Simulation-Based Selective Opening Security for Receivers under Chosen-Ciphertext Attacks

Zhengan Huang[1], Junzuo Lai[2,3,✉], Wenbin Chen[1], Man Ho Au[4], Zhen Peng[5], and Jin Li[1]

1. School of Computer Science, Guangzhou University, Guangzhou, China
zhahuang.sjtu@gmail.com, cwb2011@gzhu.edu.cn, jinli71@gmail.com
2. College of Information Science and Technology, Jinan University, Guangzhou, China
3. State Key Laboratory of Cryptology, Beijing, China
laijunzuo@gmail.com
4. Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
csallen@comp.polyu.edu.hk
5. Westone Cryptologic Research Center, Beijing, China
peng.zhen@westone.com.cn

August 16, 2018

Abstract. Security against selective opening attack (SOA) for receivers requires that in a multi-user setting, even if an adversary has access to all ciphertexts, and adaptively corrupts some fraction of the users to obtain the decryption keys corresponding to some of the ciphertexts, the remaining (potentially related) ciphertexts retain their privacy. In this paper, we study simulation-based selective opening security for receivers of public key encryption (PKE) schemes under chosen-ciphertext attacks (RSIM-SO-CCA).

Concretely, we first show that some known PKE schemes meet RSIM-SO-CCA security. Then, we introduce the notion of master-key SOA security for identity-based encryption (IBE), and extend the Canetti-Halevi-Katz (CHK) transformation to show generic PKE constructions achieving RSIM-SO-CCA security. Finally, we show how to construct an IBE scheme achieving master-key SOA security.

Keywords: simulation-based security, selective opening security for receivers, chosen-ciphertext attacks, public-key encryption, identity-based encryption

1 Introduction

Selective opening attacks (SOA) concern a multi-user scenario, where an adversary breaks into a subset of honestly created ciphertexts and tries to learn information on the plaintexts of some unopened (but potentially related) ciphertexts. The notion of selective opening attacks is considered in two settings: sender corruption and receiver corruption. In the sender corruption setting, there is one receiver and many senders; senders may be corrupted, with the corruption exposing their coins and messages. In the receiver corruption setting, there is a single sender and many receivers; each receiver has its own public and secret key; receivers may be corrupted, with corruption exposing their secret key. For each setting, there are two ways to formalize the requirement of selective opening security notion [1, 4], namely indistinguishability-based (IND-SO) and simulation-based (SIM-SO) ones.

IND-SO security requires that no adversary can distinguish an unopened ciphertext from an encryption of a fresh message, which is distributed according to the conditional probability distribution (conditioned on the opened ciphertexts). Such a security notion requires that

the joint plaintext distribution should be "efficiently conditionally re-samplable". On the other hand, SIM-SO security requires that anything that can be computed by an adversary from all the ciphertexts and the opened messages together with the corrupted information can also be computed by a simulator with only the opened messages. SIM-SO security imposes no limitation on the message distribution, and it implies IND-SO security. The selective opening security is further classified into two notions, security against selective opening chosen-plaintext attacks (SO-CPA) and that against selective opening chosen-ciphertext attacks (SO-CCA), depending on whether the adversary has access to a decryption oracle or not.

In this paper, we study simulation-based selective opening security for receivers of public key encryption (PKE) schemes under chosen-ciphertext attacks (RSIM-SO-CCA).

1.1 Our Contribution

We first show that some known PKE schemes meet RSIM-SO-CCA security. More specifically, we show that the DDH-based PKE scheme (for single-bit messages) proposed by Cramer and Shoup in [11] and the DCR-based PKE scheme proposed by Cramer and Shoup in [12] both achieve RSIM-SO-CCA security.

Then, we introduce the notion of master-key selective opening security (mSO) for identity-based encryption (IBE), which focuses on IBE in the setting of *multiple private key generators* [34, 33]. In a master-key selective opening attack for IBE, after seeing the challenge ciphertexts, the adversary can corrupt some of the private key generators, by obtaining their master secret keys. The goal of master-key SOA security is to guarantee that the messages of the users, whose private key generators are uncorrupted, are still confidential. As pointed out by Bellare et al. [3], the standard security notions for IBE naturally provide security against *non-adaptive* receiver corruption (i.e., the adversaries are allowed to query the private key generation oracle). However, to the best of our knowledge, SOA security notions for IBE in the *adaptive* receiver corruption setting have never been formalized, and are less studied. Our mSOA security notions focus on SOA security in the adaptive receiver corruption setting. Furthermore, we stress that in the experiments defining mSOA security, the adversaries are actually *more powerful*. They are allowed to corrupt the *private key generators*, *not the receivers (users)*, even *after* seeing the challenge ciphertexts.

Next, we show a generic construction of RSIM-SO-CCA secure PKE schemes, by applying the CHK method [9] to any SIM-sID-mSO-CPA secure IBE scheme. Note that the CHK method does not work in the sender corruption setting [19]. Our result shows that it does work in the receiver corruption setting.

Finally, we show that a hybrid IBE scheme constructed from an identity-based key encapsulation mechanism (IB-KEM) scheme and a data encapsulation mechanism (DEM) scheme, is SIM-sID-mSO-CPA secure, if the underlying IB-KEM scheme is IND-sID-CPA secure, and the DEM scheme has some special properties.

1.2 Related Work

SOA for Senders. In [1], Bellare, Hofheinz and Yilek showed that any lossy encryption is able to achieve indistinguishability-based selective opening security for senders under chosen-plaintext attacks (SIND-SO-CPA), and simulation-based selective opening security for senders under

chosen-plaintext attacks (SSIM-SO-CPA) is achievable as well if the lossy encryption is "efficiently openable". Recently, Hofheinz et al. [23] showed how to construct SSIM-SO-CPA secure PKE schemes with compact ciphertexts.

Hemenway et al. [19] showed that SIND-SO-CCA secure PKE can be obtained from selective-tag weakly secure and separable tag-based PKE with the help of chameleon hashing. Hofheinz [22] showed how to get (SIND/SSIM-SO-CCA) secure PKE with compact ciphertexts from all-but-many lossy trapdoor functions (ABM-LTF). Recently, Boyen and Li [8] presented an ABM-LTF construction from lattices and obtained a SIND-SO-CCA secure PKE from lattices.

For SSIM-SO-CCA security, Fehr et al. [17] proved that sender-equivocable (NC-CCA) security implies SSIM-SO-CCA security, and showed how to construct PKE schemes with NC-CCA security based on hash proof systems with explainable domains and L-cross-authentication codes (L-XAC, in short). Huang et al. [24, 25] showed that using the method proposed in [17] to construct SSIM-SO-CCA secure PKE, L-XAC needs to be *strong*. Heuer et al. [20, 21] showed that some practical PKE schemes is SSIM-SO-CCA secure in the random oracle model. Recently, Libert et al. [31] presented an ABM-LTF construction from lattices and used it to obtain a SSIM-SO-CCA secure PKE from lattices.

SOA for Receivers. Hazay et al. [18] showed that RSIM-SO-CPA secure PKE schemes can be achieved from non-committing encryption for receiver (NCER) [10] and RIND-SO-CPA secure PKE schemes can be achieved from a tweaked variant of NCER which, in turn, can be instantiated from a variety of basic, well-established, assumptions. Recently, Jia et al. [27] showed that the PKE scheme based on HPS [12] achieves RIND-SO-CCA security and provided a general construction of RIND-SO-CCA secure PKE schemes by combining any RIND-SO-CPA secure scheme with any regular CCA secure scheme, along with an appropriate non-interactive zero-knowledge proof. In this paper, we consider RSIM-SO-CCA secure PKE schemes.

SOA for IBE. Bellare et al. [3] considered the SOA security in the IBE setting and proposed a general paradigm to achieve SIM-SO-CPA security from IND-ID-CPA secure and "One-Sided Publicly Openable" (1SPO) IBE schemes. For SIM-SO-CCA secure IBE, Lai et al. [28] gave a generic construction from an IND-ID-CCA secure extractable IBE with "One-Sided Public Openability"(1SPO), a collision-resistant hash function and a strengthened cross-authentication code.

IBE in the multi-PKG setting. In some application scenarios (e.g., a multi-domain ad hoc network which is formed by different organizations [30]), IBE in the multi-PKG setting is required. Motivated by earlier work on anonymous credential systems, Holt [26] proposed a security notion for IBE in the multi-PKG setting. But this security model just focuses on anonymity, and does not allow the adversary to extract any user secret key at all. Wang and Cao [34] put forth some IBE schemes, which consider multiple public parameters, but the security model in [34] is still in the standard single-PKG setting. In 2008, Paterson and Srinivasan [33] formalized several security notions for IBE in the multi-PKG setting. In the security games of these notions, an adversary is allowed to obtain the master secret keys of some of the PKGs by oracle queries. Paterson and Srinivasan also provided IBE constructions meeting these security requirements in [33]. However, to the best of our knowledge, no SOA security notions for IBE in the multi-PKG setting has been proposed or researched.

2 Preliminaries

Notations. Throughout this paper, we denote the security parameter by $\lambda \in \mathbb{N}$. For $n \in \mathbb{N}$, let $[n] := \{1, 2, \cdots, n\}$. We use boldface to denote vectors, e.g., $\mathbf{x}$. For vector $\mathbf{x}$, let $|\mathbf{x}|$ denote the number of components in this vector, and $\mathbf{x}[i]$ denote its i^{th} component for $i \in [|\mathbf{x}|]$. For a set $I \subseteq [|\mathbf{x}|]$, let $\mathbf{x}[I] := (\mathbf{x}[i])_{i \in I}$, and $\mathbf{x}[|\mathbf{x}|] \backslash I := (\mathbf{x}[i])_{i \notin I}$. For a finite set S, we denote by $s \leftarrow S$ the process of sampling s uniformly random from S. For a distribution X, we denote by $x \leftarrow X$ the process of sampling x from X. For a probabilistic algorithm A, let $\mathcal{R}_A$ denote the randomness space of A. We denote by $y \leftarrow A(x; r)$ the process of running A on input x and with randomness $r \in \mathcal{R}_A$, and assigning y the result. We write $y \leftarrow A(x)$ for $y \leftarrow A(x; r)$ with uniformly chosen $r \leftarrow \mathcal{R}_A$. For vectors $\mathbf{x}$ and $\mathbf{r}$ with $|\mathbf{x}| = |\mathbf{r}|$, we write $A(\mathbf{x}; \mathbf{r}) := (A(\mathbf{x}[1]; \mathbf{r}[1]), A(\mathbf{x}[2]; \mathbf{r}[2]), \cdots, A(\mathbf{x}[|\mathbf{x}|]; \mathbf{r}[|\mathbf{x}|]))$. We write PPT for probabilistic polynomial-time.

For any event evt (e.g., bad_1 in Fig. 13) defined in a game $\mathbf{G}$, wlog, we denote by $\Pr[\mathsf{evt}]$ the probability that evt occurs in $\mathbf{G}$.

The decisional Diffie-Hellman assumption. We recall the decisional Diffie-Hellman (DDH) assumption and its variant as follows.

Definition 1 (The DDH assumption). *Let $\mathbb{G}_q$ be a cyclic group of prime order q. We say that* the decisional Diffie-Hellman (DDH) assumption holds for $\mathbb{G}_q$*, if for any PPT distinguisher D,*

$$|\Pr[D(g, g^a, g^b, g^{ab}) = 1] - \Pr[D(g, g^a, g^b, g^c) = 1]|$$

is negligible, where $g \leftarrow \mathbb{G}_q \setminus \{1\}$ and $a, b, c \leftarrow \mathbb{Z}_q$.

As pointed out in [10], the DDH assumption implies that for $g \leftarrow \mathbb{G}_q \setminus \{1\}$, $a, b \leftarrow \mathbb{Z}_q$, (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^{ab+1}) are computationally indistinguishable. Formally, we have the following lemma.

Lemma 1. *If the DDH assumption holds for $\mathbb{G}_q$, then for any PPT distinguisher (denoted by D_{DDH1}),*

$$|\Pr[D_{DDH1}(g, g^a, g^b, g^{ab}) = 1] - \Pr[D_{DDH1}(g, g^a, g^b, g^{ab+1}) = 1]|$$

is negligible, where $g \leftarrow \mathbb{G}_q \setminus \{1\}$ and $a, b \leftarrow \mathbb{Z}_q$.

The Decision Composite Residuosity assumption. We recall Paillier's decision composite residuosity (DCR) assumption [32] and its variant as follows.

Let p, q, p', q' be distinct odd primes with $p = 2p' + 1$ and $q = 2q' + 1$, where both p' and q' are λ bits in length. Let $N = pq$ and $N' = p'q'$. Then the group $\mathbb{Z}^*_{N^2}$ can be decomposed as an inner direct product $\mathbb{G}_N \cdot \mathbb{G}_{N'} \cdot \mathbb{G}_2 \cdot \mathbb{S}$, where $\mathbb{G}_i$ is a cyclic group of order i, and $\mathbb{S}$ is the subgroup of $\mathbb{Z}^*_{N^2}$ generated by $(-1 \bmod N^2)$. It's easy to see that the order of $1 + N \bmod N^2$ in $\mathbb{Z}^*_{N^2}$ is N, and for any $m \in \mathbb{Z}_N$, $(1+N)^m \bmod N^2 = 1 + mN$. Let $\mathcal{G}$ denote a DCR instance generator, which takes 1^λ as input and returns numbers N and N' as above.

Definition 2 (The DCR assumption). *We say that* the Decision Composite Residuosity (DCR) assumption holds for $\mathcal{G}$*, if for any PPT distinguisher D,*

$$|\Pr[D(N, g^2) = 1] - \Pr[D(N, g^{2N}) = 1]|$$

is negligible, where $(N, N') \leftarrow \mathcal{G}(1^\lambda)$ and $g \leftarrow \mathbb{Z}^*_{N^2}$.

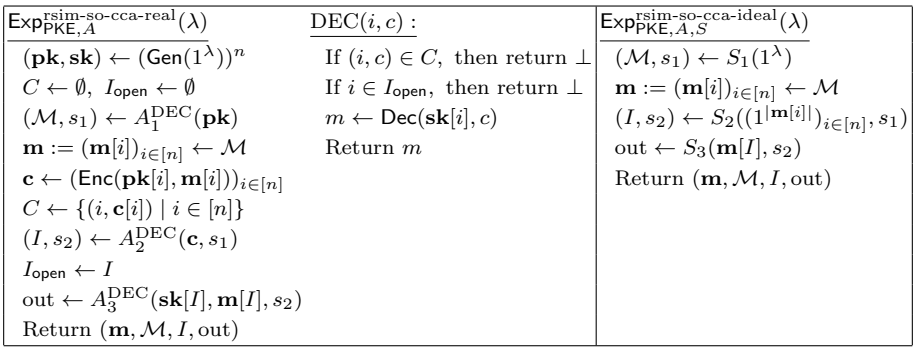

Fig. 1. Experiments for defining RSIM-SO-CCA security of a PKE scheme PKE.

As pointed out in [10], the DCR assumption implies that for $(N, N') \leftarrow \mathcal{G}(1^\lambda)$ and $g \leftarrow \mathbb{Z}^*_{N^2}$, (N, g^{2N}) and $(N, (1+N)\cdot g^{2N})$ are computationally indistinguishable. Formally, we have the following lemma.

Lemma 2. *If the DCR assumption holds for $\mathcal{G}$, then for any PPT distinguisher (denoted by D_{DCR1}),*

$$|\Pr[D_{DCR1}(N, g^{2N}) = 1] - \Pr[D_{DCR1}(N, (1+N)\cdot g^{2N}) = 1]|$$

is negligible, where $(N, N') \leftarrow \mathcal{G}(1^\lambda)$ and $g \leftarrow \mathbb{Z}^*_{N^2}$.

The formal definitions of IBE, strong one-time signature and IB-KEM are recalled in Appendix B, C and D, respectively.

3 Simulation-based SOA security for receivers under chosen-ciphertext attacks

In this section, we formalize the notion of simulation-based selective opening security for receivers under chosen-ciphertext attacks (RSIM-SO-CCA security). Similar to other security notions in the CCA setting, in the following experiment defining SO-CCA security for receivers, the adversary is given access to a decryption oracle. However, compared with SIM-SO-CCA security for senders, there are some subtleties in our formalization. First, in the receiver setting, there are n public/secret key pairs, so the adversary has to specify the secret key when she makes decryption queries. Second, in our following definition, we require that the adversary cannot query the decryption oracle on the opened secret keys. This is a very mild and reasonable condition, since once the adversary has obtained the secret key of some user, she can decrypt all the ciphertexts received by the user. It's not necessary for the adversary to query the decryption oracle on the opened secret keys.

The formal definition is as follows.

Definition 3 (RSIM-SO-CCA). *A public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is RSIM-SO-CCA secure, if for any polynomially bounded function $n > 0$, and any PPT adversary A, there is a PPT simulator S, such that for any PPT distinguisher D, the advantage*

$$\mathbf{Adv}^{\text{rsim-so-cca}}_{\mathsf{PKE},A,S,D}(\lambda) := |\Pr[D(\mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE},A}(\lambda)) = 1] - \Pr[D(\mathsf{Exp}^{\text{rsim-so-cca-ideal}}_{\mathsf{PKE},A,S}(\lambda)) = 1]|$$

is negligible, where $\mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE},A}(\lambda)$ and $\mathsf{Exp}^{\text{rsim-so-cca-ideal}}_{\mathsf{PKE},A,S}(\lambda)$ are defined in Fig. 1.

$\mathsf{CS.Gen}(1^\lambda)$:	$\mathsf{CS.Enc}(pk, m)$:	$\mathsf{CS.Dec}(sk, (u, v, w, e))$:
$g_1 \leftarrow \mathbb{G}_q \setminus \{1\},\ z \leftarrow \mathbb{Z}_q^*$	$(g_1, g_2, h, \theta, \varphi, hk) \leftarrow pk$	$(x, y, a, b, a', b') \leftarrow sk$
$x, y, a, b, a', b' \leftarrow \mathbb{Z}_q$	$r \leftarrow \mathbb{Z}_q$	$\alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$
$g_2 \leftarrow g_1^z,\ h \leftarrow g_1^x g_2^y$	$u \leftarrow g_1^r,\ v \leftarrow g_2^r$	If $u^{a+\alpha a'} v^{b+\alpha b'} \neq e$, then return $\perp$
$\theta \leftarrow g_1^a g_2^b,\ \varphi \leftarrow g_1^{a'} g_2^{b'}$	$w \leftarrow g_1^m h^r$	$\delta \leftarrow w \cdot (u^x v^y)^{-1}$
$hk \leftarrow \mathsf{HGen}(1^\lambda)$	$\alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$	If $\delta = 1$, then $m \leftarrow 0$
$pk \leftarrow (g_1, g_2, h, \theta, \varphi, hk)$	$e \leftarrow (\theta \varphi^\alpha)^r$	If $\delta = g_1$, then $m \leftarrow 1$
$sk \leftarrow (x, y, a, b, a', b')$	Return (u, v, w, e)	Else, then $m \leftarrow \perp$
$tk \leftarrow z$		Return m
Return (pk, sk, tk)		

Fig. 2. Construction of $\mathsf{PKE}_{\mathsf{CS}}$.

