ chooses id^* as the target identity. $\mathcal{CH}$ computes $sk_{id^*} \leftarrow \mathsf{Extract}(msk, id^*)$,

$$
\begin{array}{lll}
& \mathcal{O}^0_{\mathsf{enc}}(f_i) & \\
\equiv & (x^*, \mathsf{Pub}(id^*, x^*, w^*) + \Lambda_{sk_{id^*}}(u_i) + v_i) & \mathrm{Exp}_{i,0} : x^* \xleftarrow{\mathrm{R}} L_{id^*} \\
\equiv & (x^*, \underline{\Lambda_{sk_{id^*}}(x^*)} + \Lambda_{sk_{id^*}}(u_i) + v_i) & \mathrm{Exp}_{i,1} : x^* \xleftarrow{\mathrm{R}} L_{id^*}, \text{via projective property} \\
\equiv & (x^*, \underline{\Lambda_{sk_{id^*}}(x^* \cdot u_i)} + v_i) & \mathrm{Exp}_{i,2} : x^* \xleftarrow{\mathrm{R}} L_{id^*}, \text{via homomorphism} \\
\approx_c & (x^*, \Lambda_{sk_{id^*}}(x^* \cdot u_i) + v_i) & \mathrm{Exp}_{i,3} : \underline{x^* \xleftarrow{\mathrm{R}} X}, \text{via language membership} \\
\equiv & (\underline{x^* \cdot u_i^{-1}}, \underline{\Lambda_{sk_{id^*}}(x^*)} + v_i) & \mathrm{Exp}_{i,4} : x^* \xleftarrow{\mathrm{R}} X, X \text{ is a group} \\
\approx_c & (x^* \cdot u_i^{-1}, \Lambda_{sk_{id^*}}(x^*) + v_i) & \mathrm{Exp}_{i,5} : \underline{x^* \xleftarrow{\mathrm{R}} L_{id^*}}, \text{via language membership} \\
\equiv & (x^* \cdot u_i^{-1}, \underline{\mathsf{Pub}(id^*, x^*, w^*)} + v_i) & \mathrm{Exp}_{i,6} : x^* \xleftarrow{\mathrm{R}} L_{id^*}, \text{via projective property} \\
& \mathcal{O}_{\mathsf{sim}}(f_i) &
\end{array}
$$

Figure 5: Transitions between $\mathcal{O}^0_{\mathsf{enc}}(f_i)$ and $\mathcal{O}_{\mathsf{sim}}(f_i)$

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)$, computes $y^* \leftarrow \mathsf{Pub}(id^*, x^*, w^*) + \Lambda_{sk_{id^*}}(u_i) + v_i$, returns $c^* = (x^*, y^*)$. Besides, the first $i-1$ encryption queries are answered with $\mathcal{O}_{\mathsf{sim}}(\cdot)$, while the last $q-i$ encryption queries are answered with $\mathcal{O}^0_{\mathsf{enc}}(\cdot)$.
5. On extraction query $\langle id \rangle$ where $id \neq id^*$, $\mathcal{CH}$ responds the same way as in Phase 1.

$\mathrm{Exp}_{i,1}$ (compute $\Lambda_{sk_{id^*}}(x^*)$ privately): $\mathrm{Exp}_{i,1}$ is identical to $\mathrm{Exp}_{i,0}$ except that $\mathcal{CH}$ computes $\Lambda_{sk_{id^*}}(x^*)$ privately in step 4.

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)$, computes $y^* \leftarrow \underline{\Lambda_{sk_{id^*}}(x^*)} + \Lambda_{sk_{id^*}}(u_i) + v_i$, returns $c^* = (x^*, y^*)$.

$\mathrm{Exp}_{i,2}$ (compute y^* via homomorphism): $\mathrm{Exp}_{i,2}$ is identical to $\mathrm{Exp}_{i,1}$ except that $\mathcal{CH}$ computes y^* via homomorphism in step 4.

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)$, computes $\underline{y^* \leftarrow \Lambda_{sk_{id^*}}(x^* \cdot u_i) + v_i}$, returns $c^* = (x^*, y^*)$.

$\mathrm{Exp}_{i,3}$ (sample x^* from X): $\mathrm{Exp}_{i,3}$ is identical to $\mathrm{Exp}_{i,2}$ except that $\mathcal{CH}$ samples $x^* \xleftarrow{\mathrm{R}} X$.

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ picks $\underline{x^* \leftarrow \mathsf{SampAll}(mpk)}$, then computes $y^* \leftarrow \Lambda_{sk_{id^*}}(x^* \cdot u_i) + v_i$, returns $c^* = (x^*, y^*)$.

$\mathrm{Exp}_{i,4}$ (replace x^* with $x^* \cdot u_i^{-1}$): $\mathrm{Exp}_{i,4}$ is identical to $\mathrm{Exp}_{i,3}$ except that $\mathcal{CH}$ replaces x^* with $x^* \cdot u_i^{-1}$ in the ciphertext.

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ first picks $x^* \leftarrow \mathsf{SampAll}(mpk)$, then computes $\underline{y^* \leftarrow \Lambda_{sk_{id^*}}(x^*) + v_i}$, returns $\underline{c^* = (x^* \cdot u_i^{-1}, y^*)}$.

$\mathrm{Exp}_{i,5}$ (sample x^* from L_{id^*}): $\mathrm{Exp}_{i,5}$ is identical to $\mathrm{Exp}_{i,4}$ except that $\mathcal{CH}$ samples x^* from L_{id^*} instead of X.

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $\underline{(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)}$, computes $y^* \leftarrow \Lambda_{sk_{id^*}}(x^*) + v_i$, returns $c^* = (x^* \cdot u_i^{-1}, y^*)$.

$\mathrm{Exp}_{i,6}$ (compute $\Lambda_{sk_{id^*}}(x^*)$ publicly): $\mathrm{Exp}_{i,6}$ is identical to $\mathrm{Exp}_{i,5}$ except that $\mathcal{CH}$ compute $\Lambda_{sk_{id^*}}(x^*)$ publicly.

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)$, computes $y^* \leftarrow \mathsf{Pub}(id^*, x^*, w^*) + v_i$, returns $c^* = (x^* \cdot u_i^{-1}, y^*)$.

The differences between $\mathrm{Exp}_{i,0}$ and $\mathrm{Exp}_{i,1}$, $\mathrm{Exp}_{i,1}$ and $\mathrm{Exp}_{i,2}$, $\mathrm{Exp}_{i,3}$ and $\mathrm{Exp}_{i,4}$, $\mathrm{Exp}_{i,5}$ and $\mathrm{Exp}_{i,6}$ are only conceptual. Therefore, they are perfectly equivalent.

Claim 5.3. $\mathrm{Exp}_{i,2}$ *and* $\mathrm{Exp}_{i,3}$ *are computationally indistinguishable, given the hardness of the language membership problem.*

Proof. Suppose there is an adversary $\mathcal{A}$ that can distinguish $\mathrm{Exp}_{i,2}$ and $\mathrm{Exp}_{i,3}$, we show how to build an algorithm $\mathcal{B}$ breaks the language membership problem. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

1. Given mpk from its own challenger where $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$, $\mathcal{B}$ sends mpk to $\mathcal{A}$.
2. On extraction query $\langle id \rangle$, $\mathcal{B}$ forwards the query to its own challenger and sends the reply to $\mathcal{A}$.
3. $\mathcal{A}$ chooses id^* as the target identity. $\mathcal{B}$ submits id^* to its own challenger and receives back x^*, which is either sampled from L_{id^*} or X. $\mathcal{B}$ also makes an extraction query $\langle id^* \rangle$ and receives back sk_{id^*}.
4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{B}$ computes $y^* \leftarrow \Lambda_{sk_{id^*}}(x^* \cdot u_i) + v_i$, sends $c^* = (x^*, y^*)$ to $\mathcal{A}$. Besides, $\mathcal{B}$ answers the first $i-1$ encryption queries with $\mathcal{O}_{\mathsf{sim}}(\cdot)$, and the last $q-i$ encryption queries with $\mathcal{O}^0_{\mathsf{enc}}(\cdot)$. Since $\mathcal{B}$ can obtain a secret key for any identities by querying its challenger, it is able handle all the encryption queries properly.
5. On extraction query $\langle id \rangle$ where $id \neq id^*$, $\mathcal{B}$ responds the same way as in Phase 1.

It is easy to see that if $x^* \xleftarrow{\text{R}} L_{id^*}$, $\mathcal{B}$ simulates $\mathrm{Exp}_{i,2}$ perfectly; if $x^* \xleftarrow{\text{R}} X$, $\mathcal{B}$ simulates $\mathrm{Exp}_{i,3}$ perfectly. Therefore, $\mathcal{B}$ breaks the language membership problem with the same advantage as $\mathcal{A}$ distinguishing $\mathrm{Exp}_{i,2}$ and $\mathrm{Exp}_{i,3}$. This proves Claim 5.3. $\square$

Claim 5.4. $\mathrm{Exp}_{i,4}$ *and* $\mathrm{Exp}_{i,5}$ *are computationally indistinguishable, given the hardness of the language membership problem.*

Proof. We omit the proof since it is similar to that for Claim 5.3. $\square$

Note that $\mathrm{Exp}_{i,0}$ is exactly Hyb_{i-1}, while $\mathrm{Exp}_{i,6}$ is exactly Hyb_i. Combining all these above, we have $|\mathsf{Adv}_{\mathcal{A}}(\mathrm{Hyb}_i) - \mathsf{Adv}_{\mathcal{A}}(\mathrm{Hyb}_{i-1})| \leq 2 \cdot \epsilon_{\mathrm{lmp}}$ for each $i \in [q]$, and thus $|\mathsf{Adv}_{\mathcal{A}}(\text{Game 1}) - \mathsf{Adv}_{\mathcal{A}}(\text{Game 0})| \leq 2q \cdot \epsilon_{\mathrm{lmp}}$. This proves Lemma 5.2. $\square$

Lemma 5.5. *Game 1 and Game 2 are computationally indistinguishable.*

Proof. We introduce $q+1$ hybrids indexed by $0 \leq i \leq q$ between Game 1 and Game 2, where in Hyb_i the first i encryption queries are answered with $\mathcal{O}^1_{\mathsf{enc}}(\cdot)$ and the rest encryption queries are answered with $\mathcal{O}_{\mathsf{sim}}(\cdot)$. Clearly, Hyb_0 is exactly Game 1 and Hyb_q is exactly Game 2. In what follows, we show that for each $1 \leq i \leq q$, Hyb_{i-1} and Hyb_i are computationally indistinguishable. Note that these two each successive hybrids only differ at the response to the i-th encryption queries, the crux is to show that:

$$\mathcal{O}_{\mathsf{sim}}(f_i) \approx_c \mathcal{O}^1_{\mathsf{enc}}(f_i)$$

To this end, we further introduce nine experiments (from $\mathrm{Exp}_{i,0}$ to $\mathrm{Exp}_{i,8}$) between each successive hybrids Hyb_{i-1} and Hyb_i. In all the nine intermediate experiments, the first $i-1$ encryption queries are answered with $\mathcal{O}^1_{\mathsf{enc}}(\cdot)$, while the last $q-i$ encryption queries are answered with $\mathcal{O}_{\mathsf{sim}}(\cdot)$. They only differ at the response to the i-th encryption query $\langle f_i = f_{u_i,v_i} \rangle$ as highlighted below.

