Designs, Codes and Cryptography (2023) 91:1233–1259
https://doi.org/10.1007/s10623-022-01147-8

Receiver selective opening security for identity-based encryption in the multi-challenge setting

Zhengan Huang[1] · **Junzuo Lai**[2] · **Gongxian Zeng**[1] · **Xin Mu**[1]

Received: 12 July 2022 / Revised: 12 October 2022 / Accepted: 25 October 2022 /
Published online: 17 November 2022

Abstract

Receiver selective opening (RSO) security requires that in a situation where there are one sender and multiple receivers, even if an adversary has access to all ciphertexts and adaptively corrupts some fraction of the receivers to obtain their secret keys, the (potentially related) ciphertexts of the uncorrupted receivers remain secure. All of the existing works construct RSO secure identity-based encryption (IBE) in the single-challenge setting, where each identity is used only once for encryption. This restriction makes RSO security for IBE unrealistic in practice. It is preferable to have IBE schemes with RSO security in the multi-challenge setting in practice, where each identity can be used to encrypt multiple messages. In this paper, we initiate the study of RSO security in the multi-challenge setting (which we call RSO_k security) for IBE. Concretely, we show that the conclusion of lower bound, proposed by Yang et al. (in: ASIACRYPT 2020, Springer, 2020), on the secret key size of RSO secure public-key encryption also holds in the IBE setting (i.e., an IBE scheme cannot be RSO_k secure if the length of its secret key is not k times larger than the length of message). For construction, we propose a generic construction of IBE achieving RSO_k security. Through our generic construction, we can obtain RSO_k secure IBE schemes based on decisional linear (DLIN) assumption and learning with error (LWE) assumption. Furthermore, we show that the well-known Fujisaki–Okamoto transformation can be applied to construct a practical IBE scheme achieving RSO_k security.

Communicated by K. Matsuura.

✉ Zhengan Huang
zhahuang.sjtu@gmail.com

Junzuo Lai
laijunzuo@gmail.com

Gongxian Zeng
gxzeng@cs.hku.hk

Xin Mu
mux@pcl.ac.cn

[1] Peng Cheng Laboratory, Shenzhen, China

[2] College of Information Science and Technology, Jinan University, Guangzhou, China

Keywords Identity-based encryption · Selective opening security · Multi-challenge setting · Chosen-ciphertext attack

Mathematics Subject Classification 94A60 · 68P25

1 Introduction

Selective opening attacks, firstly formally considered by Bellare et al. [3] for public key encryption (PKE), concern some multi-user scenarios, where an adversary is able to break into a subset of honestly generated ciphertexts and tries to learn information on the messages of some unopened (but maybe potentially related) ciphertexts. Bellare et al. [3] introduce two ways to formalize security notions against selective opening attacks (SO security notions), namely indistinguishability-based (IND-SO) security and simulation-based (SIM-SO) security. Generally, IND-SO security requires that the unopened ciphertexts and the ciphertexts of freshly sampled messages, which are distributed according to the conditional probability distribution (conditioned on the opened ciphertexts), are computationally indistinguishable; SIM-SO security requires that anything, which can be computed from all the ciphertexts and the opened messages together with the corrupted information, can also be computed only from the opened messages. Compared with SIM-SO security, IND-SO security has a limitation that the message distribution should be "efficiently conditionally re-samplable", while SIM-SO security imposes no limitation on the message distributions; it is already known that under chosen-plaintext attacks (CPA) for PKE, SIM-SO security is strictly stronger than IND-SO security [2, 6, 15]. Thus, for SO security, it is desirable to consider simulation-based definitions.

To date, SO security notions are usually considered in two settings: sender corruption [3, 9, 16, 19] and receiver corruption [2, 13, 15, 22]. In the sender corruption setting, part of the senders may be corrupted (we say that "their ciphertexts are opened"), exposing their messages and random coins employed during the encryption. In the receiver corruption setting, part of them may be corrupted, exposing their messages and secret keys. We denote SO security in the sender corruption setting and in the receiver corruption setting by SSO security and RSO security, respectively.