Remark 1. We note that $I_{\mathsf{open}} \neq \emptyset$ only if the adversary has made the opening query. The definition can be extended to the version where the adversary is allowed to make multiple opening queries adaptively [6, 17]. For simplicity, in this paper we only consider the adversaries making one opening query.

4 Concrete constructions of RSIM-SO-CCA secure PKE

In this section, we show that some known PKE schemes meet RSIM-SO-CCA security. More specifically, we show that the DDH-based PKE scheme (for single-bit messages) proposed by Cramer and Shoup in [11] and the DCR-based PKE scheme proposed by Cramer and Shoup in [12] both achieve RSIM-SO-CCA security.

4.1 RSIM-SO-CCA secure PKE under the DDH assumption

We show that the Cramer-Shoup encryption scheme [11] for single-bit messages meets RSIM-SO-CCA security.

Let $\mathbb{G}_q$ be a cyclic group of prime order q, and $\mathsf{Hash} = (\mathsf{HGen}, \mathsf{HEvl})$ be a family of collision-resistant (CR) hash functions $\Lambda_{(\cdot)} : \mathbb{G}_q^3 \to \mathbb{Z}_q$. The Cramer-Shoup encryption scheme [11] (for single bit) $\mathsf{PKE}_{\mathsf{CS}} := (\mathsf{CS.Gen}, \mathsf{CS.Enc}, \mathsf{CS.Dec})$ is shown in Fig. 2. We refer the readers to [11] for the correctness of $\mathsf{PKE}_{\mathsf{CS}}$. We stress that here we set a *trapdoor key* tk which is z satisfying $g_2 = g_1^z$ (see Fig. 2). The additional trapdoor key is useless for normal encryption/decryption, but will be used in the following security proof.

Formally, we have the following theorem. The proof is inspired by that of Hazay et al. [18].

Theorem 1. $\mathsf{PKE}_{\mathsf{CS}}$ *(for single-bit messages) is RSIM-SO-CCA secure.*

Proof. For any PPT adversary A attacking $\mathsf{PKE}_{\mathsf{CS}}$ in the sense of RSIM-SO-CCA, let q_d denote the number of decryption queries made by A. We prove the theorem with a sequence of games $\mathbf{G}_{-1} - \mathbf{G}_{3n}, \mathbf{G}'_{3n}$ in Fig. 3.

Firstly, note that the view of A in game $\mathbf{G}_{-1}$ is exactly the same as in $\mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE}_{\mathsf{CS}}, A}$, so the final outputs of these two games are identical, i.e., $\mathbf{G}_{-1} = \mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE}_{\mathsf{CS}}, A}$.

Games $\mathbf{G}_0$ and $\mathbf{G}_{-1}$ are identical except that the challenge ciphertext vector is encrypted with secret key $\mathbf{sk}$ (lines 06, 12, 19). We stress that the view of A in game $\mathbf{G}_0$ is exactly the same as in $\mathbf{G}_{-1}$. That's because for all $j \in [n]$, $\mathbf{sk}[j] = (x, y, a, b, a', b', z)$, $\mathbf{pk}[j] = (g_1, g_2, h, \theta, \varphi, hk)$, we have that

$$w = g_1^{\mathbf{m}[j]} u^x v^y = g_1^{\mathbf{m}[j]} g_1^{rx} g_2^{ry} = g_1^{\mathbf{m}[j]} (g_1^x g_2^y)^r = g_1^{\mathbf{m}[j]} h^r,$$

Games $\mathbf{G}_{-1} - \mathbf{G}_{3n}$

01 $(\mathbf{pk}, \mathbf{sk}, \mathbf{tk}) \leftarrow (\mathsf{CS.Gen}(1^\lambda))^n$
02 $C \leftarrow \emptyset,\ I_{\mathsf{open}} \leftarrow \emptyset,\ \mathbf{sk}' \leftarrow \mathbf{sk}$
03 $(\mathcal{M}, s_1) \leftarrow A_1^{\mathrm{DEC}}(\mathbf{pk}),\ \mathbf{m} \leftarrow \mathcal{M}$
04 For $j = 1$ to n
05 $\quad (g_1, g_2, h, \theta, \varphi, hk) \leftarrow \mathbf{pk}[j]$
06 $\quad (x, y, a, b, a', b') \leftarrow \mathbf{sk}[j]$ //$\mathbf{G}_0 - \mathbf{G}_{3n}$
07 $\quad r \leftarrow \mathbb{Z}_q,\ u \leftarrow g_1^r$
08 $\quad v \leftarrow g_2^r$
09 $\quad$ If $j \le i$, then $v \leftarrow g_1 g_2^r$ //$\mathbf{G}_1 - \mathbf{G}_i$
10 $\quad v \leftarrow g_1 g_2^r$ //$\mathbf{G}_{n+1} - \mathbf{G}_{3n}$
11 $\quad w \leftarrow g_1^{\mathbf{m}[j]} h^r$ //$\mathbf{G}_{-1}$
12 $\quad w \leftarrow g_1^{\mathbf{m}[j]} u^x v^y$ //$\mathbf{G}_0 - \mathbf{G}_{3n}$
13 $\quad$ If $j \le i$, then //$\mathbf{G}_{2n+1} - \mathbf{G}_{2n+i}$
14 $\quad\quad w \leftarrow g_1^y h^r,\ z \leftarrow \mathbf{tk}[j]$ //$\mathbf{G}_{2n+1} - \mathbf{G}_{2n+i}$
15 $\quad\quad (x', y') \leftarrow (x + \mathbf{m}[j] \cdot z, y - \mathbf{m}[j])$ //$\mathbf{G}_{2n+1} - \mathbf{G}_{2n+i}$
16 $\quad\quad \mathbf{sk}'[j] \leftarrow (x', y', a, b, a', b')$ //$\mathbf{G}_{2n+1} - \mathbf{G}_{2n+i}$
17 $\quad \alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$
18 $\quad e \leftarrow (\theta \varphi^\alpha)^r$ //$\mathbf{G}_{-1}$
19 $\quad e \leftarrow u^{a+\alpha a'} v^{b+\alpha b'}$ //$\mathbf{G}_0 - \mathbf{G}_{3n}$
20 $\quad \mathbf{c}[j] \leftarrow (u, v, w, e),\ C \leftarrow C \bigcup \{(j, \mathbf{c}[j])\}$
21 $(I, s_2) \leftarrow A_2^{\mathrm{DEC}}(\mathbf{c}, s_1),\ I_{\mathsf{open}} \leftarrow I$
22 $out \leftarrow A_3^{\mathrm{DEC}}(\mathbf{sk}'[I], \mathbf{m}[I], s_2)$
23 Return $(\mathbf{m}, \mathcal{M}, I, out)$

On query $\mathrm{DEC}(j, c')$:
24 If $(j, c') \in C$, then return $\perp$
25 If $j \in I_{\mathsf{open}}$, then return $\perp$
26 $(x, y, a, b, a', b') \leftarrow \mathbf{sk}[j],\ (u, v, w, e) \leftarrow c'$
27 If $j \le i$, then //$\mathbf{G}_{n+1} - \mathbf{G}_{n+i}$
28 $\quad$ If $u^{\mathbf{tk}[j]} \ne v$, then return $\perp$ //$\mathbf{G}_{n+1} - \mathbf{G}_{n+i}$
29 If $u^{\mathbf{tk}[j]} \ne v$, then return $\perp$ //$\mathbf{G}_{2n+1} - \mathbf{G}_{3n}$
30 $\alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$
31 If $u^{a+\alpha a'} v^{b+\alpha b'} \ne e$, then return $\perp$
32 $\delta \leftarrow w \cdot (u^x v^y)^{-1}$
33 If $\delta = 1$, then $m' \leftarrow 0$
34 If $\delta = g_1$, then $m' \leftarrow 1$
35 Else, then $m' \leftarrow \perp$
36 Return m'

Game $\mathbf{G}'_{3n}$

37 $(\mathbf{pk}, \mathbf{sk}, \mathbf{tk}) \leftarrow (\mathsf{CS.Gen}(1^\lambda))^n$
38 $C \leftarrow \emptyset,\ I_{\mathsf{open}} \leftarrow \emptyset$
39 $(\mathcal{M}, s_1) \leftarrow A_1^{\mathrm{DEC}}(\mathbf{pk}),\ \mathbf{m} \leftarrow \mathcal{M}$
40 For $j = 1$ to n
41 $\quad (g_1, g_2, h, \theta, \varphi, hk) \leftarrow \mathbf{pk}[j]$
42 $\quad (x, y, a, b, a', b') \leftarrow \mathbf{sk}[j]$
43 $\quad r \leftarrow \mathbb{Z}_q,\ u \leftarrow g_1^r$
44 $\quad v \leftarrow g_1 g_2^r$
45 $\quad w \leftarrow g_1^y h^r$
46 $\quad \alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$
47 $\quad e \leftarrow u^{a+\alpha a'} v^{b+\alpha b'}$
48 $\quad \mathbf{c}[j] \leftarrow (u, v, w, e)$
49 $\quad C \leftarrow C \bigcup \{(j, \mathbf{c}[j])\}$
50 $(I, s_2) \leftarrow A_2^{\mathrm{DEC}}(\mathbf{c}, s_1),\ I_{\mathsf{open}} \leftarrow I$
51 For $j \in I$, then
52 $\quad (x, y, a, b, a', b') \leftarrow \mathbf{sk}[j]$
53 $\quad (x', y') \leftarrow (x + \mathbf{m}[j] \cdot z, y - \mathbf{m}[j])$
54 $\quad \mathbf{sk}[j] \leftarrow (x', y', a, b, a', b')$
55 $out \leftarrow A_3^{\mathrm{DEC}}(\mathbf{sk}[I], \mathbf{m}[I], s_2)$
56 Return $(\mathbf{m}, \mathcal{M}, I, out)$

On query $\mathrm{DEC}(j, c')$:
57 If $(j, c') \in C$, then return $\perp$
58 If $j \in I_{\mathsf{open}}$, then return $\perp$
59 $(x, y, a, b, a', b') \leftarrow \mathbf{sk}[j]$
60 $(u, v, w, e) \leftarrow c'$
61 If $u^{\mathbf{tk}[j]} \ne v$, then return $\perp$
62 $\alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$
63 If $u^{a+\alpha a'} v^{b+\alpha b'} \ne e$, then return $\perp$
64 $\delta \leftarrow w \cdot (u^x v^y)^{-1}$
65 If $\delta = 1$, then $m' \leftarrow 0$
66 If $\delta = g_1$, then $m' \leftarrow 1$
67 Else, then $m' \leftarrow \perp$
68 Return m'

Fig. 3. Games $\mathbf{G}_{-1} - \mathbf{G}_{3n}$ in the proof of Theorem 1. Note that lines ending with a range of games $\mathbf{G}_{j_1} - \mathbf{G}_{j_2}$ (resp. $\mathbf{G}_j$) are only executed when a game within the range is run, and for the "$\mathbf{G}_i$, $\mathbf{G}_{n+i}$, $\mathbf{G}_{2n+i}$" above, we require that $1 \le i \le n$.

$$e = u^{a+\alpha a'} v^{b+\alpha b'} = g_1^{r(a+\alpha a')} g_2^{r(b+\alpha b')} = (g_1^a g_2^b \cdot (g_1^{a'} g_2^{b'})^\alpha)^r = (\theta \cdot \varphi^\alpha)^r.$$

Hence, we derive that $\mathbf{G}_0 = \mathbf{G}_{-1}$.

For $i \in [n]$, games $\mathbf{G}_i$ and $\mathbf{G}_{i-1}$ are identical, except for the generation of v *for* $j = i$ (line 09). More specifically, when $j = i$, $v = g_1 g_2^r$ in $\mathbf{G}_i$, and $v = g_2^r$ in $\mathbf{G}_{i-1}$. It is easy to see that for any PPT algorithm D distinguishing $\mathbf{G}_i$ and $\mathbf{G}_{i-1}$, we can construct a PPT distinguisher D_{DDH1} to distinguish $(g_1, g_1^z, g_1^r, g_1^{rz})$ and $(g_1, g_1^z, g_1^r, g_1^{rz+1})$ based on D with almost the same advantage. The construction is very trivial so we omit the details here. According to Lemma 1, we derive that for any PPT algorithm D,

$$\begin{aligned} &|\Pr[D(\mathbf{G}_i) = 1] - \Pr[D(\mathbf{G}_{i-1}) = 1]| \\ \le\ &|\Pr[D_{DDH1}(g_1, g_1^z, g_1^r, g_1^{rz+1}) = 1] - \Pr[D_{DDH1}(g_1, g_1^z, g_1^r, g_1^{rz}) = 1]| \end{aligned}$$

is negligible. Thus, with a standard hybrid argument, we derive that for any PPT algorithm D, $|\Pr[D(\mathbf{G}_n) = 1] - \Pr[D(\mathbf{G}_0) = 1]|$ is negligible.

Now we stress that in games $\mathbf{G}_{n+1} - \mathbf{G}_{3n}$, during the generation of the challenge ciphertext vector, for every $j \in [n]$, the computation of v is $v = g_1 g_2^r$ (line 10).

For $i \in [n]$, let bad denote the event in game $\mathbf{G}_{n+i-1}$ that the adversary submits a decryption query $(j, c' = (u, v, w, e))$ satisfying $((j, c') \notin C) \bigwedge (j \notin I_{\mathsf{open}}) \bigwedge (j = i) \bigwedge (u^{\mathbf{tk}[j]} \neq v) \bigwedge (u^{a+\alpha a'} v^{b+\alpha b'} = e)$ where $\alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$. Games $\mathbf{G}_{n+i}$ and $\mathbf{G}_{n+i-1}$ are identical until bad occurs. According to the fundamental lemma of game-playing [2], for any PPT distinguisher D, $|\Pr[D(\mathbf{G}_{n+i}) = 1] - \Pr[D(\mathbf{G}_{n+i-1}) = 1]| \leq \Pr[\mathsf{bad}]$. With the help of the following lemma, we have that $|\Pr[D(\mathbf{G}_{n+i}) = 1] - \Pr[D(\mathbf{G}_{n+i-1}) = 1]|$ is negligible. Then with a standard hybrid argument, for any PPT algorithm D, $|\Pr[D(\mathbf{G}_{2n}) = 1] - \Pr[D(\mathbf{G}_n) = 1]|$ is negligible.

Lemma 3. $\Pr[\mathsf{bad}]$ *is negligible.*

The proof of this lemma is very similar to that of [11, Claim 2]. For completeness, we provide it in Appendix A.

Now we stress that in games $\mathbf{G}_{2n+1} - \mathbf{G}_{3n}$, for any decryption query $(j, c' = (u, v, w, e))$ satisfying $u'^{\mathbf{tk}[j]} \neq v'$, the decryption oracle always returns $\perp$ (line 29).

For $i \in [n]$, games $\mathbf{G}_{2n+i}$ and $\mathbf{G}_{2n+i-1}$ are identical, except that in the generation of challenge ciphertext vector of $\mathbf{G}_{2n+i}$, when $j = i$, w and $\mathbf{sk}'[j]$ will be updated (lines 13-16). For clarity, we use $\mathbf{sk}[i] = (x, y, a, b, a', b')$ to denote the original i^{th} secret key, and $\mathbf{sk}'[i] = (x + \mathbf{m}[i] \cdot z, y - \mathbf{m}[i], a, b, a', b')$ to denote the corresponding updated secret key. Firstly, note that $\mathbf{sk}'[i]$ is indeed a valid secret key for $\mathbf{pk}[i] = (g_1, g_2, h, \theta, \varphi, hk)$, since the tuple (a, b, a', b') of $\mathbf{sk}'[i]$ is the same as that of the original, valid $\mathbf{sk}[i]$, and

$$g_1^{x+\mathbf{m}[i]\cdot z} g_2^{y-\mathbf{m}[i]} = g_1^{x+\mathbf{m}[i]\cdot z} g_1^{z(y-\mathbf{m}[i])} = g_1^x g_1^{zy} = g_1^x g_2^y = h.$$

Next, note that decrypting the updated $\mathbf{c}[i] = (u, v, w, e)$ where $w = g_1^y h^r$, with the updated secret key $\mathbf{sk}'[i]$, we can still recover $\mathbf{m}[i]$. That's because when decrypting $\mathbf{c}[i] = (u, v, w, e)$, we have that for $\alpha \leftarrow \mathsf{HEvl}(u, v, w)$, $e = u^{a+\alpha a'} v^{b+\alpha b'}$ and

$$\begin{aligned} w \cdot (u^{x+\mathbf{m}[i]\cdot z} v^{y-\mathbf{m}[i]})^{-1} &= g_1^y h^r (g_1^{r(x+\mathbf{m}[i]\cdot z)} (g_1 g_2^r)^{y-\mathbf{m}[i]})^{-1} \\ &= g_1^y g_1^{rx} g_2^{ry} (g_1^{rx+\mathbf{m}[i]\cdot rz} g_1^{y-\mathbf{m}[i]} g_2^{ry-\mathbf{m}[i]\cdot r})^{-1} \\ &= g_1^y g_1^{rx} g_2^{ry} (g_1^{rx} g_2^{\mathbf{m}[i]\cdot r} g_1^{y-\mathbf{m}[i]} g_2^{ry-\mathbf{m}[i]\cdot r})^{-1} \\ &= g_1^{\mathbf{m}[i]}. \end{aligned}$$