$$
\begin{array}{lll}
 & \mathcal{O}_{\mathsf{sim}}(f_i) & \\
\equiv & (x^* \cdot u_i^{-1}, \mathsf{Pub}(id^*, x^*, w^*) + v_i) & \mathrm{Exp}_{i,0} : x^* \xleftarrow{R} L_{id^*} \\
\equiv & (x^* \cdot u_i^{-1}, \underline{\Lambda_{sk_{id^*}}(x^*)} + v_i) & \mathrm{Exp}_{i,1} : x^* \xleftarrow{R} L_{id^*}, \text{via projective property} \\
\approx_c & (x^* \cdot u_i^{-1}, \Lambda_{sk_{id^*}}(x^*) + v_i) & \mathrm{Exp}_{i,2} : \underline{x^* \xleftarrow{R} X}, \text{via language membership} \\
\approx_s & (x^* \cdot u_i^{-1}, \underline{\pi^*} + v_i) & \mathrm{Exp}_{i,3} : \pi^* \xleftarrow{R} \Pi, \text{via smoothness} \\
\equiv & (\underline{x^*}, \pi^* + v_i) & \mathrm{Exp}_{i,4} : x^* \xleftarrow{R} X, X \text{ is a group} \\
\equiv & (x^*, \pi^* + \underline{0^{|m|}}) & \mathrm{Exp}_{i,5} : \pi^* \xleftarrow{R} \Pi, \Pi \text{ is a group} \\
\approx_s & (x^*, \underline{\Lambda_{sk_{id^*}}(x^*)} + 0^{|m|}) & \mathrm{Exp}_{i,6} : \text{via smoothness} \\
\approx_c & (x^*, \Lambda_{sk_{id^*}}(x^*) + 0^{|m|}) & \mathrm{Exp}_{i,7} : \underline{x^* \xleftarrow{R} L_{id^*}}, \text{via language membership} \\
\equiv & (x^*, \underline{\mathsf{Pub}(id^*, x^*, w^*)} + 0^{|m|}) & \mathrm{Exp}_{i,8} : x^* \xleftarrow{R} L_{id^*}, \text{via projective property} \\
 & \mathcal{O}^1_{\mathsf{enc}}(f_i) &
\end{array}
$$

Figure 6: Transitions between $\mathcal{O}_{\mathsf{sim}}(f_i)$ and $\mathcal{O}^1_{\mathsf{enc}}(f_i)$

As depicted in Figure 6, we need to prove $\mathrm{Exp}_{i,1} \approx_c \mathrm{Exp}_{i,2}$ and $\mathrm{Exp}_{i,6} \approx_c \mathrm{Exp}_{i,7}$ based on the language membership assumption, and show $\mathrm{Exp}_{i,2} \approx_s \mathrm{Exp}_{i,3}$ and $\mathrm{Exp}_{i,5} \approx_s \mathrm{Exp}_{i,6}$ based on the smoothness of Λ. Similar to previous analysis, a reduction to the language membership assumption knows exactly one secret key for any $id \in I$. This fact allows us to carry out hybrid arguments between $\mathrm{Exp}_{i,1}, \mathrm{Exp}_{i,2}$ and $\mathrm{Exp}_{i,6}, \mathrm{Exp}_{i,7}$. In addition, throughout $\mathrm{Exp}_{i,2}$ and $\mathrm{Exp}_{i,5}$, the information of sk_{id^*} is not leaked elsewhere except when answering i-th encryption query, thus we can safely apply smoothness of Λ for the transitions between $\mathrm{Exp}_{i,2}, \mathrm{Exp}_{i,3}$ and $\mathrm{Exp}_{i,5}, \mathrm{Exp}_{i,6}$.

$\mathrm{Exp}_{i,0}$ (identical to Hyb_{i-1}): $\mathcal{CH}$ interacts with $\mathcal{A}$ as follows.

1. Run $\mathsf{Setup}(\lambda)$ to generate (mpk, msk), send mpk to $\mathcal{A}$.
2. On extraction query $\langle id \rangle$, return $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$.
3. $\mathcal{A}$ chooses id^* as the target identity. $\mathcal{CH}$ computes $sk_{id^*} \leftarrow \mathsf{Extract}(msk, id^*)$.
4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)$, computes $y^* \leftarrow \mathsf{Pub}(id^*, x^*, w^*) + v_i$, returns $c^* = (x^* \cdot u_i^{-1}, y^*)$. Besides, $\mathcal{B}$ answers the first $i-1$ encryption queries with $\mathcal{O}^1_{\mathsf{enc}}(\cdot)$, and the last $q-i$ encryption queries with $\mathcal{O}_{\mathsf{sim}}(\cdot)$.
5. On extraction query $\langle id \rangle$ where $id \neq id^*$, $\mathcal{CH}$ responds the same way as in Phase 1.

$\mathrm{Exp}_{i,1}$ (compute $\Lambda_{sk_{id^*}}(x^*)$ privately): $\mathrm{Exp}_{i,1}$ is identical to $\mathrm{Exp}_{i,0}$ except that $\mathcal{CH}$ computes $\Lambda_{sk_{id^*}}(x^*)$ privately in step 4.

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)$, computes $y^* \leftarrow \underline{\Lambda_{sk^*_{id}}(x^*)} + v_i$, sends $c^* = (x^* \cdot u_i^{-1}, y^*)$ to $\mathcal{A}$.

$\mathrm{Exp}_{i,2}$ (sample x^* from X): $\mathrm{Exp}_{i,2}$ is identical to $\mathrm{Exp}_{i,1}$ except that $\mathcal{CH}$ samples $x^* \xleftarrow{R} X$.

4. On the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ first picks $\underline{x^* \leftarrow \mathsf{SampAll}(mpk)}$, then computes $y^* \leftarrow \Lambda_{sk_{id^*}}(x^*) + v_i$, sends $c^* = (x^* \cdot u_i^{-1}, y^*)$ to $\mathcal{A}$.

$\mathrm{Exp}_{i,3}$ (replace $\Lambda_{sk_{id^*}}(x^*)$ with $\pi^* \xleftarrow{R} \Pi$): $\mathrm{Exp}_{i,3}$ is identical to $\mathrm{Exp}_{i,2}$ except that $\mathcal{CH}$ replaces $\Lambda_{sk_{id^*}}(x^*)$ with $\pi^* \xleftarrow{R} \Pi$.

4. For the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ picks $x^* \leftarrow \mathsf{SampAll}(mpk)$, picks $\pi^* \xleftarrow{\text{R}} \Pi$, computes $y^* \leftarrow \underline{\pi^*} + v_i$, sends $c^* = (x^* \cdot u_i^{-1}, y^*)$ to $\mathcal{A}$.

$\text{Exp}_{i,4}$ (replace $x^* \cdot u_i^{-1}$ with x^*): $\text{Exp}_{i,4}$ is identical to $\text{Exp}_{i,3}$ except that $\mathcal{CH}$ replaces $x^* \cdot u_i^{-1}$ with x^* in the ciphertext.

4. For the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ picks $x^* \leftarrow \mathsf{SampAll}(mpk)$, picks $\pi^* \xleftarrow{\text{R}} \Pi$, computes $y^* \leftarrow \pi^* + v_i$, sends $c^* = (\underline{x^*}, y^*)$ to $\mathcal{A}$.

$\text{Exp}_{i,5}$ (replace v_i with $0^{|m|}$): $\text{Exp}_{i,5}$ is identical to $\text{Exp}_{i,4}$ except that $\mathcal{CH}$ replaces v_i with $0^{|m|}$ in the ciphertext.

4. For the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $x^* \leftarrow \mathsf{SampAll}(mpk)$, picks $\pi^* \xleftarrow{\text{R}} \Pi$, computes $y^* \leftarrow \pi^* + \underline{0^{|m|}}$, sends $c^* = (x^*, y^*)$ to $\mathcal{A}$.

$\text{Exp}_{i,6}$ (replace π^* with $\Lambda_{sk_{id^*}}(x^*)$): $\text{Exp}_{i,6}$ is identical to $\text{Exp}_{i,5}$ except that $\mathcal{CH}$ replaces π^* with $\Lambda_{sk_{id^*}}(x^*)$ when computing y^* in the ciphertext.

4. For the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ picks $x^* \leftarrow \mathsf{SampAll}(mpk)$, then computes $y^* \leftarrow \underline{\Lambda_{sk_{id^*}}(x^*)} + 0^{|m|}$, sends $c^* = (x^*, y^*)$ to $\mathcal{A}$.

$\text{Exp}_{i,7}$ (sample x^* from L_{id^*}): $\text{Exp}_{i,7}$ is identical to $\text{Exp}_{i,6}$ except that $\mathcal{CH}$ samples $x^* \xleftarrow{\text{R}} L_{id^*}$.

4. For the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $\underline{(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)}$, then computes $y^* \leftarrow \Lambda_{sk_{id^*}}(x^*) + 0^{|m|}$, sends $c^* = (x^*, y^*)$ to $\mathcal{A}$.

$\text{Exp}_{i,8}$ (compute $\Lambda_{sk_{id^*}}(x^*)$ publicly): $\text{Exp}_{i,8}$ is identical to $\text{Exp}_{i,7}$ except that $\mathcal{CH}$ compute $\Lambda_{sk_{id^*}}(x^*)$ publicly.

4. For the i-th encryption query $\langle f_i \rangle$, $\mathcal{CH}$ runs $(x^*, w^*) \leftarrow \mathsf{SampYes}(mpk, id^*)$, then computes $y^* \leftarrow \underline{\mathsf{Pub}(id^*, x^*, w^*)} + 0^{|m|}$, sends $c^* = (x^*, y^*)$ to $\mathcal{A}$.

The differences between $\text{Exp}_{i,0}$ and $\text{Exp}_{i,1}$, $\text{Exp}_{i,3}$ and $\text{Exp}_{i,4}$, $\text{Exp}_{i,4}$ and $\text{Exp}_{i,5}$, $\text{Exp}_{i,7}$ and $\text{Exp}_{i,8}$ are only conceptual. Therefore, they are perfectly equivalent. $\text{Exp}_{i,2}$ and $\text{Exp}_{i,3}$ (resp. $\text{Exp}_{i,5}$ and $\text{Exp}_{i,6}$) are statistically close due to the smoothness of Λ.