Standard RSO security is formalized in the single-challenge setting for PKE, where each public key is used only once to encrypt a single challenge message. This restriction makes RSO security unrealistic, since in practice a public key is often used multiple times. More realistic RSO_k security (i.e., RSO security in the multi-challenge setting, where each public key can be used to encrypt $k \geq 1$ challenge messages) is introduced by Yang et al. [42]. They prove that SIM-RSO security is not enough to guarantee SIM-RSO_k security ($k > 1$), providing a lower bound on the secret key length for any PKE scheme with RSO_k security in the non-programmable random oracle model, and show SIM-RSO_k-CPA/CCA secure PKE constructions with nearly optimal secret key length.

SO security for IBE The study of SO secure identity-based encryption (IBE) is initiated by Bellare et al. [5]. Compared with PKE, IBE allows a sender to generate ciphertexts using a receiver's identity as a public key, and the subtlety of proving security of IBE comes from the fact that a key generation oracle is provided to an adversary to answer private key queries with respect to different identities, and the adversary is free to choose the challenge identities. Bellare et al. [5] firstly formalize a simulation-based notion of SSO security under CPA attacks for IBE (which we denote as *SIM-ID-SSO-CPA security*), via adapting the SO

framework to IBE in a natural way (i.e., informally, replacing the public keys with the target receivers' identities, and allowing the adversary to access to the key generation oracle). Furthermore, they construct IBE schemes meeting this security requirement. Later, the first IBE scheme achieving simulation-based SSO security under CCA attacks (which we denote as *SIM-ID-SSO-CCA security*) is proposed by Lai et al. [30].

As for the receiver corruption setting, the notion of simulation-based RSO security under CPA attacks for IBE (which we denote as *SIM-ID-RSO-CPA security*) is formalized by Kitagawa and Tanaka [29]. They construct a SIM-ID-RSO-CPA secure IBE scheme from an IBE scheme with basic security (i.e., IND-ID-CPA security). Recently, Hara et al. [14] formalize the notion of simulation-based RSO security under CCA attacks for IBE (which we denote as *SIM-ID-RSO-CCA security*), and show an IBE construction meeting SIM-ID-RSO-CCA security via the classical double encryption technique [36, 38].

To the best of our knowledge, currently all the known receiver selective opening security notions for IBE [14, 29] are considered in the single-challenge setting (i.e., each identity as used only once to encrypt a single challenge message). However, in practice, usually an identity (i.e., public key) is used multiple times for encryption. More importantly, no RSO security notions in the multi-challenge setting for IBE have ever been considered before.

In this paper, we initiate the study of simulation-based RSO security in the multi-challenge setting for IBE.

Our contributions We firstly formalize the notion of simulation-based RSO security in the multi-challenge setting (which we denote as SIM-ID-RSO$_k$-CPA/CCA security) for IBE. In particular, a SIM-ID-RSO$_k$-CPA/CCA adversary is allowed to access to the key generation oracle, can obtain k challenge ciphertexts for each identity, and can corrupt some of the receivers and know the secret keys; SIM-ID-RSO$_k$-CPA/CCA security requires that anything, which can be computed by the adversary, can also be computed by a simulator only from the opened messages (of the corrupted receivers).

We show that the conclusion (proposed in [42]) of lower bound on the secret key size of RSO$_k$ secure encryption scheme in the PKE setting also holds in the IBE setting. More specifically, we provide a lower bound on the secret key length for any IBE scheme with SIM-ID-RSO$_k$ security in the non-programmable random oracle model. This result implies that for any SIM-ID-RSO$_k$ secure IBE scheme, assuming that the size of its message space (resp., secret key space) is $2^{\ell_{\mathrm{m}}}$ (resp., $2^{\ell_{\mathrm{sk}}}$), we have $\ell_{\mathrm{sk}} \geq \ell_{\mathrm{m}} k$. This result also implies that it is impossible for IBE to achieve SIM-ID-RSO$_k$ security without restricting the number of challenge ciphertexts for each identity (i.e., k).