Thirdly, note that the only differences between $\mathbf{sk}'[i]$ and $\mathbf{sk}[i]$ are the elements $(x + \mathbf{m}[i] \cdot z, y - \mathbf{m}[i])$ (resp. (x, y)). We also note that the only procedure involving $(x + \mathbf{m}[i] \cdot z, y - \mathbf{m}[i])$ (resp. (x, y)) during the decryption in $\mathbf{G}_{2n+i}$ (resp. $\mathbf{G}_{2n+i-1}$) is the computation of $u^{x+\mathbf{m}[i]\cdot z} v^{y-\mathbf{m}[i]}$ (resp. $u^x v^y$), for any ciphertext (u, v, w, e). For any ciphertext (u, v, w, e) satisfying $\log_{g_1} u = \log_{g_2} v = r$ for some r, we obtain that

$$u^{x+\mathbf{m}[i]\cdot z} v^{y-\mathbf{m}[i]} = g_1^{r(x+\mathbf{m}[i]\cdot z)} g_2^{r(y-\mathbf{m}[i])} = g_1^{rx} g_2^{ry} = u^x v^y.$$

In other words, for any ciphertext (u, v, w, e) satisfying $\log_{g_1} u = \log_{g_2} v$, its decryption with $\mathbf{sk}'[i]$ and that with $\mathbf{sk}[i]$ (i.e., *the response of the decryption oracle*) are identical. Therefore, even receiving $\mathbf{sk}'[i]$ as the response of the opening query (when $i \in I$) in $\mathbf{G}_{2n+i}$, the view of A is the same as that in $\mathbf{G}_{2n+i-1}$, except that A submits some decryption query $(i, (u, v, w, e))$ satisfying $\log_{g_1} u \neq \log_{g_2} v$ before the opening query. On the other hand, both in $\mathbf{G}_{2n+i}$ and $\mathbf{G}_{2n+i-1}$, for any decryption query $(i, (u, v, w, e))$ satisfying $\log_{g_1} u \neq \log_{g_2} v$, we have $u^{\mathbf{tk}[i]} \neq v$, so the

$S_1(1^\lambda)$:
$(\mathbf{pk}, \mathbf{sk}, \mathbf{tk}) \leftarrow (\mathsf{CS.Gen}(1^\lambda))^n$
$C \leftarrow \emptyset$
$I_{\mathsf{open}} \leftarrow \emptyset$
$(\mathcal{M}, s_1) \leftarrow A_1^{\mathrm{DEC}}(\mathbf{pk})$
$\widetilde{s_1} \leftarrow (\mathbf{pk}, \mathbf{sk}, \mathbf{tk}, C, s_1)$
Return $(\mathcal{M}, \widetilde{s_1})$

$S_2((1^{|\mathbf{m}[i]|})_{i\in[n]}, \widetilde{s_1})$:
$(\mathbf{pk}, \mathbf{sk}, \mathbf{tk}, C, s_1) \leftarrow \widetilde{s_1}$
$I_{\mathsf{open}} \leftarrow \emptyset$
For $j = 1$ to n
$(g_1, g_2, h, \theta, \varphi, hk) \leftarrow \mathbf{pk}[j]$
$(x, y, a, b, a', b') \leftarrow \mathbf{sk}[j]$
$r \leftarrow \mathbb{Z}_q,\ u \leftarrow g_1^r$
$v \leftarrow g_1 g_2^r,\ \ w \leftarrow g_1^y h^r$
$\alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$
$e \leftarrow u^{a+\alpha a'} v^{b+\alpha b'}$
$\mathbf{c}[j] \leftarrow (u, v, w, e)$
$C \leftarrow C \bigcup \{(j, \mathbf{c}[j])\}$
$(I, s_2) \leftarrow A_2^{\mathrm{DEC}}(\mathbf{c}, s_1)$
$\widetilde{s_2} \leftarrow (\mathbf{pk}, \mathbf{sk}, \mathbf{tk}, C, I, s_2)$
Return $(I, \widetilde{s_2})$

$S_3(\mathbf{m}[I], \widetilde{s_2})$:
$(\mathbf{pk}, \mathbf{sk}, \mathbf{tk}, C, I, s_2) \leftarrow \widetilde{s_2},\ I_{\mathsf{open}} \leftarrow I$
For $j \in I$, then
$(x, y, a, b, a', b') \leftarrow \mathbf{sk}[j], z \leftarrow \mathbf{tk}[j]$
$(x', y') \leftarrow (x + \mathbf{m}[j] \cdot z, y - \mathbf{m}[j])$
$\mathbf{sk}[j] \leftarrow (x', y', a, b, a', b')$
$out \leftarrow A_3^{\mathrm{DEC}}(\mathbf{sk}[I], \mathbf{m}[I], s_2)$
Return out

On query $\mathrm{DEC}(j, (u, v, w, e))$:
If $(j, c') \in C$, then return $\perp$
If $j \in I_{\mathsf{open}}$, then return $\perp$
$(x, y, a, b, a', b') \leftarrow \mathbf{sk}[j],\ z \leftarrow \mathbf{tk}[j]$
If $u^z \neq v$, then $\mathsf{bad} \leftarrow \mathsf{true}$; return $\perp$
$\alpha \leftarrow \mathsf{HEvl}(hk, (u, v, w))$
If $u^{a+\alpha a'} v^{b+\alpha b'} \neq e$, then return $\perp$
$\delta \leftarrow w \cdot (u^x v^y)^{-1}$
If $\delta = 1$, then return $m' = 0$
If $\delta = g_1$, then return $m' = 1$
Else, return $m' = \perp$

Fig. 4. Simulator $S = (S_1, S_2, S_3)$ in the proof of Theorem 1.

decryption oracle will always return $\perp$. Hence, we obtain the conclusion that $\mathbf{G}_{2n+i} = \mathbf{G}_{2n+i-1}$. Then with a sequence of hybrid games, we obtain $\mathbf{G}_{3n} = \mathbf{G}_{2n}$.

Now we stress that in game $\mathbf{G}_{3n}$, during the generation of the challenge ciphertext vector, for every $j \in [n]$, $w = g_1^y h^r$ and $\mathbf{sk}'[j] = (x + \mathbf{m}[j] \cdot \mathbf{tk}[j], y - \mathbf{m}[j])$.

Note that game $\mathbf{G}_{3n}$ can be written as $\mathbf{G}'_{3n}$, which implies that $\mathbf{G}'_{3n} = \mathbf{G}_{3n}$. Therefore, a PPT simulator S for A can be constructed as shown in Fig. 4. S simulates $\mathbf{G}'_{3n}$ for A perfectly, so we derive that $\mathsf{Exp}^{\text{rsim-so-cca-ideal}}_{\mathsf{PKE}_{\mathsf{CS}},A,S} = \mathbf{G}'_{3n}$, which concludes this proof. □

Remark 2. As pointed out in [10], the aforementioned scheme can be extended to support any polynomial-size message space, i.e., $m = \log_{g_1}(w \cdot (u^x v^y)^{-1})$ can be determined efficiently.

4.2 RSIM-SO-CCA secure PKE under the DCR assumption

We show that the PKE scheme under the DCR assumption, proposed by Cramer and Shoup in [12], meets RSIM-SO-CCA security. We note that this scheme supports exponential-size message space.

Let $\mathsf{Hash} = (\mathsf{HGen}, \mathsf{HEvl})$ be a family of collision-resistant (CR) hash functions $\Lambda_{(\cdot)} : (\mathbb{Z}^*_{N^2})^2 \to \{0, 1, \cdots, 2^\lambda - 1\}$. The DCR-based PKE scheme [12] $\mathsf{PKE}_{\mathsf{PaCS}} := (\mathsf{PaCS.Gen}, \mathsf{PaCS.Enc}, \mathsf{PaCS.Dec})$ is shown in Fig. 5. For public key $pk = (N, g, h, \theta_0, \theta_1, hk)$, the corresponding message space is $\mathbb{Z}_N$. We refer the readers to [12] for the correctness of $\mathsf{PKE}_{\mathsf{PaCS}}$. Again, here we set a trapdoor key tk which is N' as above (see Fig. 5). The additional trapdoor key is useless for normal encryption/decryption, but will be used in the following security proof.

Formally, we have the following theorem. The proof is inspired by that of Hazay et al. [18].

Theorem 2. $\mathsf{PKE}_{\mathsf{PaCS}}$ *is RSIM-SO-CCA secure.*

$\mathsf{PaCS.Gen}(1^\lambda)$:	$\mathsf{PaCS.Enc}(pk, m)$:	$\mathsf{PaCS.Dec}(sk, (u, e, v))$:
$(N, N') \leftarrow \mathcal{G}(1^\lambda)$	$(N, g, h, \theta_0, \theta_1, hk) \leftarrow pk$	$(x, y_0, y_1) \leftarrow sk$
$x, y_0, y_1 \leftarrow \mathbb{Z}_{\lfloor N^2/4 \rfloor}$	$r \leftarrow \mathbb{Z}_{\lfloor N/4 \rfloor}$	$\alpha \leftarrow \mathsf{HEvl}(hk, (u, e))$
$g' \leftarrow \mathbb{Z}^*_{N^2},\ g \leftarrow g'^{2N}$	$u \leftarrow g^r$	If $u^{y_0+y_1\alpha} \neq v$, then return $\perp$
$h \leftarrow g^x,\ \theta_0 \leftarrow g^{y_0},\ \theta_1 \leftarrow g^{y_1}$	$e \leftarrow (1+N)^m h^r$	$M \leftarrow (e/(u^x))^{N+1}$
$hk \leftarrow \mathsf{HGen}(1^\lambda)$	$\alpha \leftarrow \mathsf{HEvl}(hk, (u, e))$	If $M = 1 + mN$, then return m
$pk \leftarrow (N, g, h, \theta_0, \theta_1, hk)$	$v \leftarrow (\theta_0 \theta_1^\alpha)^r$	Else, return $\perp$
$sk \leftarrow (x, y_0, y_1)$	Return (u, e, v)	
$tk \leftarrow N'$		
Return (pk, sk, tk)		

Fig. 5. Construction of $\mathsf{PKE}_{\mathsf{PaCS}}$.

Proof. Let CRTM denote a PPT algorithm which takes $a_0, N_0, a_1, N_1 \in \mathbb{N}$ as input, where N_0, N_1 are relatively prime, and employs the Chinese Remainder Theorem to return $x \in \mathbb{Z}_{N_0 N_1}$ such that $x = a_0 \bmod N_0$, and $x = a_1 \bmod N_1$.

For any PPT adversary A attacking $\mathsf{PKE}_{\mathsf{PaCS}}$ in the sense of RSIM-SO-CCA, let q_d denote the number of decryption queries made by A. We prove the theorem with a sequence of games $\mathbf{G}_{-1} - \mathbf{G}_{3n}, \mathbf{G}'_{3n}$ in Fig. 6.

Firstly, note that the view of A in game $\mathbf{G}_{-1}$ is exactly the same as in $\mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE}_{\mathsf{PaCS}},A}$, so the final outputs of these two games are identical, i.e., $\mathbf{G}_{-1} = \mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE}_{\mathsf{PaCS}},A}$.

In game $\mathbf{G}_0$, lines 06, 18 are added. $\mathbf{G}_0$ and $\mathbf{G}_{-1}$ are identical except that the challenge ciphertext vector is encrypted with secret key $\mathbf{sk}$. We stress that the view of A in $\mathbf{G}_0$ is exactly the same as in $\mathbf{G}_{-1}$. That's because for all $j \in [n]$, $\mathbf{sk}[j] = (x, y_0, y_1)$, $\mathbf{pk}[j] = (N, g, h, \theta_0, \theta_1, hk)$, we have

$$v = u^{y_0+y_1\alpha} = g^{r(y_0+y_1\alpha)} = (g^{y_0} g^{y_1\alpha})^r = (\theta_0 \theta_1^\alpha)^r.$$

Hence, we derive that $\mathbf{G}_0 = \mathbf{G}_{-1}$.

For $i \in [n]$, games $\mathbf{G}_i$ and $\mathbf{G}_{i-1}$ are identical, except for the generation of u *for* $j = i$ (line 09). More specifically, when $j = i$, $u = (1+N)g^r = (1+N)(g'^{2N})^r$ in $\mathbf{G}_i$, and $u = g^r = (g'^{2N})^r$ in $\mathbf{G}_{i-1}$. Since $g' \leftarrow \mathbb{Z}^*_{N^2} = \mathbb{G}_N \cdot \mathbb{G}_{N'} \cdot \mathbb{G}_2 \cdot \mathbb{S}$, g'^{2N} is uniformly distributed over $\mathbb{G}_{N'}$, and with overwhelming probability the order of g'^{2N} is N'. Note that the uniform distributions over $\mathbb{Z}_{\lfloor N/4 \rfloor}$ and $\mathbb{Z}_{N'}$ are statistically indistinguishable. Therefore, the distribution of the above $(g'^{2N})^r$ and the uniform distribution over $\mathbb{G}_{N'}$ are also statistically indistinguishable. In other words, the distribution of $(g'^{2N})^r$ is statistically indistinguishable from that of $\widetilde{g}^{2N}$, where $\widetilde{g} \leftarrow \mathbb{Z}^*_{N^2}$. Hence, if there is some PPT algorithm D distinguishing $\mathbf{G}_i$ and $\mathbf{G}_{i-1}$ with non-negligible advantage, then we can construct a PPT distinguisher D_{DCR1} to distinguish $(N, (1+N)\widetilde{g}^{2N})$ and $(N, \widetilde{g}^{2N})$ based on D with non-negligible advantage, contradicting Lemma 2. The construction is very trivial so we omit the details here. Thus, with a standard hybrid argument, we derive that for any PPT algorithm D, $|\Pr[D(\mathbf{G}_n) = 1] - \Pr[D(\mathbf{G}_0) = 1]|$ is negligible.

Now we stress that in games $\mathbf{G}_{n+1} - \mathbf{G}_{3n}$, during the generation of the challenge ciphertext vector, for every $j \in [n]$, the computation of u is $u = (1+N)g^r$ (line 10).

For $i \in [n]$, let bad denote the event that in game $\mathbf{G}_{n+i-1}$ the adversary submits a decryption query $(j, c' = (u, e, v))$ satisfying $((j, c') \notin C) \bigwedge (j \notin I_{\mathsf{open}}) \bigwedge (j = i) \bigwedge (u^{2\mathbf{tk}[j]} \neq 1) \bigwedge (u^{y_0+y_1\alpha} = v)$ where $\alpha \leftarrow \mathsf{HEvl}(hk, (u, e))$. Games $\mathbf{G}_{n+i}$ and $\mathbf{G}_{n+i-1}$ are identical until bad occurs. Hence, for any PPT distinguisher D, $|\Pr[D(\mathbf{G}_{n+i}) = 1] - \Pr[D(\mathbf{G}_{n+i-1}) = 1]| \leq \Pr[\mathsf{bad}]$. With the help of the following lemma, we have $|\Pr[D(\mathbf{G}_{n+i}) = 1] - \Pr[D(\mathbf{G}_{n+i-1}) = 1]|$ is negligible. Then with

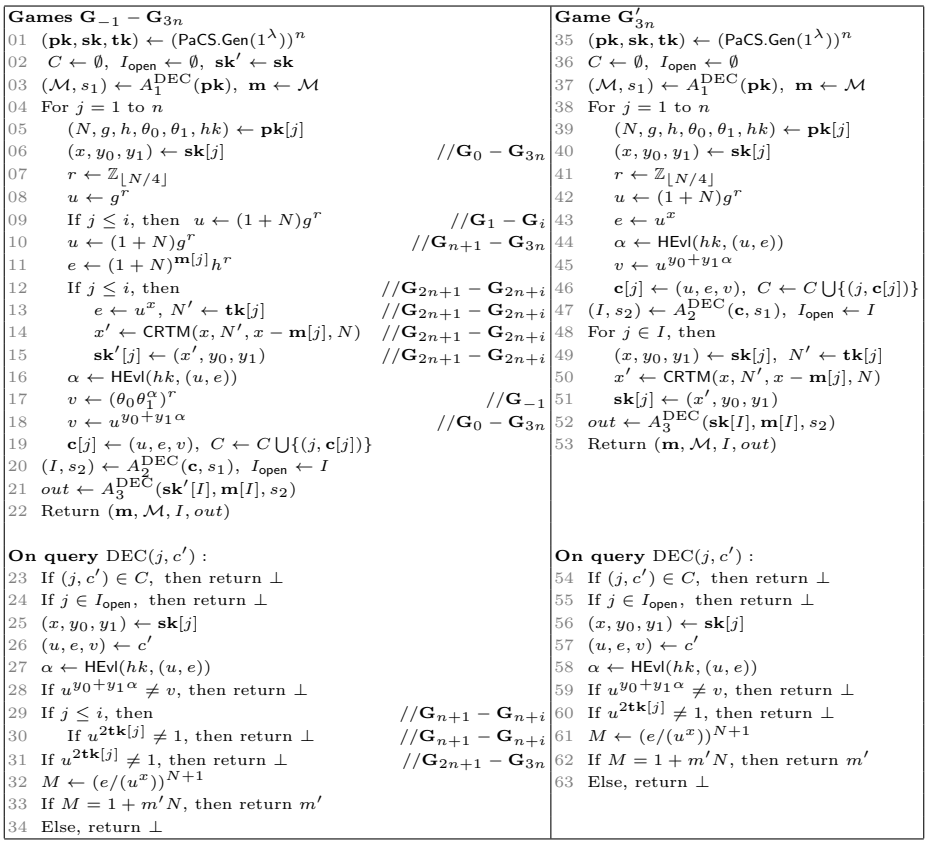

Fig. 6. Games $\mathbf{G}_{-1} - \mathbf{G}_{3n}$ in the proof of Theorem 2. Note that lines ending with a range of games $\mathbf{G}_{j_1} - \mathbf{G}_{j_2}$ (resp. $\mathbf{G}_j$) are only executed when a game within the range is run, and for the "$\mathbf{G}_i$, $\mathbf{G}_{n+i}$, $\mathbf{G}_{2n+i}$" above, we require that $1 \le i \le n$.

a standard hybrid argument, for any PPT algorithm D, $|\Pr[D(\mathbf{G}_{2n}) = 1] - \Pr[D(\mathbf{G}_n) = 1]|$ is negligible.

Lemma 4. $\Pr[\mathsf{bad}]$ *is negligible.*

Now we stress that in games $\mathbf{G}_{2n+1} - \mathbf{G}_{3n}$, for any decryption query $(j, c' = (u, e, v))$ satisfying $u^{2\mathbf{tk}[j]} \neq 1$, the decryption oracle always returns $\perp$ (line 31).