Claim 5.6. *$\text{Exp}_{i,1}$ and $\text{Exp}_{i,2}$ are computationally indistinguishable, given the hardness of the language membership problem.*

Claim 5.7. *$\text{Exp}_{i,6}$ and $\text{Exp}_{i,7}$ are computationally indistinguishable given the hardness of the language membership problem.*

Proof. We omit the detailed proofs of Claim 5.6 and 5.7 here since they are similar to that for Claim 5.3. □

According to the definition, $\text{Exp}_{i,0}$ is exactly Hyb_{i-1}, while $\text{Exp}_{i,8}$ is exactly Hyb_i. Combining all these above, we have that $|\mathsf{Adv}_{\mathcal{A}}(\text{Hyb}_i) - \mathsf{Adv}_{\mathcal{A}}(\text{Hyb}_{i-1})| \leq 2\epsilon_{\text{smooth}} + 2\epsilon_{\text{lmp}}$ for each $i \in [q]$, and thus $|\mathsf{Adv}_{\mathcal{A}}(\text{Game 2}) - \mathsf{Adv}_{\mathcal{A}}(\text{Game 1})| \leq q \cdot (2\epsilon_{\text{smooth}} + 2\epsilon_{\text{lmp}})$. This proves Lemma 5.5. □

Putting Lemma 5.2 and Lemma 5.5 together, we have $|\mathsf{Adv}_{\mathcal{A}}(\text{Game 2}) - \mathsf{Adv}_{\mathcal{A}}(\text{Game 0})| \leq q(2\epsilon_{\text{smooth}} + 4\epsilon_{\text{lmp}})$. This proves the theorem. □

Remark 5.3. So far, we do not know how to extend the above construction to the multi-key setting. The technical difficulty lies is that in the multi-key setting the encrypted message could be a function of secret keys for multiple identities. More precisely, consider $\mathbf{id} = (id_1^*, \ldots, id_n^*)$ and $f(sk_1, \ldots, sk_n) = \Lambda_{sk_1}(u_1) + \cdots + \Lambda_{sk_n}(u_n) + v$ where $u_i \in X, v \in \Pi$. The real KDM encryption for KDM query $\langle i, f \rangle$ is of the form $(x^*, \mathsf{Pub}(id_i^*, x^*, w^*) + \sum_{j \neq i} \Lambda_{sk_j^*}(u_j) + v)$, where $x^* \xleftarrow{\text{R}} L_{id_i}^*$. In this scenario, it appears that the homomorphic property of Λ combining with the language membership problem only allow us to simulate the real KDM encryption without using sk_i^* for id_i^*. The involvement of secret keys for other target identities seems unavoidable in the simulated encryption. As a consequence, we can not apply smoothness of Λ to argue the indistinguishability between simulated encryption and zero encryption, as we did in the single-identity setting. We left the extension to the multi-key setting as an interesting open problem.

6 KDM-secure IBE from $i\mathcal{O}$ and Puncturable Unique Signature

In the PKE setting, the largest possible KDM function family is the ensemble of circuits of a-priori bounded size. Though we have shown how to construct KDM-secure IBE scheme w.r.t. reasonable function family in the preceding section, seeking direct construction of bounded KDM-secure IBE is still of great interest on its own right.

In this section, we propose a bounded KDM-secure IBE from $i\mathcal{O}$ and puncturable unique signature, which is delicately adapted from the recent work by [MPS16]. Before presenting our construction, we first introduce the new notion named puncturable unique signature and show how to construct it.

6.1 Puncturable Unique Signature

We introduce a new notion named puncturable unique signature (PUS), which adds the possibility to derive punctured signing keys to unique signature [Lys02].

Definition 6.1 (Puncturable Unique Signature)**.** A PUS scheme consists of four polynomial algorithms as follows:

- $\mathsf{Setup}(\lambda)$: on input a security parameter λ, output a verification key vk and a signing key sk. We assume vk includes the descriptions of the message space M and the signature space Σ.
- $\mathsf{Puncture}(sk, m)$: on input sk and a message m^*, output a punctured signing key $sk(\{m^*\})$, which enables signing all messages but m^*.
- $\mathsf{Sign}(sk, m)$: if sk is the normal signing key, output a signature σ for m; if sk is a punctured signing key $sk(\{m^*\})$, output a signature σ for m if $m \neq m^*$ and $\perp$ otherwise.
- $\mathsf{Verify}(vk, m, \sigma)$: on input vk, m and σ, output "1" to indicate σ is a valid signature of m and "0" otherwise.

We require the following properties:

Uniqueness of signature. For all $(vk, sk) \leftarrow \mathsf{KeyGen}(\lambda)$ and all $m \in M$, there do not exists values $\sigma_1, \sigma_2 \in \Sigma$ such that $\sigma_1 \neq \sigma_2$ and $\mathsf{Verify}(vk, m, \sigma_1) = \mathsf{Verify}(vk, m, \sigma_2) = 1$.

Unforgeability. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against PUS and define its advantage in the following experiment:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\mathsf{Verify}(vk, m^*, \sigma^*) = 1 : \begin{array}{l} (vk, sk) \leftarrow \mathsf{KeyGen}(\lambda); \\ (state, m^*) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{sign}}(\cdot)}(vk); \\ sk(\{m^*\}) \leftarrow \mathsf{Puncture}(sk, m^*); \\ \sigma^* \leftarrow \mathcal{A}_2(state, sk(\{m^*\})); \end{array}\right] - \frac{1}{2},$$

where $\mathcal{O}_{\mathsf{sign}}(\cdot)$ is an oracle that on input $m \in M$ returns $\sigma \leftarrow \mathsf{Sign}(sk, m)$, and $\mathcal{A}_1$ is not allowed to choose the message that has been queried for signatures as the target one. A PUS is said to be unforgeable if for any PPT adversary $\mathcal{A}$, its advantage defined as above is negligible in λ.

Constructions of PUS. Interestingly, we observe that the short signature from OWF and $i\mathcal{O}$ by Sahai and Waters [SW14] exactly constitutes a PUS if the underlying OWF is injective. This provides us a direct construction of PUS.

On the other hand, as noted in [Lys02], the construction of unique signature from verifiable random functions (VRFs) is immediate if the proofs in the VRFs are unique. Likewise, PUS is immediately implied by punctured VRFs satisfying the uniqueness of proofs, where punctured PRFs itself is a special class of constrained VRFs [Fuc14]. Inspection of the circuit-constrained VRF construction presented in [Fuc14] reveals that it has unique proof. This provides us an indirect construction of PUS.

6.2 Bounded KDM-secure IBE Scheme

Construction. Let PUS be a puncturable unique signature with message space I, and $i\mathcal{O}$ be an indistinguishability obfuscator. Our construction is as below:

- $\mathsf{Setup}(\lambda)$: run $(vk, sk) \leftarrow \text{PUS}.\mathsf{KeyGen}(\lambda)$, output $mpk = vk$, $msk = sk$.
- $\mathsf{Extract}(msk, id)$: on input msk and $id \in I$, run $\sigma \leftarrow \text{PUS}.\mathsf{Sign}(msk, id)$, output $sk_{id} = \sigma$.
- $\mathsf{Encrypt}(mpk, id, m)$: on input $mpk = vk$ and id and a message m, output an obfuscated circuit $c \leftarrow i\mathcal{O}(\mathsf{Enc}_{mpk,id,m})$. The circuit $\mathsf{Enc}_{mpk,id,m}$ is depicted in Figure 7.
- $\mathsf{Decrypt}(sk_{id}, c)$: on input sk_{id} and a ciphertext c, output $m \leftarrow c(sk_{id})$.

$\mathsf{Enc}_{mpk,id,m}$

Constants: mpk, id, m

Input: sk

1. If $\text{PUS}.\mathsf{Verify}(mpk, id, sk) = 1$, then output m; else output $\perp$.

Figure 7: $\mathsf{Enc}_{mpk,id,m}$ takes as input sk, and has constants mpk and id and m hardwired. The size of this circuit is padded to be the maximum of itself and $\mathsf{Sim}_{mpk,id,f}$ as described in Figure 8.

The correctness of the above construction is straightforward. For security, we have the following theorem.

Theorem 6.1. *If $i\mathcal{O}$ is secure and PUS is unforgeable, then the above IBE is a bounded KDM-secure in the single-key setting.*

Proof. We prove the above theorem via a sequence of games.

$\mathsf{Sim}_{mpk,id,f}$

Constants: mpk, id, f

Input: sk

1. If PUS.$\mathsf{Verify}(mpk, id, sk) = 1$, then output $f(sk)$; else output $\perp$.

Figure 8: $\mathsf{Sim}_{mpk,id,f}$ takes as input sk, and has constants mpk and id and f hardwired.

Game 0: This game corresponds to the KDM security game that for each encryption query $\langle f \rangle$ to $\mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot)$, $\mathcal{CH}$ responds with $\mathcal{O}^{0}_{\mathsf{enc}}(f)$, i.e., the real KDM encryption $i\mathcal{O}(\mathsf{Enc}_{mpk,id^*,f(sk_{id^*})}) \leftarrow \mathsf{Encrypt}(mpk, id^*, f(sk(id^*)))$.

Game 1: This game corresponds to the KDM security game that for each encryption query $\langle f \rangle$ to $\mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot)$, $\mathcal{CH}$ responds with $\mathcal{O}_{\mathsf{sim}}(f) = i\mathcal{O}(\mathsf{Sim}_{mpk,id^*,f})$.

Game 2: This game corresponds to the KDM security game that for each encryption query $\langle f \rangle$ to $\mathcal{O}^{\beta}_{\mathsf{enc}}(\cdot)$, $\mathcal{CH}$ responds with $\mathcal{O}^{1}_{\mathsf{enc}}(f)$, i.e., the zero encryption $i\mathcal{O}(\mathsf{Enc}_{mpk,id^*,0^{|m|}}) \leftarrow \mathsf{Encrypt}(mpk, id^*, 0^{|m|})$.

To establish the desired KDM security, it suffices to show that Game 0 and Game 2 are computationally indistinguishable. To this end, we show that both Game 0 and Game 2 are computationally indistinguishable from the intermediate Game 1. We assume the maximum number of encryption queries made by the adversary is upper bounded by a polynomial q in λ.

Lemma 6.2. *Game 0 and Game 1 are computationally indistinguishable, given the security of $i\mathcal{O}$.*

Proof. We introduce $q+1$ hybrids indexed by $0 \leq i \leq q$ between Game 0 and Game 1, where in Hyb_i the first i encryption queries are answered with $\mathcal{O}_{\mathsf{sim}}(\cdot)$ and the rest encryption queries are answered with $\mathcal{O}^{0}_{\mathsf{enc}}(\cdot)$. By definition, Hyb_0 is exactly Game 0 and Hyb_q is exactly Game 1. Since $q = \mathsf{poly}(\lambda)$, it suffices to show that for each $1 \leq i \leq q$ we have $\text{Hyb}_{i-1} \approx_c \text{Hyb}_i$. Note that these two successive hybrids only differ at the response to the i-th encryption query $\langle f_i \rangle$, thus the crux of the proof is to show $\mathcal{O}^{0}_{\mathsf{enc}}(f_i) \approx_c \mathcal{O}_{\mathsf{sim}}(f_i)$, i.e., $i\mathcal{O}(\mathsf{Enc}_{mpk,id^*,f_i(sk_{id^*})}) \approx_c i\mathcal{O}(\mathsf{Sim}_{mpk,id^*,f_i})$. Next, we formally prove the above intuition by giving a reduction to the security of $i\mathcal{O}$.

Suppose there is an adversary $\mathcal{A}$ that distinguishes Hyb_{i-1} and Hyb_i with non-negligible probability, we show how to build an algorithm $\mathcal{B} = (\mathcal{S}, \mathcal{D})$ breaks the security of $i\mathcal{O}$.

$\mathcal{S}(\lambda)$ behaves as follows: It runs $(vk, sk) \leftarrow$ PUS.$\mathsf{KeyGen}(\lambda)$, sends $mpk = vk$ to $\mathcal{A}$. In Phase 1, when $\mathcal{A}$ makes extraction queries $\langle id \rangle$, $\mathcal{S}$ responds with $sk_{id} \leftarrow$ PUS.$\mathsf{Sign}(msk, id)$. In the challenge phase, $\mathcal{A}$ chooses id^* as the target identity. $\mathcal{S}$ computes $sk_{id^*} \leftarrow$ PUS.$\mathsf{Sign}(msk, id^*)$. In Phase 2, when $\mathcal{A}$ makes extraction queries $\langle id \rangle$ with $id \neq id^*$, $\mathcal{S}$ responds the same way as in Phase 1. When $\mathcal{A}$ makes the first $i-1$ encryption queries, $\mathcal{S}$ responds with $\mathcal{O}_{\mathsf{sim}}(\cdot)$. When $\mathcal{A}$ makes the i-th encryption query $\langle f_i \rangle$, $\mathcal{S}$ sets $aux = (mpk, msk, id^*)$, then builds C_0 as the circuit $\mathsf{Enc}_{mpk,id^*,f_i(sk_{id^*})}$, and C_1 as the circuit $\mathsf{Sim}_{mpk,id^*,f_i}$.