We stress that this result also suggests that for IBE, RSO security in the single-challenge setting is not enough to guarantee that in the multi-challenge setting. That's because for any IBE scheme with SIM-ID-RSO-CCA security, assuming that the size of its message space (resp., secret key space) is $2^{\ell_{\mathrm{m}}}$ (resp., $2^{\ell_{\mathrm{sk}}}$), this IBE scheme is not SIM-ID-RSO$_k$-CPA secure for any $k \geq \frac{\ell_{\mathrm{sk}}+1}{\ell_{\mathrm{m}}}$.

We provide a generic construction of IBE achieving SIM-ID-RSO$_k$-CCA security. Our generic IBE scheme is constructed from an IND-ID-CPA secure IBE scheme with message space $\{0,1\}$ and a non-interactive zero-knowledge (NIZK) proof system, via the double encryption technique [36, 38]. More concretely, to encrypt a single-bit message m with identity id, the encryption algorithm of our generic IBE scheme proceeds as follows: (i) firstly uniformly sample k bits $\mathsf{m}_1, \ldots, \mathsf{m}_k$ satisfying $\mathsf{m}_1 \oplus \cdots \oplus \mathsf{m}_k = m$; (ii) for each $\eta \in [k]$ and each $\beta \in \{0,1\}$, generate $c_{\eta,\beta}$ by encrypting m_η (with identity $(\mathsf{id}, \eta, \beta)$) using the encryption algorithm of the underlying IND-ID-CPA secure IBE scheme; (iii) generate a NIZK proof indicating that the $2k$ ciphertexts $(c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}$ are created by encrypting $\mathsf{m}_1, \ldots, \mathsf{m}_k$ via (ii). With this method, the SIM-ID-RSO$_k$-CCA security of our generic IBE

can be implied by the IND-ID-CPA security of the underlying IBE and the security properties of the NIZK proof system.

In particular, when plugging the DLIN-based (resp., the LWE based) instantiations [12, 40] (resp., [1, 37]) into our generic construction, we obtain a concrete DLIN-based (resp., LWE based) SIM-ID-RSO$_k$-CCA secure IBE scheme.

We also provide a practical IBE scheme meeting SIM-ID-RSO$_k$-CCA security in the random oracle model. Specifically, we show that the well-known Fujisaki–Okamoto transformation [11] can be applied to construct SIM-ID-RSO$_k$-CCA secure IBE from a one-way secure (under CPA attacks) IBE scheme with high min-entropy of ciphertexts in the random oracle model. In other words, to encrypt a message m with identity id (and randomness r), the encryption algorithm computes as follows:

$$C := (\mathsf{Enc}(pp, \mathsf{id}, r; H(r, m \oplus G(r))), m \oplus G(r)),$$

where Enc is the encryption algorithm of the underlying IBE scheme (pp is the public parameter), and both G and H are hash functions.

Related works Since formalized by Bellare et al. in [3], PKE with SO security has been extensively studied. Numerous constructions of SSO (resp., RSO) secure PKE have been proposed based on various assumptions in previous works [8, 9, 16–19, 24, 25, 32, 33, 35] (resp., [13, 15, 22, 27–29, 34, 35, 42]). Recently, PKE achieving both SSO security and RSO security has been proposed in [31]. Relations among SSO/RSO security and standard security for PKE are also extensively studied in previous works [2, 6, 15, 20, 21, 23, 42].

Bellare et al. [5] initiate the study of SO security in the IBE setting, proposing a general paradigm to achieve SIM-ID-SSO-CPA security from IND-ID-CPA secure and "One-Sided Publicly Openable" (1SPO) IBE schemes. The first SIM-ID-SSO-CCA secure IBE scheme is constructed by Lai et al. [30]. In 2020, Jia et al. [26] present the first SIM-ID-SSO-CCA secure IBE scheme with tight security.

Kitagawa et al. [29] formalize the notion of SIM-ID-RSO-CPA security for IBE, and show that a SIM-ID-RSO-CPA secure IBE scheme can be constructed based only on an IND-ID-CPA secure IBE scheme. Hara et al. [14] formalize SIM-ID-RSO-CCA security for IBE, and show a generic construction of SIM-ID-RSO-CCA secure IBE from an IND-ID-CPA secure IBE scheme and a NIZK system satisfying unbounded simulation soundness and unbounded zero-knowledge property.