For $i \in [n]$, games $\mathbf{G}_{2n+i}$ and $\mathbf{G}_{2n+i-1}$ are identical, except that in the generation of challenge ciphertext vector of $\mathbf{G}_{2n+i}$, when $j = i$, e and $\mathbf{sk}'[j]$ will be updated (lines 12-15). For clarity, we use $\mathbf{sk}[i] = (x, y_0, y_1)$ to denote the original i^{th} secret key, and $\mathbf{sk}'[i] = (x', y_0, y_1)$ to denote the corresponding updated secret key. Firstly, note that $\mathbf{sk}'[i]$ is indeed a valid secret key for $\mathbf{pk}[i] = (N, g, h, \theta_0, \theta_1, hk)$. The reasons are as follows: (i) the tuple (y_0, y_1) of $\mathbf{sk}'[i]$ is the same as that of the original, valid $\mathbf{sk}[i]$; (ii) $g^{x'} = h = g^x$, since $g = g'^{2N} \in \mathbb{G}_{N'}$, and the Chinese Remainder Theorem guarantees that $x' = x \bmod N'$. Next, note that decrypting the updated $\mathbf{c}[i] = (u, e, v)$, where $e = u^x$, with the updated secret key $\mathbf{sk}'[i]$, we can still recover $\mathbf{m}[i]$. That's because when decrypting $\mathbf{c}[i] = (u, e, v)$, we have that for $\alpha \leftarrow \mathsf{HEvl}(u, e)$,

$$(e/(u^{x'}))^{N+1} = \left(\frac{u^x}{u^{x'}}\right)^{N+1} = \left(\frac{(1+N)^x \cdot g^{rx}}{(1+N)^{x'} \cdot g^{rx'}}\right)^{N+1} = \left(\frac{(1+N)^x \cdot h^r}{(1+N)^{x'} \cdot h^r}\right)^{N+1}$$

$$= (1+N)^{(x-x')(N+1)} = (1+N)^{x-x'} = (1+N)^{\mathbf{m}[i]}$$
$$= 1 + \mathbf{m}[i]N.$$

Thirdly, note that the only differences between $\mathbf{sk}'[i]$ and $\mathbf{sk}[i]$ are the elements x' (resp. x). We also note that the only procedure involving x' (resp. x) during the decryption in $\mathbf{G}_{2n+i}$ (resp. $\mathbf{G}_{2n+i-1}$) is the computation of $(u^{x'})^{N+1}$ (resp. $(u^x)^{N+1}$), for any ciphertext (u, e, v). Recall that $\mathbf{tk}[i] = N'$. For any ciphertext (u, e, v) satisfying $u^{2\mathbf{tk}[i]} = u^{2N'} = 1$, we claim that $(u^{x'})^{N+1} = (u^x)^{N+1}$. The reason is as follows. The fact $u^{2N'} = 1$ implies that the order of u divides $2N'$. Thus, there are three possibilities.

(1) If the order of u is 1, then obviously we have $(u^{x'})^{N+1} = (u^x)^{N+1}$.
(2) If the order of u is 2, then $u^{N+1} = 1$ because N is odd. Thus, we still have $(u^{x'})^{N+1} = (u^x)^{N+1}$.
(3) If the order of u is $\eta \in \{p', q', N'\}$, then we derive that $\eta \mid (x - x')$ (i.e., $x = x' \bmod \eta$), since $x = x' \bmod N'$ and $N' = p'q'$. Hence, $(u^{x'})^{N+1} = (u^x)^{N+1}$.

In other words, for any ciphertext (u, e, v) satisfying $u^{2\mathbf{tk}[i]} = 1$, its decryption with $\mathbf{sk}'[i]$ and that with $\mathbf{sk}[i]$ (i.e., *the response of the decryption oracle*) are identical. Therefore, even receiving $\mathbf{sk}'[i]$ as the response of the opening query (when $i \in I$) in $\mathbf{G}_{2n+i}$, the view of A is the same as that in $\mathbf{G}_{2n+i-1}$, except that A submits some decryption query $(i, (u, e, v))$ satisfying $u^{2\mathbf{tk}[i]} \neq 1$ before the opening query. On the other hand, both in $\mathbf{G}_{2n+i}$ and $\mathbf{G}_{2n+i-1}$, for any decryption query $(i, (u, e, v))$ satisfying $u^{2\mathbf{tk}[i]} \neq 1$, the decryption oracle always returns $\perp$ (line 31). Hence, we have the conclusion that $\mathbf{G}_{2n+i} = \mathbf{G}_{2n+i-1}$. Then with a sequence of hybrid games, we obtain $\mathbf{G}_{3n} = \mathbf{G}_{2n}$.

Now we stress that in game $\mathbf{G}_{3n}$, during the generation of the challenge ciphertext vector, for every $j \in [n]$, $e = u^x$ and $\mathbf{sk}'[j] = (x', y_0, y_1)$.

Note that game $\mathbf{G}_{3n}$ can be written as $\mathbf{G}'_{3n}$, which implies that $\mathbf{G}'_{3n} = \mathbf{G}_{3n}$. Therefore, a PPT simulator S for A can be constructed as shown in Fig. 7. S simulates $\mathbf{G}'_{3n}$ for A perfectly, so we derive that $\mathsf{Exp}^{\text{rsim-so-cca-ideal}}_{\mathsf{PKE}_{\mathsf{PaCS}},A,S} = \mathbf{G}'_{3n}$, which concludes this proof.

We catch up with the proof of Lemma 4.

Proof (of Lemma 4). Parse $\mathbf{pk}[i] = (N, g, h, \theta_0, \theta_1, hk)$, $\mathbf{sk}[i] = (x, y_0, t_1)$ and $\mathbf{tk}[i] = N'$. Denote the i^{th} challenge ciphertext in $\mathbf{G}_{n+i-1}$ by $\mathbf{c}[i] = (u, e, v)$.

Game $\mathbf{G}_{n+i-1}$ sets bad iff adversary A, without obtaining $\mathbf{sk}[i]$ by opening query, submits a decryption query $(i, c' = (u', e', v'))$ such that $(i, c') \notin C$, $u'^{2N'} \neq 1$, and $u'^{y_0 + y_1\alpha'} = v'$, where $\alpha' = \mathsf{HEvl}(hk, (u', e'))$. There are four possible cases.

Case 1. $(u', e', v') = (u, e, v)$.

Since $(i, c') \notin C$, this case occurs only when A submits such a decryption query *before* receiving $\mathbf{c}$. In both $\mathbf{G}_{n+i}$ and $\mathbf{G}_{n+i-1}$, $u = (1+N)g^r = (1+N)(g'^{2N})^r$. As mentioned earlier, the distribution of $(g'^{2N})^r$ and the uniform distribution over $\mathbb{G}_{N'}$ are statistically indistinguishable. Hence, the distribution of u is statistically indistinguishable from the uniform distribution over $(1+N)\cdot\mathbb{G}_{N'}$. Notice that A makes at most q_d decryption queries, and that the best circumstance for A is that each decryption query will help A to eliminate one possible value of u. Hence, the probability that Case 1 occurs is at most $\frac{q_d}{N'-q_d}$, which is negligible.

We stress that this is also a loose bound. That's because in both $\mathbf{G}_{n+i}$ and $\mathbf{G}_{n+i-1}$, $e = (1+N)^{\mathbf{m}[i]}h^r$ is computed with $\mathbf{m}[i]$, which A does not know, and v is computed with $\alpha =$

$S_1(1^\lambda)$:
$(\mathbf{pk}, \mathbf{sk}, \mathbf{tk}) \leftarrow (\mathsf{PaCS.Gen}(1^\lambda))^n$
$C \leftarrow \emptyset$
$I_{\mathsf{open}} \leftarrow \emptyset$
$(\mathcal{M}, s_1) \leftarrow A_1^{\mathrm{DEC}}(\mathbf{pk})$
$\widetilde{s_1} \leftarrow (\mathbf{pk}, \mathbf{sk}, \mathbf{tk}, C, s_1)$
Return $(\mathcal{M}, \widetilde{s_1})$

$S_2((1^{|\mathbf{m}[i]|})_{i\in[n]}, \widetilde{s_1})$:
$(\mathbf{pk}, \mathbf{sk}, \mathbf{tk}, C, s_1) \leftarrow \widetilde{s_1}$
$I_{\mathsf{open}} \leftarrow \emptyset$
For $j = 1$ to n
  $(N, g, h, \theta_0, \theta_1, hk) \leftarrow \mathbf{pk}[j]$
  $(x, y_0, y_1) \leftarrow \mathbf{sk}[j]$
  $r \leftarrow \mathbb{Z}_{\lfloor N/4 \rfloor}$
  $u \leftarrow (1+N)g^r$
  $e \leftarrow u^x$
  $\alpha \leftarrow \mathsf{HEvl}(hk, (u, e))$
  $v \leftarrow u^{y_0 + y_1\alpha}$
  $\mathbf{c}[j] \leftarrow (u, e, v)$
  $C \leftarrow C \bigcup \{(j, \mathbf{c}[j])\}$
$(I, s_2) \leftarrow A_2^{\mathrm{DEC}}(\mathbf{c}, s_1)$
$\widetilde{s_2} \leftarrow (\mathbf{pk}, \mathbf{sk}, \mathbf{tk}, C, I, s_2)$
Return $(I, \widetilde{s_2})$

$S_3(\mathbf{m}[I], \widetilde{s_2})$:
$(\mathbf{pk}, \mathbf{sk}, \mathbf{tk}, C, I, s_2) \leftarrow \widetilde{s_2}$
$I_{\mathsf{open}} \leftarrow I$
For $j \in I$, then
  $(x, y_0, y_1) \leftarrow \mathbf{sk}[j]$
  $x' \leftarrow \mathsf{CRTM}(x, \mathbf{tk}[j], x - \mathbf{m}[j], N)$
  $\mathbf{sk}[j] \leftarrow (x', y_0, y_1)$
$out \leftarrow A_3^{\mathrm{DEC}}(\mathbf{sk}[I], \mathbf{m}[I], s_2)$
Return out

On query $\mathrm{DEC}(j, (u, e, v))$:
If $(j, c') \in C$, then return $\perp$
If $j \in I_{\mathsf{open}}$, then return $\perp$
$(x, y_0, y_1) \leftarrow \mathbf{sk}[j]$, $(u, e, v) \leftarrow c'$
$\alpha \leftarrow \mathsf{HEvl}(hk, (u, e))$
If $u^{y_0 + y_1\alpha} \neq v$, then return $\perp$
If $u^{2\mathbf{tk}[j]} \neq 1$, then return $\perp$
$M \leftarrow (e/(u^x))^{N+1}$
If $M = 1 + m'N$, then return m'
Else, return $\perp$

Fig. 7. Simulator $S = (S_1, S_2, S_3)$ in the proof of Theorem 2.

$\mathsf{HEvl}(hk, (u, e))$. Thus, the probability that A generates $\mathbf{c}[i]$ beforehand is less than $\frac{q_d}{N' - q_d}$.

Case 2. $(u', e') = (u, e)$ and $v' \neq v$.
In this case, $\alpha' = \alpha$ and $u'^{y_0 + y_1\alpha'} = u^{y_0 + y_1\alpha} = v \neq v'$. Hence, $\mathbf{G}_{n+i-1}$ will not set bad.

Case 3. $(u', e') \neq (u, e)$ and $\alpha' = \alpha$.
Since Hash is a CR hash function, Case 3 occurs with negligible probability.

Case 4. $(u', e') \neq (u, e)$ and $\alpha' \neq \alpha$.
Since $\mathbb{Z}_{N^2}^*$ can be decomposed as an inner direct product $\mathbb{G}_N \cdot \mathbb{G}_{N'} \cdot \mathbb{G}_2 \cdot \mathbb{S}$, for any $a \in \mathbb{Z}_{N^2}^*$, a can be uniquely written as $a = a_{(N)}a_{(N')}a_{(2)}a_{(S)}$, where $a_{(i)} \in \mathbb{G}_i$ and $a_{(S)} \in \mathbb{S}$.

From the public key $\mathbf{pk}$ and the challenge ciphertext vector $\mathbf{c}$, for $\mathbf{sk}[i] = (x, y_0, y_1)$, all the information about (y_0, y_1) that A learns is: (i) $\theta_0 = g^{y_0}$; (ii) $\theta_1 = g^{y_1}$; (iii) $v = u^{y_0 + y_1\alpha} = (1+N)^{y_0 + y_1\alpha}(g^r)^{y_0 + y_1\alpha}$. Since the order of g (resp., $1+N$) is $\eta' \in \{p', q', N'\}$ (resp., N), all A learns from (i)-(iii) is actually

$$\log_g \theta_0 = y_0 \bmod \eta', \tag{1}$$

$$\log_g \theta_1 = y_1 \bmod \eta', \tag{2}$$

$$\log_{1+N}(v/(g^r)^{y_0 + y_1\alpha \bmod N'}) = y_0 + y_1\alpha \bmod N. \tag{3}$$

And for the decryption query $(i, (u', e', v'))$, $\mathbf{G}_{n+i-1}$ sets bad only if $u'^{y_0 + y_1\alpha'} = v'$ and $u'^{2N'} \neq 1$. Notice that $u'^{y_0 + y_1\alpha'} = v'$ implies $(u'^{2N'})^{y_0 + y_1\alpha'} = v'^{2N'}$, and $u'^{2N'} \neq 1$ implies

$$(u'_{(N)})^{2N'} = (u'_{(N)}u'_{(N')}u'_{(2)}u'_{(S)})^{2N'} = u'^{2N'} \neq 1.$$

Denote the order of $(u'_{(N)})^{2N'}$ by η. Then we derive $\eta \in \{p, q, N\}$, and

$$\log_{(u'_{(N)})^{2N'}} v'^{2N'} = y_0 + y_1\alpha' \bmod \eta. \tag{4}$$

Since $\eta \mid N$, equation (3) implies that

$$\log_{1+N}(v/(g^r)^{y_0+y_1\alpha \bmod N'}) \equiv y_0 + y_1\alpha \pmod{\eta}. \quad (5)$$

Because N and N' are relatively prime, for every $b \in \{0,1\}$, $y_b \bmod \eta$ and $y_b \bmod \eta'$ are independent. So equations (1)-(2) provide no information about $y_0 \bmod \eta$ or $y_1 \bmod \eta$. Equations (3) (resp. equation (5)) and (4) are linearly independent since $\alpha' \neq \alpha$. So we have the following conclusions:

(i) If $\eta = N$, equations (3)-(4) uniquely determine $y_0, y_1 \in \mathbb{Z}_N$. Hence, the probability that A submits a decryption query $(i, (u', e', v'))$ where $u'^{2N'} \neq 1$ *for the first time*, such that $\mathbf{G}_{n+i-1}$ sets bad, is $\frac{1}{N}$. At best, each decryption query will help the adversary to eliminate one possible value. Therefore, when $\eta = N$, the possibility that $\mathbf{G}_{n+i-1}$ sets bad in Case 4 is at most $\frac{q_d}{N-q_d}$, which is negligible.

(ii) If $\eta \neq N$, then $\eta = p$ or q. Equations (5)-(4) uniquely determine $y_0, y_1 \in \mathbb{Z}_\eta$. Hence, the probability that A submits a decryption query $(i, (u', e', v'))$ where $u'^{2N'} \neq 1$ *for the first time*, such that $\mathbf{G}_{n+i-1}$ sets bad, is $\frac{1}{\eta}$. At best, each decryption query will help the adversary to eliminate one possible value. Therefore, when $\eta \neq N$, the possibility that $\mathbf{G}_{n+i-1}$ sets bad in Case 4 is at most $\frac{q_d}{\eta-q_d}$, which is negligible.

□

5 Generic construction of RSIM-SO-CCA secure PKE

In this section, we firstly introduce the notion of master-key selective opening security for IBE. Then based on this notion, we extend the Canetti-Halevi-Katz (CHK) transformation [9] to show generic constructions achieving RSIM-SO-CCA security. Finally, we show how to construct an IBE scheme meeting master-key SOA security.

5.1 The notions of master-key SOA security for IBE

Roughly speaking, in a master-key selective opening attack for IBE, after seeing the challenge ciphertexts, the adversary can corrupt *some of the private key generators*, by obtaining their master secret keys. The goal of master-key SOA security is to guarantee that the messages of the users, whose private key generators are uncorrupted, are still confidential.

Master-key SOA security for IBE. We formalize the notion of master-key selective opening security (mSOA security) for IBE as follows. More specifically, we present the notion of mSOA security under selective-identity, chosen-plaintext attacks. This security notion is sufficient for our RSIM-SO-CCA secure PKE constructions.

Definition 4 (SIM-sID-mSO-CPA). *An IBE scheme* $\mathsf{IBE} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* SIM-sID-mSO-CPA secure*, if for any polynomially bounded $n > 0$, any PPT adversary A, there is a PPT simulator S, such that for any PPT distinguisher D, the advantage*

$$\mathbf{Adv}^{\text{sim-sid-mso-cpa}}_{\mathsf{IBE},A,S,D}(\lambda) := |\Pr[D(\mathsf{Exp}^{\text{sim-sid-mso-cpa-real}}_{\mathsf{IBE},A}(\lambda)) = 1] - \Pr[D(\mathsf{Exp}^{\text{sim-sid-mso-cpa-ideal}}_{\mathsf{IBE},A,S}(\lambda)) = 1]|$$

is negligible, where $\mathsf{Exp}^{\text{sim-sid-mso-cpa-real}}_{\mathsf{IBE},A}(\lambda)$ *and* $\mathsf{Exp}^{\text{sim-sid-mso-cpa-ideal}}_{\mathsf{IBE},A,S}(\lambda)$ *are defined in Fig. 8.*

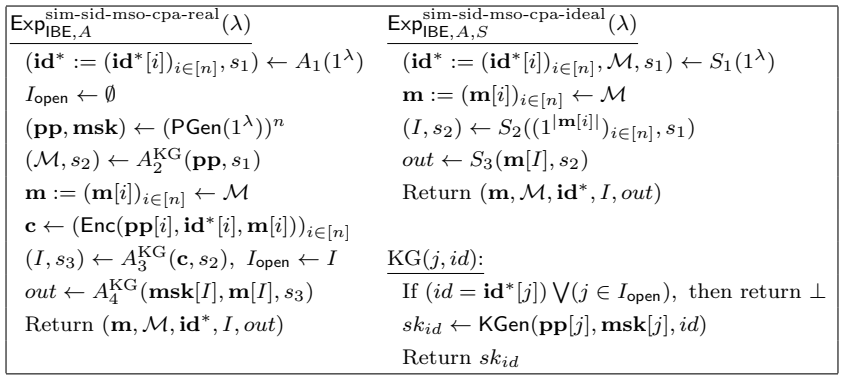

Fig. 8. Experiments for defining SIM-sID-mSO-CPA security of an IBE scheme IBE.