Before describing $\mathcal{D}$, we observe that C_0 and C_1 behaves identically. It is easy to see that both C_0 and C_1 output $f_i(sk_{id^*})$ on the single input sk_{id^*} (this is guaranteed by the unique property of PUS) and $\perp$ elsewhere. Thus, $\mathcal{S}$ satisfies the conditions needed for invoking the indistinguishability property of $i\mathcal{O}$.

Now, we can describe the algorithm $\mathcal{D}$. Given aux and $i\mathcal{O}(C_b)$ as challenge, $\mathcal{D}$ continues to interact with $\mathcal{A}$ with the aim to determine b. When $\mathcal{A}$ makes extraction queries $\langle id \rangle$ where

$id \neq id^*$, $\mathcal{D}$ responds with $sk_{id} \leftarrow \text{PUS}.\mathsf{Sign}(msk, id)$. When $\mathcal{A}$ makes the i-th encryption query, $\mathcal{D}$ answers with $i\mathcal{O}(C_b)$. When $\mathcal{A}$ makes the rest encryption queries, $\mathcal{D}$ responds with $\mathcal{O}^0_{\mathsf{enc}}(\cdot)$.

By construction, if $\mathcal{B}$ receives $i\mathcal{O}(C_0)$ (resp. $i\mathcal{O}(C_1)$), then $\mathcal{A}$'s view is identical to that in Hyb_{i-1} (resp. Hyb_i). Thereby, we have $\text{Hyb}_{i-1} \approx_c \text{Hyb}_i$ for each $1 \leq i \leq q$ based on the security of $i\mathcal{O}$. By the definitions of the hybrids and the fact that $q = \mathsf{poly}(\lambda)$, we have Game 0 $\approx_c$ Game 1. The lemma follows. $\square$

Lemma 6.3. *Game 1 and Game 2 are computationally indistinguishable, given the security of $i\mathcal{O}$ and the security of puncturable unique signature.*

Proof. We introduce $q+1$ hybrids indexed by $0 \leq i \leq q$ between Game 1 and Game 2, where in Hyb_i the first i encryption queries are answered with $\mathcal{O}^1_{\mathsf{enc}}(\cdot)$ and the rest encryption queries are answered with $\mathcal{O}_{\mathsf{sim}}(\cdot)$. By definition, Hyb_0 is exactly Game 1 and Hyb_q is exactly Game 2. Since $q = \mathsf{poly}(\lambda)$, it suffices to show that for each $1 \leq i \leq q$ we have $\text{Hyb}_{i-1} \approx_c \text{Hyb}_i$. Note that these two successive hybrids only differ at the response to the i-th encryption query $\langle f_i \rangle$, thus the crux of the proof is to show $\mathcal{O}_{\mathsf{sim}}(f_i) \approx_c \mathcal{O}^1_{\mathsf{enc}}(f_i)$, i.e., $i\mathcal{O}(\mathsf{Sim}_{mpk,id^*,f_i}) \approx_c i\mathcal{O}(\mathsf{Sim}_{mpk,id^*,0^{|m|}})$. Next, we formally prove the above intuition by giving a reduction to the security of $i\mathcal{O}$ and the security of puncturable unique signature.

Suppose there is an adversary $\mathcal{A}$ that distinguishes Hyb_{i-1} and Hyb_i with non-negligible probability, we show how to build an algorithm $\mathcal{B} = (\mathcal{S}, \mathcal{D})$ breaks the security of $i\mathcal{O}$.

$\mathcal{S}(\lambda)$ behaves as follows: It invokes a PUS challenger and receives a verification vk, then simulates $\mathcal{A}$'s challenger by sending him $mpk = vk$. In Phase 1, when $\mathcal{A}$ makes extraction queries $\langle id \rangle$, $\mathcal{S}$ submits signing queries $\langle id \rangle$ to its PUS challenger and forwards the reply to $\mathcal{A}$. In the challenge phase, $\mathcal{A}$ chooses id^* as the target identity. $\mathcal{S}$ submits id^* to its PUS challenger and receives back a punctured signing key $sk(\{id^*\}) \leftarrow \text{PUS}.\mathsf{Puncture}(sk, id^*)$. In Phase 2, when $\mathcal{A}$ makes extraction queries $\langle id \rangle$ with $id \neq id^*$, $\mathcal{S}$ responds the same way as in Phase 1. When $\mathcal{A}$ makes the first $i-1$ encryption queries, $\mathcal{S}$ responds with $\mathcal{O}^1_{\mathsf{enc}}(\cdot)$. When $\mathcal{A}$ makes the i-th encryption query $\langle f_i \rangle$, $\mathcal{S}$ sets $aux = (mpk, sk(\{id^*\}), id^*)$, then builds C_1 as the circuit $\mathsf{Sim}_{mpk,id^*,f_i}$, and C_2 as the circuit $\mathsf{Enc}_{mpk,id^*,0^{|m|}}$.

Before describing $\mathcal{D}$, we observe that the circuits C_1 and C_2 have at most one differing-input. To see this, note that C_1 outputs $f_i(sk_{id^*})$ and C_2 outputs $0^{|m|}$ on the single input sk_{id^*} (this is guaranteed by the unique property of PUS), and the two circuits output $\perp$ elsewhere. For the case $f_i(sk_{id^*}) = 0^{|m|}$, C_0 and C_1 are functionally equivalent. For the case $f_i(sk_{id^*}) \neq 0^{|m|}$, it remains to show that no PPT adversary is able to find the only differing-input. Observe that the only differing-input sk_{id^*} is exactly the unique signature on id^*, a reduction to the security of PUS is immediate: suppose given (C_1, C_2, aux) there exists a PPT adversary $\mathcal{F}$ can find such differing-input, say sk_{id^*}, of C_1 and C_2 with non-negligible probability, then $\mathcal{S}$ breaks the security of PUS with the same probability.

Now, we can describe the algorithm $\mathcal{D}$. Given aux and $i\mathcal{O}(C_b)$ as challenge, $\mathcal{D}$ continues to interact with $\mathcal{A}$ with the aim to determine b. When $\mathcal{A}$ makes extraction queries $\langle id \rangle$ where $id \neq id^*$, $\mathcal{D}$ responds with $sk_{id} \leftarrow \text{PUS}.\mathsf{Sign}(sk(\{id^*\}), id)$. When $\mathcal{A}$ makes the i-th encryption queries, $\mathcal{D}$ responds with $i\mathcal{O}(C_b)$. When $\mathcal{A}$ makes the rest encryption queries, $\mathcal{A}$ responds with $\mathcal{O}_{\mathsf{sim}}(\cdot)$.

By construction, if $\mathcal{B}$ receives $i\mathcal{O}(C_1)$ (resp. $i\mathcal{O}(C_2)$), then $\mathcal{A}$'s view is identical to that in Hyb_{i-1} (resp. Hyb_i). Thereby, we have $\text{Hyb}_{i-1} \approx_c \text{Hyb}_i$ for each $1 \leq i \leq q$ based on the security of $i\mathcal{O}$. By the definitions of the hybrids and the fact that $q = \mathsf{poly}(\lambda)$, we have Game1 $\approx_c$ Game2. The lemma follows. $\square$

The theorem follows from Lemma 1 and Lemma 2. $\square$

Remark 6.1. In the proof of Lemma 2, we actually need to use $di\mathcal{O}$. Nevertheless, the two circuits C_1 and C_2 have at most one differing-input. Thereby, according to Lemma 3.1 we could safely use $i\mathcal{O}$ rather than resort to $di\mathcal{O}$.

Currently, we do not know how to extend the above construction to the multiple-keys setting. The challenge is that the circuit $\mathsf{Sim}_{mpk,id,f}$ is given as input one of the secret keys but now has to output a function of (possibly) n secret keys. In the PKE setting, Marcedone et al. [MPS16] solved this problem by embedding a special relationship among secret keys into $\mathsf{Sim}_{mpk,id,f}$. However, their approach seems not work here, because in IBE secret keys are derived from identities and thus it is hard to manipulate the relationship among them. We left the extension to the multiple-keys setting as an interesting problem.

7 Counterexample for n-Circular Security from Differing-Input Obfuscation and Puncturable IBE

Other than constructing IBE schemes for which we can prove KDM security, one may ask the more fundamental question of "if standard security notions already imply KDM security". A series of recent works [ABBC10, CGH12, MO14, KRW15, BHW15, KW16, AP16] give negative answer to this question in the public-key setting.

In this section, we try to make some progress toward the truth of this question in the identity-based setting. Our goal is to figure out whether n-circular security (which is a special case of KDM security) is implied by IND-CPA/CCA security.

Similar to the public-key setting [BHW15], such a counterexample is easy to construct for the case $n = 1$. Concretely, start from a CPA secure IBE scheme $\Pi =$(Setup, $\mathsf{Extract}$, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$) which admits efficient $\mathsf{CheckSK}$ algorithm (c.f. Definition 3.5), one can modify it to a new IBE scheme $\Pi' = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}', \mathsf{Decrypt}')$, where the algorithms Setup and $\mathsf{Extract}$ are same as that of Π; $\mathsf{Encrypt}'(mpk, id, m)$ outputs $\mathsf{Encrypt}(mpk, id, m)||0$ if $m \neq sk_{id}$ and $m||1$ otherwise (this could be done with the help of $\mathsf{CheckSK}$); $\mathsf{Decrypt}'(sk_{id}, c||b)$ outputs $\mathsf{Decrypt}(sk_{id}, c)$ if $b = 0$ and c otherwise. Clearly, Π' is correct and inherits IND-CPA security from that of Π, but it is completely 1-circular insecure. The strategy behind this counterexample is "check-then-mark", that is, the encryption algorithm first checks if the encrypted message is a valid secret key, then encrypts in two distinguished manners (e.g., by attaching a bit mark) according to the check result.

From the proceeding discussion in introduction, while it can be easily shown that IND-CPA security does not imply 1-circular security, the case for $n \geq 2$ turns out to be much challenging. When $n \geq 2$ it seems difficult to implement the "check-then-mark" strategy since the circle is specified by the adversary "on the fly". To circumvent this difficulty, we embed an obfuscation of a circuit to the ciphertext with the hope that the circuit admits dynamic cycle detection without compromising IND-CPA security. As we sketched earlier, we need a new notion called puncturable IBE as the basis of our counterexample. In what follows, we first formally introduce puncturable IBE and show how to construct it.

7.1 Puncturable IBE

Definition 7.1 (Punctureable IBE)**.** A puncturable IBE (PIBE) scheme is an IBE scheme whose master secret key allows efficient puncturing (analogous to puncturable PRF). The syntax of puncturable IBE is identical to standard IBE except it equips two additional polynomial algorithms as follows:

- $\mathsf{Puncture}(msk, id)$: on input msk and an identity $id^* \in I$, output a punctured master secret key $msk(\{id^*\})$.

- $\mathsf{Derive}(msk(\{id^*\}), id)$: on input $msk(\{id^*\})$ and an identity $id \in I$, output a secret key sk_{id} for id if $id \neq id^*$ and $\perp$ otherwise. We require that for all $id \neq id^*$, the outputs of $\mathsf{Extract}(msk, id)$ and $\mathsf{Derive}(msk(\{id^*\}), id)$ have the same distribution.