Roadmap We firstly recall some preliminaries in Sect. 2. Then, we present formal definitions of SIM-ID-RSO$_k$-CPA/CCA security ($k \geq 1$) for IBE in Sect. 3. We provide a lower bound for SIM-ID-RSO$_k$ secure IBE scheme in Sect. 4. Next, we show a generic construction of IBE achieving SIM-ID-RSO$_k$-CCA security in Sect. 5. Finally, we provide a practical SIM-ID-RSO$_k$-CCA secure IBE scheme in Sect. 6.

2 Preliminaries

Notations Let $\lambda \in \mathbb{N}$ denote the security parameter. For any $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. For a finite set S, we use $|S|$ to denote the size of S, and use $s \leftarrow S$ to denote the process of sampling s uniformly from S. For a distribution Dist, we use $x \leftarrow \mathsf{Dist}$ to denote the process of sampling x from Dist.

For a probabilistic algorithm $\mathcal{A}$, let $\mathcal{R}_{\mathcal{A}}$ denote the randomness space of $\mathcal{A}$. We use $y \leftarrow \mathcal{A}(x; r)$ denote the process of running $\mathcal{A}$ on input x and inner randomness $r \leftarrow \mathcal{R}_{\mathcal{A}}$ and outputting y. We write $y \leftarrow \mathcal{A}(x)$ for $y \leftarrow \mathcal{A}(x; r)$ with uniformly chosen $r \in \mathcal{R}_{\mathcal{A}}$. We write

PPT for probabilistic polynomial-time. For a function $f(\lambda)$, we write that $f(\lambda) \leq \mathsf{negl}(\lambda)$ if it is negligible.

NIZK proof system Let R be an efficiently computable binary relation, and $\mathsf{L} := \{x \mid \exists w \text{ s.t. } (x, w) \in \mathsf{R}\}$. A NIZK proof NIZK for L consists of the following three algorithms:

- $\mathsf{CRSGen}(1^\lambda)$: On input the security parameter 1^λ, the common reference string (CRS) generation algorithm outputs a CRS crs.
- $\mathsf{Prove}(crs, x, w)$: The proving algorithm, taking a CRS crs, a statement $x \in \mathsf{L}$ and a witness w for the fact that $x \in \mathsf{L}$ as input, outputs a proof π.
- $\mathsf{Verify}(crs, x, \pi)$: The verification algorithm, taking a CRS crs, a statement $x \in \mathsf{L}$ and a proof π as input, outputs a bit $b \in \{0, 1\}$.

It also satisfies the following conditions:

– **Completeness** For all $\lambda \in \mathbb{N}$, all crs generated by CRSGen and all $(x, w) \in \mathsf{R}$, we always have $\mathsf{Verify}(crs, x, \mathsf{prove}(crs, x, w)) = 1$.

– **Unbounded Zero-knowledge** There is a PPT simulator $\mathcal{S}^{(\mathrm{zk})} = (\mathcal{S}_1^{(\mathrm{zk})}, \mathcal{S}_2^{(\mathrm{zk})})$ such that for any PPT adversary $\mathcal{A}$,

$$\mathbf{Adv}^{\mathrm{zk}}_{\mathsf{NIZK},\mathcal{A},\mathcal{S}^{(\mathrm{zk})}}(\lambda) := |\Pr[\mathbf{G}^{\text{zk-real}}_{\mathsf{NIZK},\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{G}^{\text{zk-sim}}_{\mathsf{NIZK},\mathcal{A},\mathcal{S}^{(\mathrm{zk})}}(\lambda) = 1]|$$

is negligible, where $\mathbf{G}^{\text{zk-real}}_{\mathsf{NIZK},\mathcal{A}}(\lambda)$ and $\mathbf{G}^{\text{zk-sim}}_{\mathsf{NIZK},\mathcal{A},\mathcal{S}^{(\mathrm{zk})}}(\lambda)$ are shown in Fig. 1.