Remark 3. As pointed out by Bellare et al. [3], the standard security notions for IBE naturally provide security against *non-adaptive* receiver corruption (i.e., the adversaries are able to obtain the secret key of any identity *which is not equal to the challenge identity*, by querying the private key generation oracle $\mathrm{KG}(\cdot)$). However, to the best of our knowledge, SOA security notions for IBE in the *adaptive* receiver corruption setting (i.e., the adversaries are able to obtain the secret key of any identity, *including the challenge identity*, after receiving the challenge ciphertext vectors) have never been formalized, and are less studied. Our mSOA security notions focus on SOA security in the adaptive receiver corruption setting. Furthermore, we stress that in the experiments defining mSOA security, the adversaries are actually *more powerful*. They are allowed to corrupt the *private key generators*, *not the receivers (users)*, even *after* seeing the challenge ciphertexts. In other words, an mSOA adversary can obtain *the master secret keys* corresponding to the opened ciphertexts, and an adversary, in the experiments for some "general" SOA security for IBE in the receiver corruption setting, can only obtain *the user secret keys*, which the mSOA adversary can also obtain by running algorithm KGen. Hence, our mSOA security notions are strictly stronger than the "general" SOA security for IBE in the receiver corruption setting.

Remark 4. The notion of indistinguishability-based mSOA security for IBE, and the notions of mSOA security under fully identity or/and chosen-ciphertext attacks can be similarly defined.

5.2 RSIM-SO-CCA secure PKE from IBE

We show a generic construction of RSIM-SO-CCA secure PKE schemes, by applying the CHK method [9] to any SIM-sID-mSO-CPA secure IBE scheme. Note that the CHK method does not work in the sender corruption setting [19]. Our result shows that it does work in the receiver corruption setting.

Generic construction. Let $\mathsf{IBE} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE scheme, and $\mathsf{SIG} = (\mathsf{SGen}, \mathsf{Sign}, \mathsf{Verf})$ be a signature scheme. The PKE scheme $\mathsf{PKE}_{\mathsf{CHK}} = (\mathsf{Gen}_{\mathsf{CHK}}, \mathsf{Enc}_{\mathsf{CHK}}, \mathsf{Dec}_{\mathsf{CHK}})$, constructed according to the well-known CHK transformation [9], is shown in Fig. 9. Verifying correctness is trivial. We turn to security analysis.

Theorem 3. *If* IBE *is SIM-sID-mSO-CPA secure, and* SIG *is strong one-time, then* $\mathsf{PKE}_{\mathsf{CHK}}$ *is RSIM-SO-CCA secure.*

$\mathsf{Gen}_{\mathsf{CHK}}(1^\lambda)$:	$\mathsf{Enc}_{\mathsf{CHK}}(pk, m)$:	$\mathsf{Dec}_{\mathsf{CHK}}(sk, (vk_{\mathsf{s}}, c, sg))$:
$(pp, msk) \leftarrow \mathsf{PGen}(1^\lambda)$	$pp \leftarrow pk$	$msk \leftarrow sk$
$pk \leftarrow pp$	$(sk_{\mathsf{s}}, vk_{\mathsf{s}}) \leftarrow \mathsf{SGen}(1^\lambda)$	If $\mathsf{Verf}(vk_{\mathsf{s}}, c, sg) \neq 1$, then return $\perp$
$sk \leftarrow msk$	$c \leftarrow \mathsf{Enc}(pp, vk_{\mathsf{s}}, m)$	$sk_{vk_{\mathsf{s}}} \leftarrow \mathsf{KGen}(pp, msk, vk_{\mathsf{s}})$
Return (pk, sk)	$sg \leftarrow \mathsf{Sign}(sk_{\mathsf{s}}, c)$	$m \leftarrow \mathsf{Dec}(pp, sk_{vk_{\mathsf{s}}}, c)$
	Return (vk_{s}, c, sg)	Return m

Fig. 9. Construction of $\mathsf{PKE}_{\mathsf{CHK}}$.

Proof. For any PPT adversary $A = (A_1, A_2, A_3)$ attacking $\mathsf{PKE}_{\mathsf{CHK}}$ in the sense of RSIM-SO-CCA, we construct a PPT adversary $A' = (A'_1, A'_2, A'_3, A'_4)$, attacking IBE in the sense of SIM-sID-mSO-CPA, as shown in Fig. 10.

In Fig. 10, let bad denote the event that A submits a decryption $(j, (vk', c', sg')) \notin C_{\mathsf{CHK}}$ satisfying $j \notin I_{\mathsf{open}}$, $vk' = \mathbf{vk}_{\mathsf{s}}[j]$ and $\mathsf{Verf}(vk', (c', sg')) = 1$. Note that $(c', sg') \neq (\mathbf{c}[j], \mathbf{sg}[j])$ since $(j, (vk', c', sg')) \notin C_{\mathsf{CHK}}$. Hence, it's easy to show a PPT adversary A_{sig}, based on A, can break the underlying strong one-time signature SIG with advantage $\frac{1}{n}\Pr[\mathsf{bad}]$. Strong one-time unforgeability of SIG guarantees $\Pr[\mathsf{bad}]$ is negligible.

Denote the final output of $\mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE}_{\mathsf{CHK}},A}$ by $(\mathbf{m}_A, \mathcal{M}_A, I_A, out_A)$, and the final output of $\mathsf{Exp}^{\text{sim-sid-mso-cpa-real}}_{\mathsf{IBE},A'}$ by $(\mathbf{m}_{A'}, \mathcal{M}_{A'}, \mathbf{vk}_{\mathsf{s}}, I_{A'}, out_{A'})$. Since A' perfectly simulates $\mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE}_{\mathsf{CHK}},A}$ for A unless event bad occurs, we derive that $\Pr[(\mathbf{m}_{A'}, \mathcal{M}_{A'}, I_{A'}, out_{A'}) = (\mathbf{m}_A, \mathcal{M}_A, I_A, out_A)] = 1 - \Pr[\mathsf{bad}]$. In other words, for any PPT distinguisher D,

$$|\Pr[D(\mathbf{m}_{A'}, \mathcal{M}_{A'}, I_{A'}, out_{A'}) = 1] - \Pr[D(\mathbf{m}_A, \mathcal{M}_A, I_A, out_A) = 1]| = n\mathbf{Adv}^{\text{str-ot}}_{\mathsf{SIG},A_{sig}}(\lambda), \quad (6)$$

where $\mathbf{Adv}^{\text{str-ot}}_{\mathsf{SIG},A_{sig}}(\lambda)$ denotes the advantage of A_{sig} attacking SIG (see Appendix C).

Considering that IBE is SIM-sID-mSO-CPA secure, there is a PPT simulator $S' = (S'_1, S'_2, S'_3)$, such that the final output of $\mathsf{Exp}^{\text{sim-sid-mso-cpa-ideal}}_{\mathsf{IBE},A',S'}$ (denoted by $(\mathbf{m}_{S'}, \mathcal{M}_{S'}, \mathbf{id}^*, I_{S'}, out_{S'})$) is computationally indistinguishable from $(\mathbf{m}_{A'}, \mathcal{M}_{A'}, \mathbf{vk}_{\mathsf{s}}, I_{A'}, out_{A'})$, i.e., for any PPT distinguisher D,

$$\begin{aligned}&|\Pr[D(\mathbf{m}_{S'}, \mathcal{M}_{S'}, \mathbf{id}^*, I_{S'}, out_{S'}) = 1] \\ &\quad - \Pr[D(\mathbf{m}_{A'}, \mathcal{M}_{A'}, \mathbf{vk}_{\mathsf{s}}, I_{A'}, out_{A'}) = 1]| \leq \mathbf{Adv}^{\text{sim-sid-mso-cpa}}_{\mathsf{IBE},A',S',D}(\lambda).\end{aligned}$$

Therefore,

$$\begin{aligned}&|\Pr[D(\mathbf{m}_{S'}, \mathcal{M}_{S'}, I_{S'}, out_{S'}) = 1] \\ &\quad - \Pr[D(\mathbf{m}_{A'}, \mathcal{M}_{A'}, I_{A'}, out_{A'}) = 1]| \leq \mathbf{Adv}^{\text{sim-sid-mso-cpa}}_{\mathsf{IBE},A',S',D}(\lambda). \quad (7)\end{aligned}$$

Based on S', we construct a PPT simulator S in the sense of RSIM-SO-CCA in Fig. 10. Denote the final output of $\mathsf{Exp}^{\text{rsim-so-cca-ideal}}_{\mathsf{PKE}_{\mathsf{CHK}},A,S}$ by $(\mathbf{m}_S, \mathcal{M}_S, I_S, out_S)$. Obviously we have that $(\mathbf{m}_S, \mathcal{M}_S, I_S, out_S) = (\mathbf{m}_{S'}, \mathcal{M}_{S'}, I_{S'}, out_{S'})$, which implies that for any PPT distinguisher D,

$$\Pr[D(\mathbf{m}_S, \mathcal{M}_S, I_S, out_S) = 1] = \Pr[D(\mathbf{m}_{S'}, \mathcal{M}_{S'}, I_{S'}, out_{S'}) = 1]. \quad (8)$$

Combining equations (6)-(8), we derive that for any PPT distinguisher D,

$$\begin{aligned}&|\Pr[D(\mathsf{Exp}^{\text{rsim-so-cca-real}}_{\mathsf{PKE}_{\mathsf{CHK}},A}) = 1] - \Pr[D(\mathsf{Exp}^{\text{rsim-so-cca-ideal}}_{\mathsf{PKE}_{\mathsf{CHK}},A,S}) = 1]| \\ &\leq n\mathbf{Adv}^{\text{str-ot}}_{\mathsf{SIG},A_{sig}}(\lambda) + \mathbf{Adv}^{\text{sim-sid-mso-cpa}}_{\mathsf{IBE},A',S',D}(\lambda).\end{aligned}$$

□

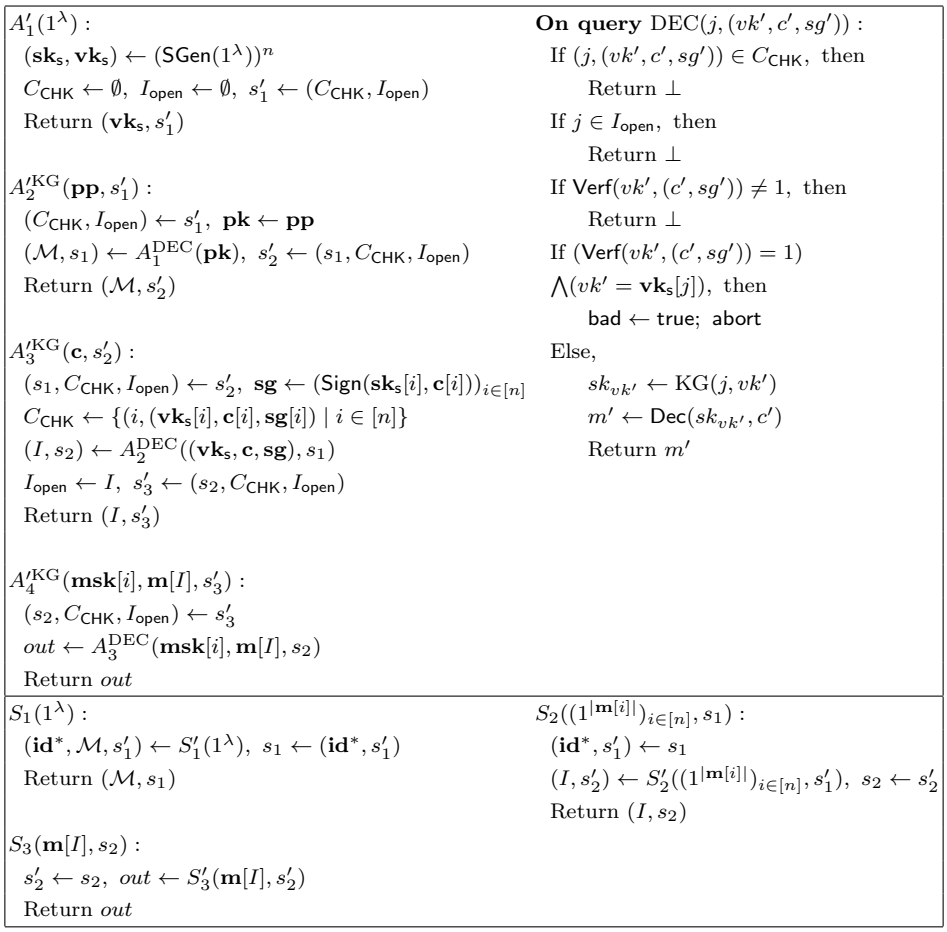

Fig. 10. Adversary $A' = (A'_1, A'_2, A'_3, A'_4)$ and simulator $S = (S_1, S_2, S_3)$ in the proof of Theorem 3.

Remark 5. Similar to the original CHK transformation [9], the above theorem is proved in the standard model. Since strong one-time signatures can be constructed based on any one-way functions [29], if we construct an IBE scheme achieving SIM-sID-mSO-CPA security in the standard model (resp. ROM), we obtain a standard-model (resp. ROM) construction of RSIM-SO-CCA secure PKE.

Remark 6. Theorem 3 can be extended to the indistinguishability-based notions.

5.3 Construction of master-key SOA secure IBE

In the following, we show an IBE construction achieving SIM-sID-mSO-CPA security in the ideal cipher model. Our construction is inspired by the work of Heuer and Poettering [21]. Roughly speaking, consider a hybrid IBE scheme constructed from an identity-based key encapsulation mechanism (IB-KEM) scheme and a data encapsulation mechanism (DEM) scheme. Our result is as follows. The IBE scheme is SIM-sID-mSO-CPA secure in the ideal cipher model, if the underlying IB-KEM scheme is IND-sID-CPA secure, and the DEM scheme has some special properties, "permutation-driven" and "simulatability", introduced in [21].

We begin by recalling the building blocks, and then present our hybrid construction.

(Partial) permutation. For a finite domain $\mathcal{D}$, we say that a relation $R = R^{(l)} \times R^{(r)} \subseteq \mathcal{D} \times \mathcal{D}$ is a partial permutation, if the following two conditions hold: (i) for any $a_1, a_2, b \in \mathcal{D}$, if $(a_1, b) \in R$ and $(a_2, b) \in R$, then $a_1 = a_2$; (ii) for any $a, b_1, b_2 \in \mathcal{D}$, if $(a, b_1) \in R$ and $(a, b_2) \in R$, then $b_1 = b_2$. We say that R is a permutation on $\mathcal{D}$ if in addition $|R| = |\mathcal{D}|$. Trivially, any blockcipher E_k, where k is the secret key, with domain $\mathcal{D}$ is a permutation on $\mathcal{D}$. Throughout this paper, let $\mathcal{P}(\mathcal{D})$ (resp. $\mathcal{PP}(\mathcal{D})$) denote the set of all permutations (resp. partial permutations) on $\mathcal{D}$. Any $R \in \mathcal{PP}(\mathcal{D})$ can be completed to a full permutation by adding sufficiently many $(a, b) \in (\mathcal{D} \setminus R^{(l)}) \times (\mathcal{D} \setminus R^{(r)})$ to it (of course, updating $R^{(l)}$ and $R^{(r)}$ at every step). Furthermore, as pointed out in [21], if $R \leftarrow \mathcal{PP}(\mathcal{D})$ and $(a, b) \leftarrow (\mathcal{D} \setminus R^{(l)}) \times (\mathcal{D} \setminus R^{(r)})$, then the above obtained full permutation is uniformly distributed in $\mathcal{P}(\mathcal{D})$.

Data encapsulation mechanism. A data encapsulation mechanism (DEM) scheme, associated with a finite key space $\mathcal{K}$, consists of two efficient algorithms $(\mathsf{D.enc}, \mathsf{D.dec})$. The encapsulation algorithm $\mathsf{D.enc}(k, m)$ takes a key $k \in \mathcal{K}$ and a valid message m as input, and outputs a ciphertext c. The decapsulation algorithm $\mathsf{D.dec}(k, c)$ takes $k \in \mathcal{K}$ and c as input, and returns a message m or $\perp$, which indicates that c is invalid. For correctness, we require that for any valid message m and any $k \in \mathcal{K}$, if $c \leftarrow \mathsf{D.enc}(k, m)$, then $\mathsf{D.dec}(k, c) = m$.

An oracle DEM for a domain $\mathcal{D}$ is a DEM where both the encapsulation/decapsulation algorithms are allowed to access to a permutation on $\mathcal{D}$ *(in both directions)*. Specifically, for an oracle DEM $\mathsf{ODEM} = (\mathsf{OD.enc}, \mathsf{OD.dec})$ associated with key space $\mathcal{K}$, we write

$$c \leftarrow \mathsf{OD.enc}^{f_p}(k, m) \quad \text{and} \quad m \text{ or } \perp \leftarrow \mathsf{OD.dec}^{f_p}(k, c),$$

which means that $\mathsf{OD.enc}, \mathsf{OD.dec}$ both access to permutation f_p. Correctness requires that for any valid message m, any $k \in \mathcal{K}$ and any $f_p \in \mathcal{P}(\mathcal{D})$, if $c \leftarrow \mathsf{OD.enc}^{f_p}(k, m)$, then $\mathsf{OD.dec}^{f_p}(k, c) = m$.