Intuitively, the two algorithms ensure that there is a succinct description of the set of secret keys for all identities but one.

Security. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against puncturable IBE and define its advantage in the following experiment:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\beta = \beta' : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\lambda); \\ (state, id^*, m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{ext}}(\cdot)}(mpk); \\ msk(\{id^*\}) \leftarrow \mathsf{Puncture}(msk, id^*); \\ \beta \xleftarrow{\text{R}} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(mpk, id^*, m_\beta); \\ \beta' \leftarrow \mathcal{A}_2(state, msk(\{id^*\}), c^*); \end{array}\right] - \frac{1}{2},$$

where $\mathcal{O}_{\mathsf{ext}}(\cdot)$ is an oracle that on input $id \in I$ returns $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$, and $\mathcal{A}_1$ is not allowed to choose the identity that had been queried for secret keys as the target one. A PIBE is said to be IND-CPA secure if for any PPT adversary $\mathcal{A}$, its advantage defined as above is negligible in λ.

We then proceed to show the existence of PIBE.

PIBE from Hierarchical IBE. Let HIBE be an ℓ-level HIBE with identity space $(\{0,1\}^*)^\ell$, we can build a PIBE with identity space $\{0,1\}^\ell$ as follows.

- $\mathsf{Setup}(\lambda)$: output $(mpk, msk) \leftarrow \text{HIBE}.\mathsf{Setup}(\lambda, \ell)$.
- $\mathsf{Extract}(msk, id)$: on input msk and an identity $id \in \{0,1\}^\ell$, map id to depth ℓ ID-vector $v = (id_{[1]}, \dots, id_{[\ell]})$ where $id_{[i]}$ denotes the i-th bit of id, then compute $sk_v \leftarrow \text{HIBE}.\mathsf{Extract}(msk, v)$, output $sk_{id} = sk_v$.
- $\mathsf{Puncture}(msk, id^*)$: on input msk and an identity $id^* \in \{0,1\}^\ell$: for $1 \leq i \leq \ell$, set depth i ID-vector $v_i = (id^*_{[1]}, \dots, id^*_{[i-1]}, \overline{id^*_{[i]}})$, then compute $sk_{v_i} \leftarrow \text{HIBE}.\mathsf{Extract}(msk, v_i)$, output $msk(\{id^*\}) = (sk_{v_1}, \dots, sk_{v_\ell})$. It is easy to verify that the size of $msk(\{id^*\})$ is polynomial in λ.
- $\mathsf{Derive}(msk(\{id^*\}), id)$: on input $msk(\{id^*\}) = (sk_{v_1}, \dots, sk_{v_\ell})$ and an identity $id \in \{0,1\}^\ell$, if $id \neq id^*$, find v_j that is a prefix of id^* and output $sk_{id} \leftarrow \text{HIBE}.\mathsf{Derive}(sk_{v_j}, id)$; if $id = id^*$, output $\perp$.
- $\mathsf{Encrypt}(mpk, id, m)$: on input mpk and an identity id and a message m, map id to depth ℓ ID-vector $v = (id_{[1]}, \dots, id_{[\ell]})$, output ciphertext $c \leftarrow \text{HIBE}.\mathsf{Encrypt}(mpk, v, m)$.
- $\mathsf{Decrypt}(sk_{id}, c)$: on input sk_{id} and a ciphertext c, interpret sk_{id} as sk_v, output $m \leftarrow \text{HIBE}.\mathsf{Decrypt}(sk_v, c)$.

The correctness, checkable property and IND-CPA security of the PIBE follows readily from that of the underline HIBE. We omit the details here.

Remark 7.1. For our main purpose, we simply define puncture IBE with respect to a singleton $\{id\}$. It can be easily generalized to a polynomial-size identity set $T \subset I$. In addition, we only demonstrate the existence of PIBE by giving a direct construction from HIBE. We remark that PIBE can also be neatly derived from Binary Tree Encryption (BTE) [CHK03], which is arguably a more simple and general notion than HIBE.

7.2 Construction of the Counterexample for n-circular security

CONSTRUCTION OF COUNTEREXAMPLE. Let PIBE be a puncturable IBE scheme with efficient $\mathsf{CheckSK}$ algorithm, $di\mathcal{O}$ be a differing-inputs obfuscator. For simplicity, we also assume $SK \subseteq M$ in PIBE. We construct an IBE scheme with the same identity space, message space, secret key space as the starting PIBE:

- The Setup, $\mathsf{Extract}$ and $\mathsf{CheckSK}$ algorithms are the same as that of PIBE.
- $\mathsf{Encrypt}(mpk, id, m)$: on input mpk and an identity $id \in I$ and a message m, first compute $c_e \leftarrow \text{PIBE}.\mathsf{Encrypt}(mpk, id, m)$, then create an obfuscated circuit $c_t \leftarrow di\mathcal{O}(\lambda, \mathsf{CycleTest}_{id,m})$, output the final ciphertext $c = (c_t, c_e)$. The circuit $\mathsf{CycleTest}_{id,m}$ is depicted in Figure 9.
- $\mathsf{Decrypt}(sk_{id}, c)$: on input sk_{id} and $c = (c_e, c_t)$, output $m \leftarrow \text{PIBE}.\mathsf{Decrypt}(sk_{id}, c_e)$.

CycleTest

Constants: id, m.

Input: $\mathbf{id} = (id_1, \ldots id_n)$ and $\mathbf{c}_e = (c_{1,e}, \ldots, c_{n,e})$.

1. If $id \neq id_1$, output $\perp$.
2. Assign $sk_2 := m$.
3. For $i = 2$ to n, do:
   (a) If $\mathsf{CheckSK}(sk_i, id_i) = 0$, output $\perp$.
   (b) Else, compute $sk_{(i \bmod n)+1} \leftarrow \text{PIBE}.\mathsf{Decrypt}(sk_i, c_{i,e})$, and output $\perp$ if PIBE.Decrypt fails.
4. If $\mathsf{CheckSK}(sk_1, id_1) = 0$, output $\perp$; else output sk_1.

Figure 9: CycleTest takes as input $\mathbf{id} = (id_1, \ldots, id_n)$ and $\mathbf{c}_e = (c_{1,e}, \ldots, c_{n,e})$, and has constants id and m hardwired. The size of this circuit is padded to be the maximum of itself and CycleReject as described in Figure 10.

CycleReject

Constants: none

Input: $\mathbf{id} = (id_1, \ldots, id_n)$ and $\mathbf{c}_e = (c_{1,e}, \ldots, c_{n,e})$.

1. Output $\perp$.

Figure 10: CycleReject takes as input $\mathbf{id} = (id_1, \ldots, id_n)$ and $\mathbf{c}_e = (c_{1,e}, \ldots, c_{n,e})$, and has no constant hardwired.

The correctness of the above construction follows from that of the starting PIBE. We then prove it is still IND-CPA secure.

Theorem 7.1. *If PIBE is an IND-CPA secure puncturable IBE and $di\mathcal{O}$ is a secure differing-input obfuscator, then the above construction is IND-CPA secure.*

Proof. We prove this theorem via a sequence of games.

Game 0 (the original game):

1. $\mathcal{CH}$ runs $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$, then sends mpk to $\mathcal{A}$.

2. On extraction query $\langle id \rangle$, $\mathcal{CH}$ responds with $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$.
3. $\mathcal{A}$ submits (id^*, m_0, m_1). $\mathcal{CH}$ picks a random bit β, runs $\mathsf{Encrypt}(mpk, id^*, m_\beta)$, i.e., computes $c_e^* \leftarrow \text{PIBE}.\mathsf{Encrypt}(mpk, id^*, m_\beta)$, $c_t^* \leftarrow di\mathcal{O}(\lambda, \mathsf{CycleTest}_{id^*,m_\beta})$. $\mathcal{CH}$ sets $c^* = (c_e^*, c_t^*)$ and sends c^* to $\mathcal{A}$.
4. On extraction query $\langle id \rangle$ that $id \neq id^*$, $\mathcal{CH}$ responds the same way as in Phase 1.
5. Finally, $\mathcal{A}$ outputs a guess β' for β and wins if $\beta' = \beta$.

Game 1 (replace $\mathsf{CycleTest}_{id^*,m_\beta}$ with $\mathsf{CycleReject}$):

3. $\mathcal{CH}$ picks a random bit β, runs $c_e^* \leftarrow \text{PIBE}.\mathsf{Encrypt}(mpk, id^*, m_\beta)$, $\underline{c_t^* \leftarrow di\mathcal{O}(\lambda, \mathsf{CycleReject})}$. $\mathcal{CH}$ sets $c^* = (c_e^*, c_t^*)$ and sends c^* to $\mathcal{A}$.

Lemma 7.2. *The advantages of any PPT adversary in Game 0 and Game 1 are negligibly close, given the security of the $di\mathcal{O}$ and the security of PIBE.*

Proof. Suppose there is a PPT adversary $\mathcal{A}$ whose advantages in Game 0 and Game 1 are not negligibly close, then we can build an algorithm $\mathcal{B} = (\mathcal{S}, \mathcal{D})$ breaks the assumed security of $di\mathcal{O}$ by interacting with $\mathcal{A}$ as follows.

$\mathcal{S}(\lambda)$ behaves as follows: It invokes a PIBE challenger and receives mpk, where $(mpk, msk) \leftarrow \text{PIBE}.\mathsf{Setup}(\lambda)$. It then begin to simulates $\mathcal{A}$'s challenger by sending him mpk. In Phase 1, when $\mathcal{A}$ makes extraction queries $\langle id \rangle$, $\mathcal{S}$ forwards them to its own PIBE challenger and sends the reply back. In the challenge phase, upon receiving (id^*, m_0, m_1) from $\mathcal{A}$, $\mathcal{S}$ submits (id^*, m_0, m_1) to its PIBE challenger, and receives back a punctured master secret key $msk(\{id^*\})$ and a ciphertext $c_e^* \leftarrow \Pi.\mathsf{Encrypt}(mpk, id^*, m_\gamma)$ (where γ is unknown to $\mathcal{S}$). $\mathcal{S}$ then picks a random bit $\beta \in \{0, 1\}$, sets $aux = (mpk, msk(\{id^*\}), id^*, m_\beta, \beta)$, builds $C_0 = \mathsf{CycleTest}_{id^*,m_\beta}$ and $C_1 = \mathsf{CycleReject}$.

Before describing $\mathcal{D}$, we have to show that $\mathcal{S}$ satisfies the conditions needed for invoking the indistinguishability property of $di\mathcal{O}$, i.e., given (C_0, C_1, aux) no PPT adversary can find a differing input of C_0 and C_1 with non-negligible probability. Observe that C_0 outputs sk_{id^*} on some inputs and $\perp$ on the rest inputs, whereas C_1 always outputs $\perp$. A reduction to the security of PIBE is immediate: suppose given (C_0, C_1, aux) there exists an adversary $\mathcal{F}$ that can find a differing-input, say x, of C_0 and C_1 with non-negligible probability, then $\mathcal{S}$ obtains a valid secret key sk_{id^*} for id^* with the same probability by simply computing $C_0(x)$ and thus totally breaks the assumed IND-CPA security of PIBE (always guess the right γ).[11]

Now, we can describe the algorithm $\mathcal{D}$. Given $di\mathcal{O}(C_b)$ and auxiliary information $aux = (mpk, msk(\{id^*\}), id^*, m_\beta, \beta)$ as challenge, $\mathcal{D}$ continues to interact with $\mathcal{A}$ with the aim to determine b. To prepare the challenge ciphertext, $\mathcal{D}$ computes $c_e^* \leftarrow \Pi.\mathsf{Encrypt}(mpk, id^*, m_\beta)$, sets $c_t^* \leftarrow di\mathcal{O}(C_b)$, and sends $c^* = (c_e^*, c_t^*)$ to $\mathcal{A}$. When $\mathcal{A}$ makes extraction query $\langle id \rangle$ with $id \neq id^*$, $\mathcal{D}$ responds with $sk_{id} \leftarrow \text{PIBE}.\mathsf{Derive}(msk(\{id^*\}), id)$. Finally, $\mathcal{A}$ outputs a guess β' for β. If $\mathcal{A}$ wins, $\mathcal{D}$ outputs 1.

By construction, if $\mathcal{D}$ receives $di\mathcal{O}(C_0)$ (resp. $di\mathcal{O}(C_1)$), the probability that $\mathcal{D}$ outputs 1 is exactly the probability of $\mathcal{A}$ winning in Game 0 (resp. Game 1).

The lemma follows. □

Lemma 7.3. *No PPT adversary has non-negligible advantage in Game 1, given the starting* PIBE *is IND-CPA secure.*

Proof. Suppose there is an adversary $\mathcal{A}$ that wins in Game 1 with some non-negligible advantage, we show how to build an algorithm $\mathcal{B}$ breaks the IND-CPA security of PIBE with the same advantage. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

[11] A subtlety here is we have to require Π to satisfy perfect correctness, i.e., valid secret keys always decrypt correctly. Most known IBE schemes based on number-theoretic assumptions meet this requirement.

1. Given mpk where $(mpk, msk) \leftarrow \text{PIBE}.\mathsf{Setup}(\lambda)$, $\mathcal{B}$ sends mpk to $\mathcal{A}$.
2. On extraction query $\langle id \rangle$, $\mathcal{B}$ forwards the query to its own challenger and sends the reply to $\mathcal{A}$.
3. Upon receiving (id^*, m_0, m_1) from $\mathcal{A}$, $\mathcal{B}$ submits (id^*, m_0, m_1) to its own challenger. After receiving back a punctured master secret key $msk(\{id^*\})$ and challenge ciphertext $c_e^* \leftarrow \text{PIBE}.\mathsf{Encrypt}(mpk, id^*, m_\beta)$ for some unknown bit β, $\mathcal{B}$ computes $c_t^* \leftarrow di\mathcal{O}(\lambda, \mathsf{CycleReject})$, and sends $c^* = (c_e^*, c_t^*)$ to $\mathcal{A}$.
4. On extraction query $\langle id \rangle$ that $id \neq id^*$, $\mathcal{B}$ responds with punctured master secret key $msk(\{id^*\})$, i.e., $sk_{id} \leftarrow \text{PIBE}.\mathsf{Derive}(msk(\{id^*\}), id)$.
5. Finally, $\mathcal{A}$ outputs a guess β' for β. $\mathcal{B}$ forwards β' to its own challenger.

It is easy to check that $\mathcal{B}$ simulates Game 1 perfectly. Therefore, if $\mathcal{A}$ wins in Game 1 with some non-negligible advantage, $\mathcal{B}$ breaks the assumed IND-CPA security of PIBE with the same advantage. The lemma follows. □

Combining all these above, the theorem immediately follows. □

We then show the above construction is n-circular insecure.

Proposition 7.4. *The above construction is n-circular insecure.*

Proof. We construct a PPT algorithm Test that breaks the n-circular security of the above construction as follows. After receiving mpk from the challenger, Test randomly picks n identities $\mathbf{id} = (id_1, \ldots, id_n)$ and submits them to the challenger, and receives back $\mathbf{c} = (c_1, \ldots, c_n)$. To decide whether $\mathbf{c}$ is a circle encryption or a zero encryption, Test first parses $c_i = (c_{i,e}, c_{i,t})$. By definition, $c_{i,t}$ is $di\mathcal{O}(\lambda, \mathsf{CycleTest}_{id_i,m})$, where m is either $sk_{(i \bmod n)+1}$ or $0^{|m|}$. Test then sets $\mathbf{c}_e = (c_{1,e}, \ldots, c_{n,e})$ and runs $c_{1,t}(\mathbf{id}, \mathbf{c}_e)$, and outputs 0 if the result of is $\perp$ and 1 otherwise. If $\mathbf{c}$ is a cycle encryption w.r.t. $\mathbf{id}$, the output of $c_{1,t}(\mathbf{id}, \mathbf{c}_e)$ is 1. If $\mathbf{c}$ is a zero encryption, the output of $c_{1,t}(\mathbf{id}, \mathbf{c}_e)$ must be $\perp$ with overwhelming probability. Otherwise, this means that Test algorithm finds n identities whose secret keys are all zero strings with non-negligible probability, which contradicts to the assumed IND-CPA security of PIBE.

Clearly, Test is a PPT algorithm and wins the n-circular security game with advantage negligibly close to 1/2. The desired result follows. □

7.3 Separation from IND-CCA security

The above counterexample shows that in the IBE setting IND-CPA security does not necessarily imply n-circular security. It is interesting to know if stronger notions, say IND-CCA security, imply n-circular security.

Toward this question, we extend the framework [BHW15] of building counterexamples for circular security to the IBE setting in Appendix A, which might be of independent interest. In this framework, a so called n-cycle tester plays a crucial role. More precisely, an IND-CPA (resp. IND-CCA) secure IBE scheme in combination with a compatible IND-CPA secure n-cycle tester instantly imply a new IBE scheme which is IND-CPA (resp. IND-CCA) secure but n-circular insecure. Note that our counterexample described above can certainly serve as an IND-CPA n-cycle tester, thereby a counterexample of n-circular security from IND-CCA security follows immediately via this framework by coupling with an IND-CCA secure IBE scheme.

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, 2010.

[ABBC10] Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. Cryptographic agility and its relation to circular encryption. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 403–422. Springer, 2010.

[ABG+13] Prabhanjan Ananth, Dan Boneh, Sanjam Garg, Amit Sahai, and Mark Zhandry. Differing-inputs obfuscation and applications. IACR Cryptology ePrint Archive, Report 2013/689, 2013. http://eprint.iacr.org/2013/689.

[ABHS05] Pedro Adão, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In *Computer Security - ESORICS 2005*, volume 3679 of *LNCS*, pages 374–396. Springer, 2005.

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

[ADN+10] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 113–134. Springer, 2010.

[AP12] Jacob Alperin-Sheriff and Chris Peikert. Circular and KDM security for identity-based encryption. In *Public Key Cryptography - PKC 2012*, volume 7293 of *LNCS*, pages 334–352. Springer, 2012.

[AP16] Navid Alamati and Chris Peikert. Three's compromised too: Circular insecurity for any cycle length from (ring-)lwe. 2016.

[App11] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 527–546. Springer, 2011.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[BCC+14] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. The hunting of the SNARK. *IACR Cryptology ePrint Archive*, page 580, 2014.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Innovations in Theoretical Computer Science 2012*, pages 326–349. ACM, 2012.

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014*, volume 8349 of *LNCS*, pages 52–73. Springer, 2014.

[BDH14] Florian Böhl, Gareth T. Davies, and Dennis Hofheinz. Encryption schemes secure under related-key and key-dependent message attacks. In *Public-Key Cryptography - PKC 2014*, volume 8383 of *Lecture Notes in Computer Science*, pages 483–500. Springer, 2014.

[BDU08] Michael Backes, Markus Dürmuth, and Dominique Unruh. OAEP is secure under key-dependent messages. In *Advances in Cryptology - ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 506–523. Springer, 2008.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM Journal on Computation*, 32:586–615, 2003.

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

[BGH07] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007*, pages 647–657. IEEE Computer Society, 2007.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *17th International Conference on Practice and Theory in Public-Key Cryptography, PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, 2014.

[BGK11] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*, volume 6597 of *LNCS*, pages 201–218. Springer, 2011.

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

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, 2008.

[BHW15] Allison Bishop, Susan Hohenberger, and Brent Waters. New circular security counterexamples from decision linear and learning with errors. In *Advances in Cryptology - ASIACRYPT 2015*, volume 9453 of *LNCS*, pages 776–800. Springer, 2015.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, 2002.

[BST14] Mihir Bellare, Igors Stepanovs, and Stefano Tessaro. Poly-many hardcore bits for any one-way function and a framework for differing-inputs obfuscation. In *Advances in Cryptology - ASIACRYPT 2014*, volume 8874 of *LNCS*, pages 102–121. Springer, 2014.

[BSW16] Mihir Bellare, Igors Stepanovs, and Brent Waters. New negative results on differing-inputs obfuscation. In *Advances in Cryptology - EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 792–821. Springer, 2016.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In *Advances in Cryptology - ASIACRYPT 2013*, volume 8270 of *LNCS*, pages 280–300. Springer, 2013.

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

[CDRW10] Sherman S. M. Chow, Yevgeniy Dodis, Yannis Rouselakis, and Brent Waters. Practical leakage-resilient identity-based encryption from simple assumptions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010*, pages 152–161. ACM, 2010.

[CGH12] David Cash, Matthew Green, and Susan Hohenberger. New definitions and separations for circular security. In *Public Key Cryptography - PKC 2012*, volume 7293 of *LNCS*, pages 540–557. Springer, 2012.

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

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology - EUROCRYPT 2001*, pages 93–118. Springer, 2001.

[Cor09] Jean-Sébastien Coron. A variant of Boneh-Franklin IBE with a tight reduction in the random oracle model. *Des. Codes Cryptography*, 50(1):115–133, 2009.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[Fuc14] Georg Fuchsbauer. Constrained verifiable random functions. In *Security and Cryptography for Networks - 9th International Conference, SCN 2014*, volume 8642 of *LNCS*, pages 95–114. Springer, 2014.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464. Springer, 2006.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 169–178. ACM, 2009.

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

[GGHW14] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In *Advances in Cryptology - CRYPTO 2014*, volume 8616 of *LNCS*, pages 518–535. Springer, 2014.

[GHV12] David Galindo, Javier Herranz, and Jorge L. Villar. Identity-based encryption with master key-dependent message security and leakage-resilience. In *17th European Symposium on Research in Computer Security, ESORICS 2012*, volume 7459, pages 627–642. Springer, 2012.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GO92] Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In *Advances in Cryptology - CRYPTO 1992*, volume 740 of *LNCS*, pages 228–245. Springer, 1992.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*, pages 197–206. ACM, 2008.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology - CRYPTO 2013*, volume 8042 of *LNCS*, pages 75–92. Springer, 2013.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011*, pages 99–108. ACM, 2011.

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *LNCS*, pages 202–219. Springer, 2009.

[HKS16] Mohammad Hajiabadi, Bruce M. Kapron, and Venkatesh Srinivasan. On generic constructions of circularly-secure, leakage-resilient public-key encryption schemes. In *Public-Key Cryptography - PKC 2016*, volume 9615 of *Lecture Notes in Computer Science*, pages 129–158. Springer, 2016.

[HLL16] Shuai Han, Shengli Liu, and Lin Lyu. Efficient kdm-cca secure public-key encryption for polynomial functions. In *Advances in Cryptology - ASIACRYPT 2016*, LNCS. Springer, 2016.

[Hof13] Dennis Hofheinz. Advances in cryptology - EUROCRYPT 2013. In *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 520–536. Springer, 2013.