Definition 5 (Permutation-driven DEM [21]). *For two key spaces $\mathcal{K}_E$ and $\mathcal{K}_O$, a DEM $(\mathsf{D.enc}, \mathsf{D.dec})$, associated with key space $\mathcal{K} = \mathcal{K}_E \times \mathcal{K}_O$, is called $(\mathcal{K}_E, \mathcal{D})$-*permutation-driven*, if there exists an oracle DEM $\mathsf{ODEM} = (\mathsf{OD.enc}, \mathsf{OD.dec})$ for domain $\mathcal{D}$ associated with key space $\mathcal{K}_O$, and a blockcipher $(E_{k^e})_{k^e \in \mathcal{K}_E}$ on domain $\mathcal{D}$, such that for any valid message m, any $k^o \in \mathcal{K}_O$, and ciphertext c, $\mathsf{D.enc}((k^e, k^o), m) := \mathsf{OD.enc}^{E_{k^e}}(k^o, m)$ and $\mathsf{D.dec}((k^e, k^o), c) := \mathsf{OD.dec}^{E_{k^e}}(k^o, c)$.*

Definition 6 (Simulatable oracle DEM [21]). *Consider an oracle DEM $\mathsf{ODEM} = (\mathsf{OD.enc}, \mathsf{OD.dec})$ for domain $\mathcal{D}$ associated with key space $\mathcal{K}$. ODEM is called ϵ-*simulatable*, if there exist two PPT algorithms Fake and Make, and a negligible function ϵ, such that the following conditions hold:*

1. *Algorithm $\mathsf{Fake}(k, |m|)$ takes a key $k \in \mathcal{K}$ and the length of a message m as input, and outputs a ciphertext c and some state information st. Algorithm $\mathsf{Make}(m, st)$ takes a valid message m and st as input, and outputs a partial permutation $f_{pp} \in \mathcal{PP}(\mathcal{D})$.*
2. *For $(c, st) \leftarrow \mathsf{Fake}(k, |m|)$ and $f_{pp} \leftarrow \mathsf{Make}(m, st)$, f_{pp} can be completed to a uniformly distributed full permutation $f_p \in \mathcal{P}(\mathcal{D})$. More precisely, for every $f_{pp} \in \mathcal{PP}(\mathcal{D})$, we write $f_{pp} := R^{(l)} \times R^{(r)}$. If $(c, st) \leftarrow \mathsf{Fake}(k, |m|)$, $f_{pp} \leftarrow \mathsf{Make}(m, st)$, and $f_p \in \mathcal{P}(\mathcal{D})$ is obtained by adding all $(a, b) \leftarrow (\mathcal{D} \setminus R^{(l)}) \times (\mathcal{D} \setminus R^{(r)})$ step by step (and updating $R^{(l)}$ and $R^{(r)}$ at every step) to f_{pp}, then we will obtain $f_p \leftarrow \mathcal{P}(\mathcal{D})$.*

$\underline{\mathsf{Exp}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},A}(\lambda)}$	$\underline{\mathrm{KG}(id):}$
$(id^*, s_1) \leftarrow A_1(1^\lambda),\ b \leftarrow \{0,1\}$	If $id = id^*$, then return $\perp$
$(pp, msk) \leftarrow \mathsf{PGen}(1^\lambda),\ (k_0, c) \leftarrow \mathsf{Encap}(pp, id^*)$	$sk_{id} \leftarrow \mathsf{KGen}(pp, msk, id)$
$k_1 \leftarrow \mathcal{K},\ b' \leftarrow A_2^{\mathrm{KG}}(pp, c, k_b, s_1)$	Return sk_{id}
Return $(b' = b)$	

Fig. 11. Experiment for defining IND-sID-CPA security of an IB-KEM scheme IBKEM.

$\mathsf{PGen}_{\mathrm{hyb}}(1^\lambda)$:	$\mathsf{KGen}_{\mathrm{hyb}}(pp, msk, id)$:	$\mathsf{Enc}_{\mathrm{hyb}}(pp, id, m)$:	$\mathsf{Dec}_{\mathrm{hyb}}(pp, sk_{id}, (c_1, c_2))$:
$(pp, msk) \leftarrow \mathsf{PGen}(1^\lambda)$	$sk_{id} \leftarrow \mathsf{KGen}(pp, msk, id)$	$(k, c_1) \leftarrow \mathsf{Encap}(pp, id)$	$k \leftarrow \mathsf{Decap}(pp, sk_{id}, c_1)$
Return (pp, msk)	Return sk_{id}	$c_2 \leftarrow \mathsf{D.enc}(k, m)$	If $k = \perp$, then return $\perp$
		Return (c_1, c_2)	$m \leftarrow \mathsf{D.dec}(k, c_2)$
			Return m

Fig. 12. Construction of $\mathsf{IBE}_{\mathrm{hyb}}$.

3. *If* $(c, st) \leftarrow \mathsf{Fake}(k, |m|)$, $f_{pp} \leftarrow \mathsf{Make}(m, st)$, *and* $f_p \in \mathcal{P}(\mathcal{D})$ *is obtained by adding all* $(a, b) \leftarrow (\mathcal{D} \setminus R^{(l)}) \times (\mathcal{D} \setminus R^{(r)})$ *to* f_{pp}, *then we have*

$$\Pr\left[c \neq \mathsf{OD.enc}^{f_p}(k, m)\right] \leq \epsilon,$$

where the probability is also taken over all steps of $(a, b) \leftarrow (\mathcal{D} \setminus R^{(l)}) \times (\mathcal{D} \setminus R^{(r)})$.

4. *The total running time of* $\mathsf{Fake}(k, |m|)$ *and* $\mathsf{Make}(m, st)$ *is at most the running time of* $\mathsf{OD.enc}(k, m)$, *not counting the latter's oracle queries.*

As shown in [21], some blockcipher-based DEM schemes standardised by NIST (e.g., [13, 16, 14, 15]) are permutation-driven and simulatable.

IND-sID-CPA security for IB-KEM. We present the notion of IND-sID-CPA security for IB-KEM as follows.

Definition 7 (IND-sID-CPA). *An IB-KEM scheme* $\mathsf{IBKEM} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *is* IND-sID-CPA secure*, if for any PPT adversary A, the advantage*

$$\mathbf{Adv}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},A}(\lambda) := 2|\Pr[\mathsf{Exp}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},A}(\lambda) = 1] - \frac{1}{2}|$$

is negligible, where $\mathsf{Exp}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},A}(\lambda)$ *is defined in Fig. 11.*

Construction of master-key SOA secure IBE. Let $\mathcal{K}_E$ and $\mathcal{K}_O$ be two key spaces. Let $\mathsf{IBKEM} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ be an IB-KEM scheme for session key space $\mathcal{K} = \mathcal{K}_E \times \mathcal{K}_O$. For a DEM scheme $\mathsf{DEM} = (\mathsf{D.enc}, \mathsf{D.dec})$ and $\mathsf{IBKEM} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$, the hybrid IBE construction $\mathsf{IBE}_{\mathrm{hyb}} = (\mathsf{PGen}_{\mathrm{hyb}}, \mathsf{KGen}_{\mathrm{hyb}}, \mathsf{Enc}_{\mathrm{hyb}}, \mathsf{Dec}_{\mathrm{hyb}})$ is shown in Fig. 12. Verifying correctness is trivial. Now we turn to security analysis. Formally, we have the following theorem.

Theorem 4. *If* DEM *is* $(\mathcal{K}_E, \mathcal{D})$*-permutation-driven, where its underlying oracle DEM* ODEM *associated with key space* $\mathcal{K}_O$ *is* ϵ*-simulatable and its underlying* E *is modeled as an ideal cipher, and* IBKEM *is IND-sID-CPA secure and* E*-independent, then* IBE_{hyb} *is SIM-sID-mSO-CPA secure.*

Games $\mathbf{G}_0 - \mathbf{G}_4$

01 $I_{\mathsf{open}} \leftarrow \emptyset,\ KE \leftarrow \emptyset,\ KQ \leftarrow \emptyset$
02 $(\mathbf{id}^* := (\mathbf{id}^*[i])_{i\in[n]}, s_1) \leftarrow A_1^{\mathrm{E}}(1^\lambda)$
03 $(\mathbf{pp}, \mathbf{msk}) \leftarrow (\mathsf{PGen}(1^\lambda))^n$
04 $(\mathcal{M}, s_2) \leftarrow A_2^{\mathrm{KG},\mathrm{E}}(\mathbf{pp}, s_1)$
05 $\mathbf{m} := (\mathbf{m}[i])_{i\in[n]} \leftarrow \mathcal{M}$
06 For $i = 1$ to n
07 $\quad(\mathbf{k}[i], \mathbf{c}_1[i]) \leftarrow \mathsf{Encap}(\mathbf{pp}[i], \mathbf{id}^*[i])$
08 $\quad(\mathbf{k}^{\mathsf{e}}[i], \mathbf{k}^{\mathsf{o}}[i]) \leftarrow \mathbf{k}[i]$
09 $\quad$If $\mathbf{k}^{\mathsf{e}}[i] \in KE \bigcup KQ$, then $\quad //\mathbf{G}_1 - \mathbf{G}_4$
10 $\quad\quad\mathsf{bad}_1 \leftarrow \mathsf{true}$; abort $\quad //\mathbf{G}_1 - \mathbf{G}_4$
11 $\quad\mathbf{c}_2[i] \leftarrow \mathsf{OD.enc}^{\mathrm{E}(\mathbf{k}^{\mathsf{e}}[i],\cdot)}(\mathbf{k}^{\mathsf{o}}[i], \mathbf{m}[i])$ $\quad //\mathbf{G}_0 - \mathbf{G}_1$
12 $\quad(\mathbf{c}_2[i], st_i) \leftarrow \mathsf{Fake}(\mathbf{k}^{\mathsf{o}}[i], |\mathbf{m}[i]|)$ $\quad //\mathbf{G}_2 - \mathbf{G}_4$
13 $\quad f_{pp}^{(i)} \leftarrow \mathsf{Make}(\mathbf{m}[i], st_i)$ $\quad //\mathbf{G}_2 - \mathbf{G}_3$
14 $\quad E_{\mathbf{k}^{\mathsf{e}}[i]} \leftarrow f_{pp}^{(i)}$ $\quad //\mathbf{G}_2 - \mathbf{G}_3$
15 $\quad$If $\mathsf{OD.enc}^{\mathrm{E}(\mathbf{k}^{\mathsf{e}}[i],\cdot)}(\mathbf{k}^{\mathsf{o}}[i], \mathbf{m}[i]) \neq \mathbf{c}_2[i]$, $\quad //\mathbf{G}_2 - \mathbf{G}_3$
16 $\quad\quad$then $\mathsf{bad}_2 \leftarrow \mathsf{true}$; abort $\quad //\mathbf{G}_2 - \mathbf{G}_3$
17 $\quad KE \leftarrow KE \bigcup \{\mathbf{k}^{\mathsf{e}}[i]\},\ KQ \leftarrow KQ \bigcup \{\mathbf{k}^{\mathsf{e}}[i]\}$
18 $\quad\mathbf{c}[i] \leftarrow (\mathbf{c}_1[i], \mathbf{c}_2[i])$
19 $(I, s_3) \leftarrow A_3^{\mathrm{KG},\mathrm{E}}(\mathbf{c}, s_2),\ I_{\mathsf{open}} \leftarrow I$
20 For $i \in I$ $\quad //\mathbf{G}_4$
21 $\quad f_{pp}^{(i)} \leftarrow \mathsf{Make}(st_i, \mathbf{m}[i])$ $\quad //\mathbf{G}_4$
22 $\quad E_{\mathbf{k}^{\mathsf{e}}[i]} \leftarrow f_{pp}^{(i)}$ $\quad //\mathbf{G}_4$
23 $\quad$If $\mathsf{OD.enc}^{\mathrm{E}(\mathbf{k}^{\mathsf{e}}[i],\cdot)}(\mathbf{k}^{\mathsf{o}}[i], \mathbf{m}[i]) \neq \mathbf{c}_2[i]$, $\quad //\mathbf{G}_4$
24 $\quad\quad$then abort $\quad //\mathbf{G}_4$
25 $out \leftarrow A_4^{\mathrm{KG},\mathrm{E}}(\mathbf{msk}[I], \mathbf{m}[I], s_3)$
26 Return $(\mathbf{m}, \mathcal{M}, \mathbf{id}^*, I, out)$

On query $\mathrm{KG}(j, id)$:
27 If $(id = \mathbf{id}^*[j]) \bigvee (j \in I_{\mathsf{open}})$, then
28 $\quad$Return $\perp$
29 $sk_{id} \leftarrow \mathsf{KGen}(\mathbf{pp}[j], \mathbf{msk}[j], id)$
30 Return sk_{id}

On query $\mathrm{E}^+(k^{\mathsf{e}}, a)$:
31 If $k^{\mathsf{e}} \in KE \setminus \{\mathbf{k}^{\mathsf{e}}[I_{\mathsf{open}}]\}$, $\quad //\mathbf{G}_3 - \mathbf{G}_4$
32 $\quad$then $\mathsf{bad}_3 \leftarrow \mathsf{true}$; abort $\quad //\mathbf{G}_3 - \mathbf{G}_4$
33 If $a \notin E_{k^{\mathsf{e}}}^{(l)}$, then
34 $\quad b \leftarrow \mathcal{D} \setminus E_{k^{\mathsf{e}}}^{(r)}$
35 $\quad E_{k^{\mathsf{e}}} \leftarrow E_{k^{\mathsf{e}}} \bigcup \{(a, b)\}$
36 $KQ \leftarrow KQ \bigcup \{k^{\mathsf{e}}\}$
37 Return $E_{k^{\mathsf{e}}}^{+}(a)$

On query $\mathrm{E}^-(k^{\mathsf{e}}, b)$:
38 If $k^{\mathsf{e}} \in KE \setminus \{\mathbf{k}^{\mathsf{e}}[I_{\mathsf{open}}]\}$, $\quad //\mathbf{G}_3 - \mathbf{G}_4$
39 $\quad$then $\mathsf{bad}_3 \leftarrow \mathsf{true}$; abort $\quad //\mathbf{G}_3 - \mathbf{G}_4$
40 If $b \notin E_{k^{\mathsf{e}}}^{(r)}$, then
41 $\quad a \leftarrow \mathcal{D} \setminus E_{k^{\mathsf{e}}}^{(l)}$
42 $\quad E_{k^{\mathsf{e}}} \leftarrow E_{k^{\mathsf{e}}} \bigcup \{(a, b)\}$
43 $KQ \leftarrow KQ \bigcup \{k^{\mathsf{e}}\}$
44 Return $E_{k^{\mathsf{e}}}^{-}(b)$

Fig. 13. Games $\mathbf{G}_0 - \mathbf{G}_4$ in the proof of Theorem 4. Note that lines ending with a range of games $\mathbf{G}_{j_1} - \mathbf{G}_{j_2}$ (resp. $\mathbf{G}_j$) are only executed when a game within the range is run. Here $\{\mathbf{k}^{\mathsf{e}}[I']\} := \{\mathbf{k}^{\mathsf{e}}[i] \mid i \in I'\}$.

Proof. Before going into the formal proof, we introduce some notation. As pointed out in [21], any uniformly distributed partial permutation $R = R^{(l)} \times R^{(r)} \subseteq \mathcal{D} \times \mathcal{D}$ can be completed to a uniformly distributed full permutation, by adding sufficiently many uniformly chosen pairs from $(\mathcal{D} \setminus R^{(l)}) \times (\mathcal{D} \setminus R^{(r)})$. Although $E_{k^{\mathsf{e}}} \in \mathcal{P}(\mathcal{D})$ (for any $k^{\mathsf{e}} \in \mathcal{K}_E$), in the following proof, we abuse notation and let "$E_{k^{\mathsf{e}}}$" denote the partial permutation which will be completed to $E_{k^{\mathsf{e}}}$. Additionally, let $E_{k^{\mathsf{e}}}^{+}(\cdot) : E_{k^{\mathsf{e}}}^{(l)} \to E_{k^{\mathsf{e}}}^{(r)}$ and $E_{k^{\mathsf{e}}}^{-}(\cdot) : E_{k^{\mathsf{e}}}^{(r)} \to E_{k^{\mathsf{e}}}^{(l)}$ be two maps as follows: for any $(a, b) \in E_{k^{\mathsf{e}}}$, $E_{k^{\mathsf{e}}}^{+}(a) = b$ and $E_{k^{\mathsf{e}}}^{-}(b) = a$.

Now we turn to the formal proof. For any PPT adversary A attacking $\mathsf{IBE}_{\mathrm{hyb}}$ in the sense of SIM-sID-mSO-CPA, let q_e denote the number of permutation queries made by A. We prove the theorem with a sequence of games $\mathbf{G}_0 - \mathbf{G}_4$ in Fig. 13.

Firstly, note that the view of A in game $\mathbf{G}_0$ is exactly the same as that in $\mathsf{Exp}^{\text{sim-sid-mso-cpa-real}}_{\mathsf{IBE}_{\mathrm{hyb}},A}$, so the final outputs of these two games are identical, i.e., $\mathbf{G}_0 = \mathsf{Exp}^{\text{sim-sid-mso-cpa-real}}_{\mathsf{IBE}_{\mathrm{hyb}},A}(\lambda)$.