[IPS15] Yuval Ishai, Omkant Pandey, and Amit Sahai. Public-coin differing-inputs obfuscation and its applications. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015*, volume 9015 of *LNCS*, pages 668–697. Springer, 2015.

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, 2010.

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013*, pages 669–684. ACM, 2013.

[KRW15] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015*, volume 9015 of *LNCS*, pages 378–400. Springer, 2015.

[KW16] Venkata Koppula and Brent Waters. Circular security counterexamples for arbitrary length cycles from lwe, 2016.

[LLJ15] Xianhui Lu, Bao Li, and Dingding Jia. KDM-CCA security from RKA secure authenticated encryption. In *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 559–583. Springer, 2015.

[Lys02] Anna Lysyanskaya. Unique signatures and verifiable random functions from the dh-ddh separation. In *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 597–612. Springer, 2002.

[MO14] Antonio Marcedone and Claudio Orlandi. Obfuscation $\Rightarrow$ (IND-CPA security $!\Rightarrow$ circular security). In *Security and Cryptography for Networks - 9th International Conference, SCN 2014*, volume 8642 of *LNCS*, pages 77–90. Springer, 2014.

[MPS16] Antonio Marcedone, Rafael Pass, and Abhi Shelat. Bounded KDM security from io and OWF. In *Security and Cryptography for Networks - 10th International Conference, SCN 2016*, volume 9841 of *LNCS*, pages 571–586. Springer, 2016.

[MTY11] Tal Malkin, Isamu Teranishi, and Moti Yung. Efficient circuit-size independent public key encryption with KDM security. In *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 507–526. Springer, 2011.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, 2003.

[QLH13] Baodong Qin, Shengli Liu, and Zhengan Huang. Key-dependent message chosen-ciphertext security of the cramer-shoup cryptosystem. In *Information Security and Privacy - 18th Australasian Conference, ACISP 2013*, volume 7959 of *LNCS*, pages 136–151. Springer, 2013.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Symposium on Theory of Computing, STOC 2014*, pages 475–484. ACM, 2014.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, 2009.

[Wee16] Hoeteck Wee. Kdm-security via homomorphic smooth projective hashing. In *Public-Key Cryptography - PKC 2016*, volume 9615 of *LNCS*, pages 159–179. Springer, 2016.

A A Framework for Generating Counterexamples for n-Circular Security

Recently, Bishop et al. [BHW15] introduced a new abstraction called an n-cycle tester which greatly simplifies the process of finding and describing counterexamples in the PKE setting. In this section, we extend n-Cycle Tester to the IBE setting, and show its usefulness in separating n-circular security from IND-CPA/IND-CCA security for IBE.

Definition A.1 (n-Cycle Tester)**.** An n-cycle tester in the IBE setting consists of four algorithms specified as follows:

- $\mathsf{Setup}(\lambda)$: on input a security parameter λ, output master public key mpk and master secret key msk.
- $\mathsf{Extract}(msk, id)$: on input msk and an identity $id \in I$, output a secret key sk_{id} for id.
- $\mathsf{Encrypt}(mpk, id, m)$: on input mpk and an identity $id \in I$ and a message $m \in M$, output a ciphertext c.
- $\mathsf{Test}(\mathbf{id}, \mathbf{c})$: on input $\mathbf{id} = (id_1, \dots, id_n)$ and $\mathbf{c} = (c_1, \dots, c_n)$, output "1" to indicate $\mathbf{c}$ forms encryption cycle w.r.t. $\mathbf{id}$ and "0" otherwise.

Testing Correctness. For any $\mathbf{id} = (id_1, \dots, id_n) \in I^n$, the advantage of algorithm Test defined in the following experiment is non-negligible in Λ.

$$\mathsf{Adv}_{\mathsf{Test}}(\lambda) = \Pr\left[\beta' = \beta : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\lambda); \\ sk_i \leftarrow \mathsf{Extract}(msk, id_i) \text{ for each } i \in [n]; \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \text{For } i = 1 \text{ to } n: \\ \beta = 1 : c_i \leftarrow \mathsf{Encrypt}(mpk, id_i, sk_{(i \bmod n)+1}); \\ \beta = 0 : c_i \leftarrow \mathsf{Encrypt}(mpk, id_i, 0^\ell); \\ \mathbf{c} \leftarrow (c_1, \dots, c_n); \\ \beta' \leftarrow \mathsf{Test}(\mathbf{id}, \mathbf{c}); \end{array}\right] - \frac{1}{2},$$

where the probability is taken over the random coins used by Setup, $\mathsf{Extract}$, $\mathsf{Encrypt}$, and Test.

IND-CPA Security. Similar to the case in the PKE setting [BHW15], an n-cycle tester in the IBE setting can be viewed as an IBE scheme without decryption algorithm, and recall the IND-CPA security experiment for IBE is not involved with decryption algorithm. Therefore, we can use the same security experiment (c.f. Definition 3.5) to capture the IND-CPA security of n-cycle tester in the IBE setting.

A.1 CPA Counterexample from Cycle Testers

Let $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be an IBE scheme with identity space I and message space $M_1 \times M_2$ and secret key space $SK_1 \subseteq M_1$. Let $\Gamma = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Test})$ be an n-cycle tester with the same identity space I and message space M_2 and secret key space $SK_2 \subseteq M_2$. We compose them to an IBE scheme Ψ with identity space I and secret key space $SK = SK_1 \times SK_2$ and message space $M = M_1 \times M_2$.

- $\mathsf{Setup}(\lambda)$: run $(mpk_1, msk_1) \leftarrow \Pi.\mathsf{Setup}(\lambda)$ and $(mpk_2, msk_2) \leftarrow \Gamma.\mathsf{Setup}(\lambda)$, output master public key $mpk = (mpk_1, mpk_2)$ and master secret key $msk = (msk_1, msk_2)$.
- $\mathsf{Extract}(msk, id)$: on input $msk = (msk_1, msk_2)$ and an identity $id \in I$, compute $sk_1 \leftarrow \Pi.\mathsf{Extract}(msk_1, id)$ and $sk_2 \leftarrow \Gamma.\mathsf{Extract}(msk_2, id)$, output a secret key $sk_{id} = (sk_1, sk_2)$.
- $\mathsf{Encrypt}(mpk, id, m)$: on input $mpk = (mpk_1, mpk_2)$ and an identity $id \in I$ and a message $m = (m_e, m_t) \in M$, compute $c_e \leftarrow \Pi.\mathsf{Encrypt}(mpk_1, id, m)$, $c_t \leftarrow \Gamma.\mathsf{Encrypt}(mpk_2, id, m_t)$, output a ciphertext $c = (c_e, c_t)$.
- $\mathsf{Decrypt}(sk_{id}, c)$: on input $sk_{id} = (sk_1, sk_2)$ and $c = (c_e, c_t)$, output $m \leftarrow \Pi.\mathsf{Decrypt}(sk_1, c_e)$.

- $\mathsf{Test}(\mathbf{id}, \mathbf{c})$: on input $\mathbf{id} = (id_1, \ldots, id_n)$ and $\mathbf{c} = (c_1, \ldots, c_n)$, parse $c_i = (c_{i,e}, c_{i,t})$ for each $i \in [n]$, set $\mathbf{c}_t = (c_{1,t}, \ldots, c_{n,t})$, output $\Gamma.\mathsf{Test}(\mathbf{id}, \mathbf{c}_t)$.

The correctness of $\Psi.\mathsf{Test}$ follows from that of $\Gamma.\mathsf{Test}$. If $(\mathbf{id}, \mathbf{c})$ is an circle encryption (resp. zero encryption) under Ψ, then $(\mathbf{id}, \mathbf{c}_t)$ is an circle encryption (resp. zero encryption) under Γ. Thereby, $\Psi.\mathsf{Test}$ distinguishes the two cases with the same advantage as that of $\Gamma.\mathsf{Test}$.

It remains to show the above construction is IND-CPA secure. This follows by a simple hybrid argument based on the fact that an encryption under Ψ is a combination of two IND-CPA secure encryptions, from Π and Γ respectively. We omit this proof as it is simplified version of the proof for Theorem A.1 that we show later.

A.2 CCA Counterexample from Cycle Testers

Let $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be an IBE scheme with identity space I and message space $M_1 \times M_2 \times C_2$ and secret key space $SK_1 \subseteq M_1$. Let $\Gamma = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Test})$ be a n-Cycle Tester with the same identity space I and message space M_2 and secret key space $SK_2 \subseteq M_2$ and ciphertext space C_2. We compose them to an IBE scheme Ψ with identity space I and message space $M = M_1 \times M_2$ and secret key space $SK = SK_1 \times SK_2$.

- $\mathsf{Setup}(\lambda)$: run $(mpk_1, msk_1) \leftarrow \Pi.\mathsf{Setup}(\lambda)$ and $(mpk_2, msk_2) \leftarrow \Gamma.\mathsf{Setup}(\lambda)$, output master public key $mpk = (mpk_1, mpk_2)$ and master secret key $msk = (msk_1, msk_2)$.
- $\mathsf{Extract}(msk, id)$: on input $msk = (msk_1, msk_2)$ and an identity $id \in I$, compute $sk_1 \leftarrow \Pi.\mathsf{Extract}(msk_1, id)$ and $sk_2 \leftarrow \Gamma.\mathsf{Extract}(msk_2, id)$, output a secret key $sk_{id} = (sk_1, sk_2)$.
- $\mathsf{Encrypt}(mpk, id, m)$: on input $mpk = (mpk_1, mpk_2)$ and an identity $id \in I$ and a message $m = (m_e, m_t) \in M$, compute $c_t \leftarrow \Gamma.\mathsf{Encrypt}(mpk_2, id, m_t)$, then compute $c_e \leftarrow \Pi.\mathsf{Encrypt}(mpk_1, id, (m_e, m_t, c_t))$, output a ciphertext $c = (c_e, c_t)$.
- $\mathsf{Decrypt}(sk_{id}, c)$: on input $sk_{id} = (sk_1, sk_2)$ and ciphertext $c = (c_e, c_t)$, run $\Pi.\mathsf{Decrypt}(sk_1, c_e)$. If the decryption result is not of the form (m_e, m_t, c_t), then output $\perp$. Otherwise, output the message $m = (m_e, m_t)$.
- $\mathsf{Test}(\mathbf{id}, \mathbf{c})$: on input $\mathbf{id} = (id_1, \ldots, id_n)$ and $\mathbf{c} = (c_1, \ldots, c_n)$, parse $c_i = (c_{i,e}, c_{i,t})$ for each $i \in [n]$, set $\mathbf{c}_t = (c_{1,t}, \ldots, c_{n,t})$, output $\Gamma.\mathsf{Test}(\mathbf{id}, \mathbf{c}_t)$.

Similar to the IND-CPA setting as analyzed above, the correctness of $\Psi.\mathsf{Test}$ follows from that of $\Gamma.\mathsf{Test}$. We then proceed to examine the security of Ψ.

Theorem A.1. *If Π is an IND-CCA secure IBE scheme, Γ is an IND-CPA secure n-cycle tester, then Ψ is an IND-CCA secure IBE scheme.*

Proof. We prove the IND-CCA security of Ψ via a sequence of games. Let m_0 and m_1 be the messages submitted by the adversary. We begin with Game 0 in which $\mathcal{CH}$ encrypts m_0 as the challenge ciphertext, and end with the hybrid that $\mathcal{CH}$ encrypts m_1 as the challenge ciphertext. In all these games, mpk and msk distribute identically to the real game, but either the structure of the challenge ciphertext or the rules of answering the decryption queries are changed in each two successive games. We specify these games as follows.