Games $\mathbf{G}_1$ and $\mathbf{G}_0$ are identical until bad_1 (lines 09 and 10) occurs. Thus, for any PPT distinguisher D, $|\Pr[D(\mathbf{G}_1) = 1] - \Pr[D(\mathbf{G}_0) = 1]| \leq \Pr[\mathsf{bad}_1]$. Note that bad_1 occurs iff during the generation of the challenge ciphertext vector in $\mathbf{G}_0$, the i^{th} iteration of $\mathsf{OD.enc}$ would have access to a *non-empty* permutation $E(\mathbf{k}^{\mathsf{e}}[i], \cdot)$ for some $i \in [n]$. Let $\mathsf{bad}_1^{(i)}$ denote the event bad_1 which is caused by the i^{th} iteration of $\mathsf{OD.enc}$. For any fixed $i \in [n]$, we construct a PPT IND-sID-CPA adversary B attacking IBKEM, based on A, as shown in Fig. 14. Denote

$B_1(1^\lambda)$:	**On query** $\mathrm{KG}_A(j, id)$:
$KE \leftarrow \emptyset,\ KQ \leftarrow \emptyset,\ (\mathbf{id}^*, s_1) \leftarrow A_1^{\mathrm{E}}(1^\lambda)$	If $(id = \mathbf{id}^*[j]) \bigvee (j \in I_{\mathsf{open}})$, then return $\perp$
$s_1^B \leftarrow (\mathbf{id}^*, s_1, KE, KQ)$	If $j \neq i$, then
Return $(\mathbf{id}^*[i], s_1^B)$	$sk_{id} \leftarrow \mathsf{KGen}(\mathbf{pp}[j], \mathbf{msk}[j], id)$
	Else,
$B_2^{\mathrm{KG}}(pp, c, k, (\mathbf{id}^*, s_1, KE, KQ))$:	$sk_{id} \leftarrow \mathrm{KG}(id)$
$\mathbf{pp}[i] \leftarrow pp,\ I_{\mathsf{open}} \leftarrow \emptyset$	Return sk_{id}
$(\mathbf{pp}[n]\backslash\{i\}, \mathbf{msk}[n]\backslash\{i\}) \leftarrow (\mathsf{PGen}(1^\lambda))_{j\in[n]\backslash\{i\}}$	
$(\mathcal{M}, s_2) \leftarrow A_2^{\mathrm{KG}_A, \mathrm{E}}(\mathbf{pp}, s_1)$	**On query** $\mathrm{E}^+(k^{\mathsf{e}}, a)$:
$\mathbf{m} \leftarrow \mathcal{M}$	If $a \notin E^{(l)}_{k^{\mathsf{e}}}$, then
For $j = 1$ to $i - 1$	$b \leftarrow \mathcal{D} \setminus E^{(r)}_{k^{\mathsf{e}}}$
$(\mathbf{k}[j], \mathbf{c}_1[j]) \leftarrow \mathsf{Encap}(\mathbf{pp}[j], \mathbf{id}^*[j])$	$E_{k^{\mathsf{e}}} \leftarrow E_{k^{\mathsf{e}}} \bigcup \{(a, b)\}$
$(\mathbf{k}^{\mathsf{e}}[j], \mathbf{k}^{\mathsf{o}}[j]) \leftarrow \mathbf{k}[j]$	$KQ \leftarrow KQ \bigcup \{k^{\mathsf{e}}\}$
$\mathbf{c}_2[j] \leftarrow \mathsf{OD.enc}^{\mathrm{E}(\mathbf{k}^{\mathsf{e}}[j], \cdot)}(\mathbf{k}^{\mathsf{o}}[j], \mathbf{m}[j])$	Return $E^+_{k^{\mathsf{e}}}(a)$
$KE \leftarrow KE \bigcup \{\mathbf{k}^{\mathsf{e}}[j]\},\ KQ \leftarrow KQ \bigcup \{\mathbf{k}^{\mathsf{e}}[j]\}$	
$\mathbf{c}[j] \leftarrow (\mathbf{c}_1[j], \mathbf{c}_2[j])$	**On query** $\mathrm{E}^-(k^{\mathsf{e}}, b)$:
For $j = i$	If $b \notin E^{(r)}_{k^{\mathsf{e}}}$, then
$(\mathbf{k}^{\mathsf{e}}[i], \mathbf{k}^{\mathsf{o}}[i]) \leftarrow k$	$a \leftarrow \mathcal{D} \setminus E^{(l)}_{k^{\mathsf{e}}}$
If $\mathbf{k}^{\mathsf{e}}[i] \in KE \bigcup KQ$, then $\beta' \leftarrow 1$	$E_{k^{\mathsf{e}}} \leftarrow E_{k^{\mathsf{e}}} \bigcup \{(a, b)\}$
Else, $\beta' \leftarrow 0$	$KQ \leftarrow KQ \bigcup \{k^{\mathsf{e}}\}$
Return β'	Return $E^-_{k^{\mathsf{e}}}(b)$

Fig. 14. Adversary $B = (B_1, B_2)$ in the proof of Theorem 4.

by β the challenge bit of $\mathsf{Exp}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B}$. Note that B perfectly simulates $\mathbf{G}_0$ for A until the i^{th} iteration of $\mathsf{OD.enc}$ during the generation of the challenge ciphertext vector. When $\beta = 0$, the distribution of $(\mathbf{k}^{\mathsf{e}}[i], \mathbf{k}^{\mathsf{o}}[i])$ in B's simulation is identical to that in $\mathbf{G}_0$. In other words, $\Pr[\beta' = 1 \mid \beta = 0] = \Pr[\mathsf{bad}_1^{(i)}]$. On the other hand, when $\beta = 1$, $\mathbf{k}[i] = k$ is uniformly and independently sampled from $\mathcal{K}$. Hence, $\Pr[\beta' = 1 \mid \beta = 1] \leq \frac{q_e+n}{|\mathcal{K}|}$. Since IBKEM is IND-sID-CPA secure, we have that

$$\mathbf{Adv}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B}(\lambda) = 2\left|\Pr[\mathsf{Exp}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B}(\lambda) = 1] - \frac{1}{2}\right|$$
$$= \left|\Pr[\beta' = 1 \mid \beta = 0] - \Pr[\beta' = 1 \mid \beta = 1]\right| \geq \left|\Pr[\mathsf{bad}^{(i)}] - \frac{q_e + n}{|\mathcal{K}|}\right|$$

is negligible. Thus, $\Pr[\mathsf{bad}_1^{(i)}] \leq \mathbf{Adv}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B}(\lambda) + \frac{q_e+n}{|\mathcal{K}|}$ is negligible. A union bound shows that

$$\Pr[\mathsf{bad}_1] \leq n \cdot (\mathbf{Adv}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B}(\lambda) + \frac{q_e + n}{|\mathcal{K}|}).$$

Now, we stress that in $\mathbf{G}_1$, for each iteration of $\mathsf{OD.enc}$ during the generation of the challenge ciphertext vector, $\mathsf{OD.enc}$ always accesses to an empty permutation.

In game $\mathbf{G}_2$, $\mathsf{OD.enc}$ is replaced with Fake and Make. More precisely, line 11 is replaced with lines 12-16. For any $i \in [n]$, the partial permutation $f^{(i)}_{\mathcal{PP}}$ output by Make can embedded into $E_{\mathbf{k}^{\mathsf{e}}[i]}$, since $E_{\mathbf{k}^{\mathsf{e}}[i]}$ is empty when it is accessed to by $\mathsf{OD.enc}$. We note that ODEM is ϵ-simulatable, so from A's point of view, $\mathbf{G}_2$ and $\mathbf{G}_1$ are identical except that $\mathbf{G}_2$ sets bad_2. Similarly, a simple union bound shows that $\Pr[\mathsf{bad}_2] \leq n\epsilon$. Therefore, for any PPT distinguisher D, $|\Pr[D(\mathbf{G}_2) = 1] - \Pr[D(\mathbf{G}_1) = 1]| \leq n\epsilon$.

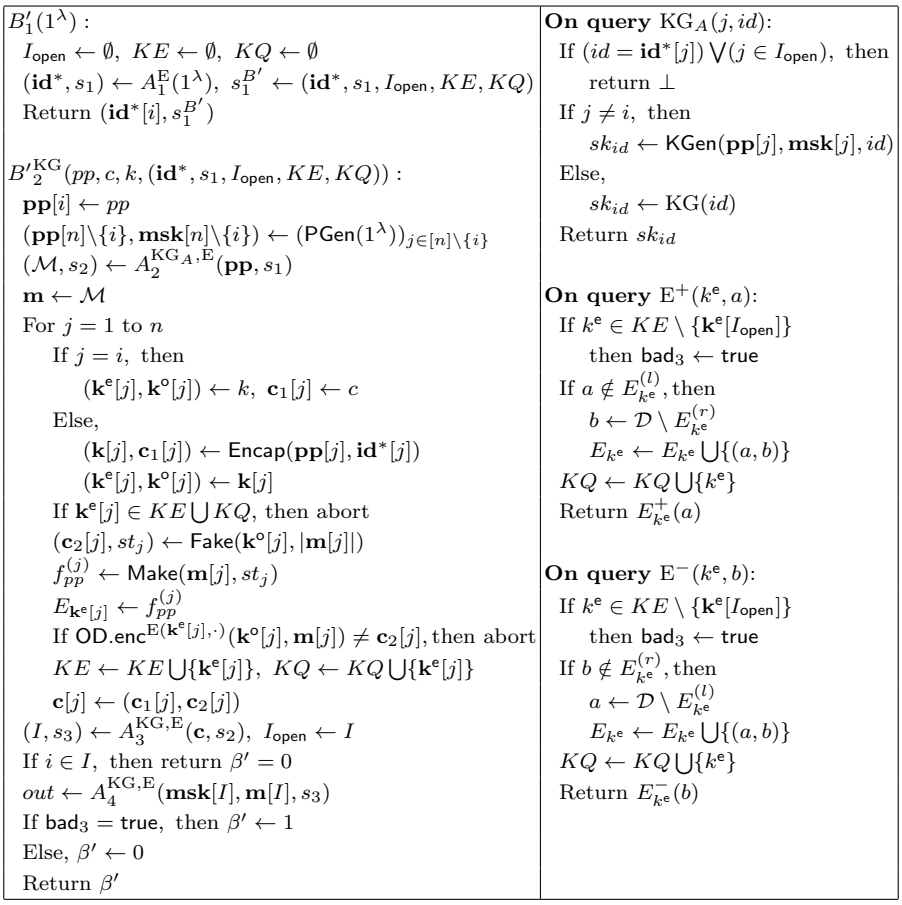

Fig. 15. Adversary $B' = (B_1', B_2')$ in the proof of Theorem 4. Here $\{\mathbf{k}^{\mathsf{e}}[I_{\mathsf{open}}]\} := \{\mathbf{k}^{\mathsf{e}}[i] \mid i \in I_{\mathsf{open}}\}$.

Games $\mathbf{G}_3$ and $\mathbf{G}_2$ are identical until bad_3 (lines 31, 32, 38 and 39) occurs. Thus, for any PPT distinguisher D, $|\Pr[D(\mathbf{G}_3) = 1] - \Pr[D(\mathbf{G}_2) = 1]| \leq \Pr[\mathsf{bad}_3]$. Note that bad_3 occurs iff A submits a query on (k^{e}, x) to oracles E^+ or E^-, such that $k^{\mathsf{e}} \in KE \setminus \{\mathbf{k}^{\mathsf{e}}[I_{\mathsf{open}}]\}$. Let $\mathsf{bad}_3^{(i)}$ denote the event bad_3 which is caused by the the query (k^{e}, x) satisfying $k^{\mathsf{e}} = \mathbf{k}^{\mathsf{e}}[i]$. Similarly, for any fixed $i \in [n]$, we construct a PPT IND-sID-CPA adversary B' attacking IBKEM, based on A, as shown in Fig. 15. Denote by β the challenge bit of $\mathsf{Exp}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B'}$. When $\beta = 0$, B''s simulation for A and game $\mathbf{G}_2$ are identical except for the case $i \in I$. In B''s simulation, if $i \in I$, B' will return $\beta' = 0$ directly, without continuing the simulation for A. We stress that in this case (i.e., $i \in I$), $\mathsf{bad}_3^{(i)}$ does not occur and will not occur. Therefore, the probability that $\mathsf{bad}_3^{(i)}$ occurs in B''s simulation (when $\beta = 0$) and the probability in $\mathbf{G}_2$ are equivalent. In other words, $\Pr[\beta' = 1 \mid \beta = 0] = \Pr[\mathsf{bad}_3^{(i)}]$. On the other hand, when $\beta = 1$, $(\mathbf{k}^{\mathsf{e}}[i], \mathbf{k}^{\mathsf{o}}[i]) = k$ is uniformly and independently sampled from $\mathcal{K}$. We claim that $\mathbf{k}^{\mathsf{e}}[i]$ is uniformly distributed from A's point of view. The reason is as follows. Among all the elements that A sees, only $(\mathbf{c}_1[i], \mathbf{c}_2[i])$ might contain information about $\mathbf{k}^{\mathsf{e}}[i]$. Note that $\mathbf{k}^{\mathsf{e}}[i]$ is independent of $\mathbf{c}_1[i]$, because $\mathbf{k}^{\mathsf{e}}[i]$ independent of the computation of Encap. $\mathbf{k}^{\mathsf{e}}[i]$ is also independent of $\mathbf{c}_2[i]$, since $\mathbf{c}_2[i]$ is computed with $\mathsf{Fake}(\mathbf{k}^{\mathsf{o}}[i], |\mathbf{m}[i]|)$ and $\mathbf{k}^{\mathsf{e}}[i]$ is independent of $\mathbf{k}^{\mathsf{o}}[i]$ and uniformly distributed. Hence, the probability that A generates $k^{\mathsf{e}} = \mathbf{k}^{\mathsf{e}}[i]$ is at most $\frac{q_e}{|\mathcal{K}|}$. Thus, $\Pr[\beta' = 1 \mid \beta = 1] \leq \frac{q_e}{|\mathcal{K}|}$.

$S_1(1^\lambda)$:
$I_{\mathsf{open}} \leftarrow \emptyset,\ KE \leftarrow \emptyset,\ KQ \leftarrow \emptyset$
$(\mathbf{id}^*, s_1) \leftarrow A_1^{\mathrm{E}}(1^\lambda)$
$(\mathbf{pp}, \mathbf{msk}) \leftarrow (\mathsf{PGen}(1^\lambda))^n$
$(\mathcal{M}, s_2) \leftarrow A_2^{\mathrm{KG,E}}(\mathbf{pp}, s_1)$
$\widetilde{s_1} \leftarrow (\mathbf{id}^*, \mathbf{pp}, \mathbf{msk}, s_2, I', KE, KQ, (E_{k^{\mathsf{e}}})_{k^{\mathsf{e}} \in KQ})$
Return $(\mathbf{id}^*, \mathcal{M}, \widetilde{s_1})$

$S_2((1^{|\mathbf{m}[i]|})_{i\in[n]}, \widetilde{s_1})$:
$(\mathbf{id}^*, \mathbf{pp}, \mathbf{msk}, s_2, I', KE, KQ, (E_{k^{\mathsf{e}}})_{k^{\mathsf{e}} \in KQ}) \leftarrow \widetilde{s_1}$
For $i = 1$ to n
$(\mathbf{k}[i], \mathbf{c}_1[i]) \leftarrow \mathsf{Encap}(\mathbf{pp}[i], \mathbf{id}^*[i])$
$(\mathbf{k}^{\mathsf{e}}[i], \mathbf{k}^{\mathsf{o}}[i]) \leftarrow \mathbf{k}[i]$
If $\mathbf{k}^{\mathsf{e}}[i] \in KE \bigcup KQ$, then abort
$(\mathbf{c}_2[i], st_i) \leftarrow \mathsf{Fake}(\mathbf{k}^{\mathsf{o}}[i], |\mathbf{m}[i]|)$
$KE \leftarrow KE \bigcup \{\mathbf{k}^{\mathsf{e}}[i]\},\ KQ \leftarrow KQ \bigcup \{\mathbf{k}^{\mathsf{e}}[i]\}$
$\mathbf{c}[i] \leftarrow (\mathbf{c}_1[i], \mathbf{c}_2[i])$
$(I, s_3) \leftarrow A_3^{\mathrm{KG,E}}(\mathbf{c}, s_2),\ I_{\mathsf{open}} \leftarrow I$
$\widetilde{s_2} \leftarrow (s_3, I_{\mathsf{open}}, KE, KQ, (E_{k^{\mathsf{e}}})_{k^{\mathsf{e}} \in KQ}, (st_i)_{i\in[n]})$
Return $(I, \widetilde{s_2})$

$S_3(\mathbf{m}[I], \widetilde{s_2})$:
$(s_3, I_{\mathsf{open}}, KE, KQ, (E_{k^{\mathsf{e}}})_{k^{\mathsf{e}} \in KQ}, (st_i)_{i\in[n]}) \leftarrow \widetilde{s_2}$
For $i \in I$
$f_{pp}^{(i)} \leftarrow \mathsf{Make}(\mathbf{m}[i], st_i),\ E_{\mathbf{k}^{\mathsf{e}}[i]} \leftarrow f_{pp}^{(i)}$
If $\mathsf{OD.enc}^{\mathrm{E}(\mathbf{k}^{\mathsf{e}}[i],\cdot)}(\mathbf{k}^{\mathsf{o}}[i], \mathbf{m}[i]) \neq \mathbf{c}_2[i],$ then abort
$out \leftarrow A_4^{\mathrm{KG,E}}(\mathbf{msk}[I], \mathbf{m}[I], s_3)$
Return out

On query $\mathrm{KG}(j, id)$:
If $(id = \mathbf{id}^*[j]) \bigvee (j \in I_{\mathsf{open}}),$ then
Return $\perp$
$sk_{id} \leftarrow \mathsf{KGen}(\mathbf{pp}[j], \mathbf{msk}[j], id)$
Return sk_{id}

On query $\mathrm{E}^+(k^{\mathsf{e}}, a)$:
If $k^{\mathsf{e}} \in KE \setminus \{\mathbf{k}^{\mathsf{e}}[I_{\mathsf{open}}]\},$ then abort
If $a \notin E_{k^{\mathsf{e}}}^{(l)}$, then
$b \leftarrow \mathcal{D} \setminus E_{k^{\mathsf{e}}}^{(r)}$
$E_{k^{\mathsf{e}}} \leftarrow E_{k^{\mathsf{e}}} \bigcup \{(a, b)\}$
$KQ \leftarrow KQ \bigcup \{k^{\mathsf{e}}\}$
Return $E_{k^{\mathsf{e}}}^{+}(a)$

On query $\mathrm{E}^-(k^{\mathsf{e}}, b)$:
If $k^{\mathsf{e}} \in KE \setminus \{\mathbf{k}^{\mathsf{e}}[I_{\mathsf{open}}]\}$, then abort
If $b \notin E_{k^{\mathsf{e}}}^{(r)}$, then
$a \leftarrow \mathcal{D} \setminus E_{k^{\mathsf{e}}}^{(l)}$
$E_{k^{\mathsf{e}}} \leftarrow E_{k^{\mathsf{e}}} \bigcup \{(a, b)\}$
$KQ \leftarrow KQ \bigcup \{k^{\mathsf{e}}\}$
Return $E_{k^{\mathsf{e}}}^{-}(b)$

Fig. 16. Simulator $S = (S_1, S_2, S_3)$ in the proof of Theorem 4.

Since IBKEM is IND-sID-CPA secure,

$$\mathbf{Adv}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B'}(\lambda) = 2\left|\Pr[\mathsf{Exp}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B'}(\lambda) = 1] - \frac{1}{2}\right|$$
$$= \left|\Pr\left[\beta' = 1 \mid \beta = 0\right] - \Pr\left[\beta' = 1 \mid \beta = 1\right]\right| \geq \left|\Pr\left[\mathsf{bad}_3^{(i)}\right] - \frac{q_e}{|\mathcal{K}|}\right|$$

is negligible. Thus, $\Pr\left[\mathsf{bad}_3^{(i)}\right] \leq \mathbf{Adv}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B'}(\lambda) + \frac{q_e}{|\mathcal{K}|}$ is negligible. A union bound shows that

$$\Pr\left[\mathsf{bad}_3\right] \leq n \cdot (\mathbf{Adv}^{\text{ind-sid-cpa}}_{\mathsf{IBKEM},B'}(\lambda) + \frac{q_e}{|\mathcal{K}|}).$$

In game $\mathbf{G}_4$, the invocation of Make, the embedding procedure of $E_{\mathbf{k}^{\mathsf{e}}[i]}$, and the consistency check of $\mathsf{OD.enc}$ (i.e., lines 13-16) are all moved from the generation of the challenge ciphertext vector to the opening procedure (i.e., lines 20-24), and the moved procedures are executed only for $i \in I$. We claim that from A's point of view, $\mathbf{G}_4 = \mathbf{G}_3$. The reasons are as follows. (i) In both $\mathbf{G}_3$ and $\mathbf{G}_4$, what A_3 sees are the challenge ciphertext vector $\mathbf{c}$ and two oracles, KG and E. (ii) The moved procedures do not affect the generation of $\mathbf{c}$. (iii) From A's point of view, the only elements, which might be affected by the moved procedures, are the partial permutations $E_{\mathbf{k}^{\mathsf{e}}[i]}$ for all $i \in [n]$. (iv) The key generation oracle KG has nothing to do with $E_{\mathbf{k}^{\mathsf{e}}[i]}$ for any

$i \in [n]$. (v) The modification from $\mathbf{G}_0$ to $\mathbf{G}_1$ (i.e., bad_1) guarantees that when Make is invoked in $\mathbf{G}_3$ (line 13), $E_{\mathbf{k}^e[i]}$ (for any $i \in [n]$) is empty; the modification from $\mathbf{G}_2$ to $\mathbf{G}_3$ (i.e., bad_3) guarantees that in $\mathbf{G}_3$, the oracles E^+ and E^- do not fill up $E_{\mathbf{k}^e[i]}$ (for any $i \in [n]$) with any pairs or reveal any information about $E_{\mathbf{k}^e[i]}$ (for any $i \in [n]$), until A makes an opening query I such that $i \in I$. In other words, in $\mathbf{G}_3$, for every $i \in [n]$ there is at most one pair in $E_{\mathbf{k}^e[i]}$ when A_3 outputs I (in line 19), and no information on these $\leq n$ pairs are revealed to A_3 other than $(\mathcal{M}, \mathbf{c})$.