Game 0 (encrypt $m_0 = (m_{0,e}, m_{0,t})$ into c^*): $\mathcal{CH}$ interacts with $\mathcal{A}$ as follows.

1. Run $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$.
2. On extraction query $\langle id \rangle$, return $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$.
3. On decryption query $\langle id, c \rangle$, return $m \leftarrow \mathsf{Decrypt}(sk_{id}, c)$.
4. $\mathcal{A}$ submits (id^*, m_0, m_1) to $\mathcal{CH}$, where $m_0 = (m_{0,e}, m_{0,t})$ and $m_1 = (m_{1,e}, m_{1,t})$.

5. $\mathcal{CH}$ runs $c^* \leftarrow \mathsf{Encrypt}(mpk, id^*, m_0)$: computes $c_t^* \leftarrow \Gamma.\mathsf{Encrypt}(mpk_2, id^*, m_{0,t})$, $c_e^* \leftarrow \Pi.\mathsf{Encrypt}(mpk_1, id^*, (m_{0,e}, m_{0,t}, c_t^*))$, return $c^* = (c_e^*, c_t^*)$.
6. On extraction query $\langle id \rangle$ where $id \neq id^*$, return $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$.
7. On decryption query $\langle id, c \rangle$ where $(id, c) \neq (id^*, c^*)$, output $m \leftarrow \mathsf{Decrypt}(sk_{id}, c)$.

Game 1 (modify the decryption rules in Phase 2 step 7):

7. On decryption query $\langle id, c \rangle \neq \langle id^*, c^* \rangle$ where $c = (c_e, c_t)$ and $c^* = (c_e^*, c_t^*)$, if $id = id^*$ and $c_e = c_e^*$ directly output $\perp$, otherwise output $m \leftarrow \mathsf{Decrypt}(sk_{id}, c)$.

Game 2 (encrypt $m_{1,t}$ rather than $m_{0,t}$ when generating c_t^*):

5. $\mathcal{CH}$ computes $c_t^* \leftarrow \Gamma.\mathsf{Encrypt}(mpk_2, id^*, m_{1,t})$, computes c_e^* as in Game 1, outputs $c^* = (c_e^*, c_t^*)$.

Game 3 (encrypt $(m_{1,e}, m_{1,t}, c_t^*)$ rather than $(m_{0,e}, m_{0,t}, c_t^*)$ when generating c_e^*):

5. $\mathcal{CH}$ computes $c_e^* \leftarrow \Pi.\mathsf{Encrypt}(mpk_1, id^*, (m_{1,e}, m_{1,t}, c_t^*))$, computes c_t^* as in Game 2, outputs $c^* = (c_e^*, c_t^*)$.

Game 4 (modify back the decryption rules in Phase 2 step 7):

7. On decryption query $\langle id, c \rangle \neq \langle id^*, c^* \rangle$, output $m \leftarrow \mathsf{Decrypt}(sk_{id}, c)$.

Lemma A.2. *Game 0 and Game 1 are equivalent.*

Proof. We note that the only difference between Game 0 and Game 1 is that when answering decryption queries in Phase 2 $\mathcal{CH}$ directly returns $\perp$ if $id = id^*$ and $c_e = c_e^*$. Note that for decryption query of the form $\langle id^*, (c_e^*, c_t) \rangle$: if $c_t = c_t^*$, the query is illegal and will be rejected with $\perp$; if $c_t \neq c_t^*$, the ciphertext is not valid since according to the construction of Ψ the third element of the decryption result of c_e^* must be c_t^*. Thus, such change of decryption rule in Phase 2 is purely conceptual and the two games are perfectly equivalent. □

Lemma A.3. *Game 1 and Game 2 are computationally indistinguishable, given Γ is IND-CPA secure.*

Proof. We prove this lemma by giving a reduction to the IND-CPA security of Γ. Suppose there is a PPT adversary $\mathcal{A}$ that can distinguish Game 1 and Game 2, then we can construct an algorithm $\mathcal{B}$ against the IND-CPA security of Γ by interacting with $\mathcal{A}$ as follows:

1. Given mpk_2 (where $(mpk_2, msk_2) \leftarrow \Gamma.\mathsf{Setup}(\lambda)$) from the n-cycle tester challenger, $\mathcal{B}$ runs $(mpk_1, msk_1) \leftarrow \Pi.\mathsf{Setup}(\lambda)$, sets $mpk = (mpk_1, mpk_2)$ and sends mpk to $\mathcal{A}$.
2. On extraction query $\langle id \rangle$, $\mathcal{B}$ first computes $sk_1 \leftarrow \Pi.\mathsf{Extract}(msk_1, id)$ on its own, then makes extraction query $\langle id \rangle$ to its challenger and gets back $sk_2 \leftarrow \Gamma.\mathsf{Extract}(msk_2, id)$, $\mathcal{B}$ sends $sk_{id} = (sk_1, sk_2)$ to $\mathcal{A}$.
3. On decryption query $\langle id, c \rangle$, $\mathcal{B}$ computes $sk_1 \leftarrow \Pi.\mathsf{Extract}(msk_1, id)$, then answers the decryption query with sk_1. Note that the second component of sk_{id}, namely sk_2, is not used in decryption, thus $\mathcal{B}$ can handle all decryption queries correctly.
4. $\mathcal{A}$ submits (id^*, m_0, m_1), where $m_0 = (m_{0,e}, m_{0,t})$ and $m_1 = (m_{1,e}, m_{1,t})$.
5. $\mathcal{B}$ submits $(id^*, m_{0,t}, m_{1,t})$ to its own challenger, and receives back a challenge ciphertext $c_t^* \leftarrow \Gamma.\mathsf{Encrypt}(mpk_2, id^*, m_{\beta,t})$ for some unknown bit β. $\mathcal{B}$ then computes $c_e^* \leftarrow \Pi.\mathsf{Encrypt}(mpk_1, id^*, (m_{0,e}, m_{0,t}, c_t^*))$, and sends $c^* = (c_e^*, c_t^*)$ to $\mathcal{A}$.
6. On extraction query $\langle id \rangle \neq \langle id^* \rangle$, $\mathcal{B}$ responds the same way as in Phase 1.

7. On decryption query $\langle id, c\rangle \neq \langle id^*, c^*\rangle$, $\mathcal{B}$ responds the same way as in Phase 1 except directly reject the queries of the form $\langle id^*, (c_e^*, c_t)\rangle$ with $\perp$.

In the above, $\mathcal{B}$ perfectly simulates Game 1 if c_t^* is a Γ-encryption of $m_{0,t}$, and $\mathcal{B}$ perfectly simulates Game 2 if c_t^* is a Γ-encryption of $m_{1,t}$. Therefore, $\mathcal{B}$ has the same advantage against the IND-CPA security of Γ as $\mathcal{A}$ distinguishes Game 1 and Game 2. According to the hypothesis that Γ is IND-CPA secure, Game 1 and Game 2 are computationally indistinguishable. This proves the lemma. $\square$

Lemma A.4. *Game 2 and Game 3 are computationally indistinguishable, given the IND-CCA security of* Π*.*

Proof. We prove this lemma by giving a reduction to the IND-CCA security of Π. Suppose there is a PPT adversary $\mathcal{A}$ that can distinguish Game 2 and Game 3, then we can construct an algorithm $\mathcal{B}$ against the IND-CCA security of Π by interacting with $\mathcal{A}$ as follows:

1. Given mpk_1 (where $(mpk_1, msk_1) \leftarrow \Pi.\mathsf{Setup}(\lambda)$) from the IBE challenger, $\mathcal{B}$ computes $(mpk_2, msk_2) \leftarrow \Gamma.\mathsf{Setup}(\lambda)$, sets $mpk = (mpk_1, mpk_2)$, and sends mpk to $\mathcal{A}$.
2. On extraction query $\langle id\rangle$, $\mathcal{B}$ computes $sk_2 \leftarrow \Gamma.\mathsf{Extract}(msk_2, id)$ on its own, makes extraction query $\langle id\rangle$ to its challenger and gets back $sk_1 \leftarrow \Pi.\mathsf{Extract}(msk_1, id)$, $\mathcal{B}$ sends $sk_{id} = (sk_1, sk_2)$ to $\mathcal{A}$.
3. On decryption query $\langle id, c\rangle$, $\mathcal{B}$ parses $c = (c_e, c_t)$, then submits decryption query $\langle id, c_e\rangle$ to its challenger and gets the reply (m_e, m_t, c_t'). If $c_t' \neq c_t$, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ returns (m_e, m_t).
4. $\mathcal{A}$ submits (id^*, m_0, m_1), where $m_0 = (m_{0,e}, m_{0,t})$ and $m_1 = (m_{1,e}, m_{1,t})$.
5. $\mathcal{B}$ computes $c_t^* \leftarrow \Gamma.\mathsf{Encrypt}(mpk_2, id^*, m_{1,t})$, submits $(id^*, (m_{0,e}, m_{0,t}, c_t^*), (m_{1,e}, m_{1,t}, c_t^*))$ to its challenger. As soon as $\mathcal{B}$ receives back $c_e^* \leftarrow \Pi.\mathsf{Encrypt}(mpk_1, id^*, (m_{\beta,e}, m_{\beta,t}, c_t^*))$ for some unknown bit β from its challenger, $\mathcal{B}$ sends $c^* = (c_e^*, c_t^*)$ to $\mathcal{A}$.
6. On extraction query $\langle id\rangle \neq \langle id^*\rangle$, $\mathcal{B}$ responds the same way as in Phase 1.
7. On decryption query $\langle id, c\rangle \neq \langle id^*, c^*\rangle$, $\mathcal{B}$ responds the same way as in Phase 1 except directly rejects the queries of the form $\langle id^*, (c_e^*, c_t)\rangle$ with $\perp$. Note that $\mathcal{B}$ is able to handle all decryption queries in Phase 2 properly since it can always make decryption queries $\langle id, c_e\rangle \neq \langle id^*, c_e^*\rangle$ to its challenger.

According to the definitions, $\mathcal{B}$ perfectly simulates Game 2 if c_e^* is a Π-encryption of $(m_{0,e}, m_{0,t}, c_t^*)$, and $\mathcal{B}$ perfectly simulates Game 3 if c_e^* is a Π-encryption of $(m_{1,e}, m_{1,t}, c_t^*)$. Therefore, $\mathcal{B}$ has the same advantage against the IND-CPA security of Π as $\mathcal{A}$ distinguishing Game 2 and Game 3. According to the hypothesis that Π is IND-CCA secure, Game 2 and Game 3 are computationally indistinguishable. This proves the lemma. $\square$

Lemma A.5. *Game 3 and Game 4 are equivalent.*

Proof. The only difference between Game 3 and Game 4 is that $\mathcal{CH}$ directly returns $\perp$ when $id = id^*$ and $c_e = c_e^*$ in Game 3 whereas $\mathcal{CH}$ returns $\perp$ when $id = id^*$ and $c = c^*$ in Game 4. Nevertheless, the response to all the decryption queries are identical. This case is the mirror image of the argument made in proof of Lemma A.2. This proves the lemma. $\square$

According to the definition, in Game 0 c^* is a Ψ-encryption of m_0, while in Game 4 c^* is a Ψ-encryption of m_1. The above lemmas indicates that Game 0 and Game 4 are computationally indistinguishable. Thus, the desired IND-CCA security immediately follows. $\square$