Therefore, we derive that $\mathbf{G}_4 = \mathbf{G}_3$.

Now, we construct a PPT simulator S as shown in Fig. 16. Obviously S simulates $\mathbf{G}_4$ for A perfectly, so we derive that $\mathsf{Exp}^{\text{sim-sid-mso-cpa-ideal}}_{\mathsf{IBE}_{\text{hyb}},A,S}(\lambda) = \mathbf{G}_4$, which concludes this proof.

□

Remark 7. As mentioned earlier, the notion of indistinguishability-based sID-mSO-CPA (IND-sID-mSO-CPA) security can be similarly defined. With similar technique as that of Theorem 4, we also can prove that $\mathsf{IBE}_{\text{hyb}}$ is IND-sID-mSO-CPA secure.

Acknowledgments. Zhengan Huang was supported by National Natural Science Foundation of China (No. 61702125, 61702126), and Scientific Research Foundation for Post-doctoral Researchers of Guangzhou (No. gdbsh2016020). Junzuo Lai was supported by National Natural Science Foundation of China (No. 61572235), Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2015A030306045), and Pearl River S&T Nova Program of Guangzhou. Wenbin Chen was partly supported by the Program for Innovative Research Team in Education Department of Guangdong Province Under Grant No.2015KCXTD014. and No.2016KCXTD017. Jin Li was supported by National Natural Science Foundation of China (No. 61472091), National Natural Science Foundation for Outstanding Youth Foundation (No. 61722203), and the State Key Laboratory of Cryptology, Beijing, China.

References

1. Bellare M., Hofheinz D., Yilek S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: EUROCRYPT 2009. LNCS, vol. 5479, pp. 1-35. Springer (2009).
2. Bellare M., Rogaway P.: Code-based game-playing proofs and the security of triple encryption. In: EUROCRYPT 2006. LNCS, vol. 4004, pp. 409-426. Springer (2006).
3. Bellare M., Waters B., Yilek S.: Identity-based encryption secure against selective opening attack. In: TCC 2011. LNCS, vol. 6597, pp. 235-252, Springer (2011).
4. Bellare M., Yilek S.: Encryption schemes secure under selective opening attack. Cryptology ePrint Archive: Report 2009/101, 2009. https://eprint.iacr.org/2009/101/20120923:212424. Accessed 09 September 2017.
5. Bentahar K., Farshim P., Malone-Lee J., Smart N. P.: Generic constructions of identity-based and certificateless KEMs. J. Cryptol. 21(2), 178-199 (2008).
6. Böhl F., Hofheinz D., Kraschewski D.: On definitions of selective opening security. In: PKC 2012. LNCS, vol. 7293, pp. 522-539. Springer (2012).
7. Boneh D., Boyen X.: Efficient selective-ID secure identity-based encryption without random oracles. In: EUROCRYPT 2004. LNCS, vol. 3027, pp. 223-238. Springer (2004).
8. Boyen X., Li Q.: All-But-Many Lossy Trapdoor Functions from Lattices and Applications. In: CRYPTO 2017. LNCS, vol, 10403, pp. 298-331. Springer (2017).
9. Canetti R., Halevi S., Katz J.: Chosen-ciphertext security from identity-based encryption. In: EUROCRYPT 2004. LNCS, vol. 3027, pp. 207-222. Springer (2004).
10. Canetti R., Halevi S., Katz J.: Adaptively-secure, non-interactive public-key encryption. In: TCC 2005. LNCS, vol. 3378, pp. 150-168. Springer (2005).
11. Cramer R., Shoup V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: CRYPTO 1998. pp. 13-25. Springer (1998).

12. Cramer R., Shoup V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: EUROCRYPT 2002. LNCS, vol. 2332, pp. 45-64. Springer (2002).
13. Dworkin M. J.: SP 800-38A: Recommendation for block cipher modes of operation: Methods and techniques. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, United States (2001). http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf. Accessed 02 October 2017.
14. Dworkin M. J.: SP 800-38C: Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, United States (2007). http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf. Accessed 02 October 2017.
15. Dworkin M. J.: SP 800-38D: Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, United States (2007). http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf. Accessed 02 October 2017.
16. Dworkin M. J.: Addendum to SP 800-38A: Recommendation for block cipher modes of operation: Three variants of ciphertext stealing for CBC mode. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, United States (2010). http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a-add.pdf. Accessed 02 October 2017.
17. Fehr S., Hofheinz D., Kiltz E., and Wee H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: EUROCRYPT 2010. LNCS, vol. 6110, pp. 381-402. Springer (2010).
18. Hazay C., Patra A., Warinschi B.: Selective opening security for receivers. In: ASIACRYPT 2015. LNCS, vol. 9452, pp. 443-469. Springer (2015).
19. Hemenway B., Libert B., Ostrovsky R., Vergnaud D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: ASIACRYPT 2011. LNCS, vol. 7073, pp. 70-88. Springer (2011).
20. Heuer F., Jager T., Kiltz E., Schäge S.: On the Selective Opening Security of Practical Public-Key Encryption Schemes. In: PKC 2015. LNCS, vol. 9020, pp. 27-51. Springer (2015).
21. Heuer F., Poettering B.: Selective opening security from simulatable data encapsulation. In: ASIACRYPT 2016, LNCS, vol. 10032. Springer (2016).
22. Hofheinz D.: All-but-many lossy trapdoor functions. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 209-227. Springer (2012).
23. Hofheinz D., Jager T., Rupp A.: Public-Key Encryption with Simulation-Based Selective-Opening Security and Compact Ciphertexts. In: TCC 2016-B. LNCS, vol. 9986, pp. 146-168. Springer (2016).
24. Huang Z., Liu S., Qin B.: Sender-Equivocable Encryption Schemes Secure against Chosen-Ciphertext Attacks Revisited. In: PKC 2013, pp. 369-385. Springer (2013).
25. Huang Z., Liu S., Qin B., Chen K.: Fixing the Sender-Equivocable Encryption Scheme in Eurocrypt 2010. In: Intelligent Networking and Collaborative Systems (INCoS), 2013 5th International Conference on, pp. 366-372. IEEE (2013).
26. Holt, J. E.: Key Privacy for Identity Based Encryption. In: IACR Cryptology ePrint Archive, 2006, 120 (2006).
27. Jia D., Lu X., Li B.: Constructions Secure Against Receiver Selective Opening and Chosen Ciphertext Attacks. In: CT-RSA 2017. LNCS, vol. 10159, pp. 417-431. Springer (2017).
28. Lai J., Deng R. H., Liu S., Weng J., Zhao Y.: Identity-Based Encryption Secure against Selective Opening Chosen-Ciphertext Attack. In: EUROCRYPT 2014. LNCS, vol. 8441, pp. 77-92. Springer (2014).
29. Lamport L.: Constructing Digital Signatures from a One-Way Function. Technical Report CSL-98, SRI International, Palo Alto (1979). http://lamport.azurewebsites.net/pubs/dig-sig.pdf. Accessed 06 October 2017.
30. Li F., Shirase M., Takagi T.: Efficient Multi-PKG ID-Based Signcryption for Ad Hoc Networks. In: Information Security and Cryptology. Inscrypt 2008. Lecture Notes in Computer Science, vol 5487, pp. 289-304. Springer, Berlin, Heidelberg (2008).
31. Libert B., Sakzad A., Stehlé D., Steinfeld R.: All-But-Many Lossy Trapdoor Functions and Selective Opening Chosen-Ciphertext Security from LWE. In: CRYPTO 2017. LNCS, vol, 10403, pp. 332-364. Springer (2017).
32. Paillier P.: Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT 1999, Vol. 1592, LNCS. pp. 223-238. Springer (1999).
33. Paterson K.G., Srinivasan S.: Security and Anonymity of Identity-Based Encryption with Multiple Trusted Authorities. In: Pairing-Based Cryptography - Pairing 2008. Pairing 2008. Lecture Notes in Computer Science, vol 5209, pp. 354-375. Springer, Berlin, Heidelberg (2008).
34. Wang, S., Cao, Z.: Practical identity-based encryption (IBE) in multiple PKG environments and its applications. In: Cryptology ePrint Archive, Report 2007/100 (2007), http://eprint.iacr.org/

A Proof of Lemma 3

Proof (of Lemma 3). First, parse $\mathbf{pk}[i] = (g_1, g_2, h, \theta, \varphi, hk)$, $\mathbf{sk}[i] = (x, y, a, b, a', b')$ and $\mathbf{tk}[i] = z$. Denote the i^{th} challenge ciphertext in $\mathbf{G}_{n+i-1}$ by $\mathbf{c}[i] = (u, v, w, e)$. Let $r_1 := \log_{g_1} u$ and $r_2 := \log_{g_2} v = \log_{g_2} g_1 + r_1$. So we have $r_1 \neq r_2$.

Game $\mathbf{G}_{n+i-1}$ sets bad iff adversary A, without obtaining $\mathbf{sk}[i]$ through the opening query, submits a decryption query $(i, c' = (u', v', w', e'))$ such that $(i, c') \notin C$, $u'^z \neq v'$, and $u'^{a+\alpha' a'} v'^{b+\alpha' b'} = e'$, where $\alpha' = \mathsf{HEvl}(hk, (u', v', w'))$. There are four possible cases.

Case 1. $(u', v', w', e') = (u, v, w, e)$.

Since $(i, c') \notin C$, this case occurs only when A submits such a decryption query *before* receiving $\mathbf{c}$. In $\mathbf{G}_{n+i-1}$, u is uniformly and independently chosen from $\mathbb{G}_q$. Notice that A makes at most q_d decryption queries, and that the best circumstance for A is that each decryption query will help A to eliminate one possible value of u. Hence, the probability that Case 1 occurs is at most $\frac{q_d}{q - q_d}$, which is negligible.

We stress that this is a very loose bound. Because in $\mathbf{G}_{n+i-1}$, $v = g_1 g_2^r$ and $\mathbf{c}[i]$ is generated with $\mathbf{sk}[i]$, the probability that A generates $\mathbf{c}[i]$ beforehand is much less than $\frac{q_d}{q - q_d}$.

Case 2. $(u', v', w') = (u, v, w)$ and $e' \neq e$.

In this case, $\alpha' = \alpha$ and $u'^{a+\alpha' a'} v'^{b+\alpha' b'} = u^{a+\alpha a'} v^{b+\alpha b'} = e \neq e'$. Hence, $\mathbf{G}_{n+i-1}$ will not set bad.

Case 3. $(u', v', w') \neq (u, v, w)$ and $\alpha' = \alpha$.

Since Hash is a CR hash function, the probability that the adversary generates (u', v', w') such that $\mathsf{HEvl}(hk, (u', v', w')) = \mathsf{HEvl}(hk, (u, v, w))$ is negligible. Hence, Case 3 occurs with negligible probability.

Case 4. $(u', v', w') \neq (u, v, w)$ and $\alpha' \neq \alpha$.

Let $r'_1 := \log_{g_1} u'$ and $r'_2 := \log_{g_2} v'$. When $\mathbf{G}_{n+i-1}$ sets bad, $u'^z \neq v'$, which implies $r'_1 \neq r'_2$.

From the public key $\mathbf{pk}$ and the challenge ciphertext vector $\mathbf{c}$, for $\mathbf{sk}[i] = (x, y, a, b, a', b')$, all the information on (a, b, a', b') that A learns is:

$$\log_{g_1} \theta = a + bz, \tag{9}$$

$$\log_{g_1} \varphi = a' + b'z, \tag{10}$$

$$\log_{g_1} e = r_1 a + r_2 z b + r_1 \alpha a' + r_2 z \alpha b'. \tag{11}$$

For the decryption query $(i, (u', v', w', e'))$, $\mathbf{G}_{n+i-1}$ sets bad only if

$$\log_{g_1} e' = r'_1 a + r'_2 z b + r'_1 \alpha' a' + r'_2 z \alpha' b'. \tag{12}$$

Because

$$\begin{vmatrix} 1 & z & 0 & 0 \\ 0 & 0 & 1 & z \\ r_1 & r_2 z & r_1 \alpha & r_2 z \alpha \\ r'_1 & r'_2 z & r'_1 \alpha' & r'_2 z \alpha' \end{vmatrix} = z^2 (r_2 - r_1)(r'_2 - r'_1)(\alpha - \alpha') \neq 0,$$

equations (9)-(12) are linearly independent. Therefore, the probability that A submits a decryption query $(i, (u', v', w', e'))$ where $u'^z \neq v'$ *for the first time*, such that $\mathbf{G}_{n+i-1}$ sets bad, is $\frac{1}{q}$.

At best, each decryption query will help the adversary to eliminate one possible value, so the possibility that $\mathbf{G}_{n+i-1}$ sets bad in Case 4 is at most $\frac{q_d}{q-q_d}$, which is negligible. □

B Identity-based encryption

An identity-based encryption (IBE) scheme consists of four PPT algorithms ($\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}$, Dec). The parameter generation algorithm $\mathsf{PGen}\ (1^\lambda)$ outputs a public parameter pp and a master secret key msk. The private key generation algorithm $\mathsf{KGen}(pp, msk, id)$ takes pp, msk and an identity id as input, and outputs a secret key sk_{id} for id. The encryption algorithm $\mathsf{Enc}(pp, id, m)$ taking pp, id and a message m as input, outputs a ciphertext c. The decryption algorithm $\mathsf{Dec}(pp, sk_{id}, c)$, taking pp, sk_{id} and c as input, outputs a message m or $\perp$, which indicates that c is invalid. For correctness, we require that for any valid identity id and valid message m, $(pp, msk) \leftarrow \mathsf{PGen}(1^\lambda)$, $c \leftarrow \mathsf{Enc}(pp, id, m)$ and $sk_{id} \leftarrow \mathsf{KGen}(pp, msk, id)$, $\mathsf{Dec}(pp, sk_{id}, c) = m$ with overwhelming probability.

C Strong one-time signature

A signature scheme consists of three PPT algorithms $\mathsf{SIG} = (\mathsf{SGen}, \mathsf{Sign}, \mathsf{Verf})$. The key generation algorithm $\mathsf{SGen}(1^\lambda)$ outputs a signing/verification key pair $(sk_\mathsf{s}, vk_\mathsf{s})$. The signing algorithm $\mathsf{Sign}(sk_\mathsf{s}, m)$ taking sk_s and a message m as input, outputs a signature sg. The verification algorithm $\mathsf{Verf}(vk_\mathsf{s}, m, sg)$, taking vk_s, m and sg as input, returns $b \in \{0, 1\}$. For correctness, we require that for any valid message m, $(sk_\mathsf{s}, vk_\mathsf{s}) \leftarrow \mathsf{SGen}(1^\lambda)$ and $sg \leftarrow \mathsf{Sign}(sk_\mathsf{s}, m)$, $\mathsf{Verf}(vk_\mathsf{s}, m, sg) = 1$ with overwhelming probability. SIG is called *strong one-time*, if for any PPT adversary A, the advantage

$$\mathbf{Adv}^{\text{str-ot}}_{\mathsf{SIG},A}(\lambda) := \Pr\left[\begin{array}{ll} (sk_\mathsf{s}, vk_\mathsf{s}) \leftarrow \mathsf{SGen}(1^\lambda) & \\ m \leftarrow A(vk_\mathsf{s}) & \vdots\ (m', sg') \neq (m, sg) \bigwedge \\ sg \leftarrow \mathsf{Sign}(sk_\mathsf{s}, m) & \vdots\ \mathsf{Verf}(vk_\mathsf{s}, m, sg) = 1 \\ (m', sg') \leftarrow A(sg) & \end{array}\right]$$

is negligible.

D Identity-based key encapsulation mechanism

According to [5], an identity-based key encapsulation mechanism (IB-KEM) scheme for a session key space $\mathcal{K}$ consists of four PPT algorithms ($\mathsf{PGen}, \mathsf{KGen}, \mathsf{Encap}$, Decap). The parameter generation algorithm $\mathsf{PGen}(1^\lambda)$ outputs a public parameter pp and a master secret key msk. The private key generation algorithm $\mathsf{KGen}(pp, msk, id)$ takes pp, msk and an identity id as input, and outputs a secret key sk_{id} for id. The encapsulation algorithm $\mathsf{Encap}(pp, id)$ taking pp and id as input, outputs a session key $k \in \mathcal{K}$ and a corresponding ciphertext c. The decapsulation algorithm $\mathsf{Decap}(pp, sk_{id}, c)$, taking pp, sk_{id} and c as input, outputs a session key k or $\perp$, which indicates that c is invalid. For correctness, we require that for any valid identity id, $(pp, msk) \leftarrow \mathsf{PGen}(1^\lambda)$, $(k, c) \leftarrow \mathsf{Encap}(pp, id)$ and $sk_{id} \leftarrow \mathsf{KGen}(pp, msk, id)$, $\mathsf{Decap}(pp, sk_{id}, c) = k$ with overwhelming probability. For any blockcipher E, an IB-KEM is called *E-independent* if none of its four underlying algorithms invokes E in either direction.

The notion of IND-sID-CPA security for IB-KEM is very similar to IND-sID-CPA security for IBE. For constructions, any IND-sID-CPA secure IBE scheme (e.g., [7]) is an IB-KEM scheme achieving this security.