– **Unbounded simulation soundness** Let $\mathcal{S}^{(\mathrm{zk})} = (\mathcal{S}_1^{(\mathrm{zk})}, \mathcal{S}_2^{(\mathrm{zk})})$ be a PPT simulator for the unbounded zero-knowledge property of NIZK. For any unbounded adversary $\mathcal{A}$, the advantage

$$\mathbf{Adv}^{\mathrm{sound}}_{\mathsf{NIZK},\mathcal{A},\mathcal{S}^{(\mathrm{zk})}}(\lambda) := \Pr[\mathbf{G}^{\mathrm{sound}}_{\mathsf{NIZK},\mathcal{A},\mathcal{S}^{(\mathrm{zk})}}(\lambda) = 1]$$

is negligible, where $\mathbf{G}^{\mathrm{sound}}_{\mathsf{NIZK},\mathcal{A},\mathcal{S}^{(\mathrm{zk})}}(\lambda)$ is shown in Fig. 1.

Identity-based encryption An identity-based encryption (IBE) scheme consists of four PPT algorithms $(\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$. The setup algorithm $\mathsf{Setup}(1^\lambda)$ outputs a public parameter pp and a master secret key msk. The private key generation algorithm $\mathsf{KGen}(pp, msk, \mathsf{id})$ takes pp, msk and an identity id as input, and outputs a secret key sk_{id} for id. The encryption algorithm $\mathsf{Enc}(pp, \mathsf{id}, m)$ taking pp, id and a message $m \in \mathcal{M}_{\mathrm{sp}}$ as input, outputs a ciphertext c, where $\mathcal{M}_{\mathrm{sp}}$ is the message space. The decryption algorithm $\mathsf{Dec}(pp, sk_{\mathsf{id}}, c)$, 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 (pp, msk) generated by Setup, any valid identity id and any valid message m, $\mathsf{Dec}(pp, \mathsf{KGen}(pp, msk, \mathsf{id}), \mathsf{Enc}(pp, \mathsf{id}, m)) = m$ with overwhelming probability.

Now we recall notions of OW-ID-CPA security and IND-ID-CPA security for IBE [7]. Note that the following recalled definition of IND-ID-CPA security considers multiple challenge identities, like [14]. This security notion is equivalent to the original one proposed in [7] except for a polynomial reduction loss based on the number of challenge identities.

Definition 1 (*OW-ID-CPA*) We say that an IBE scheme $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is OW-ID-CPA secure, if for any PPT adversary $\mathcal{A}$,

$$\mathbf{Adv}^{\text{ow-id-cpa}}_{\mathsf{IBE},\mathcal{A}}(\lambda) := \Pr[\mathbf{G}^{\text{ow-id-cpa}}_{\mathsf{IBE},\mathcal{A}}(\lambda) = 1]$$

is negligible, where $\mathbf{G}^{\text{ow-id-cpa}}_{\mathsf{IBE},\mathcal{A}}(\lambda)$ is defined in Fig. 2.

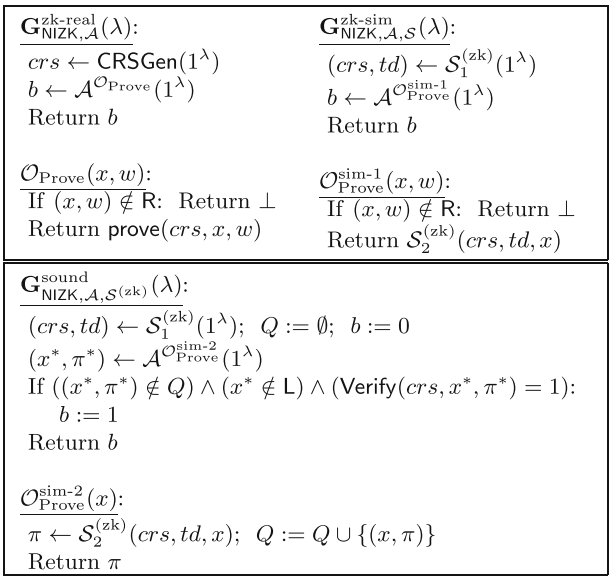

Fig. 1 Games for defining zero-knowledge property and simulation-soundness of NIZK

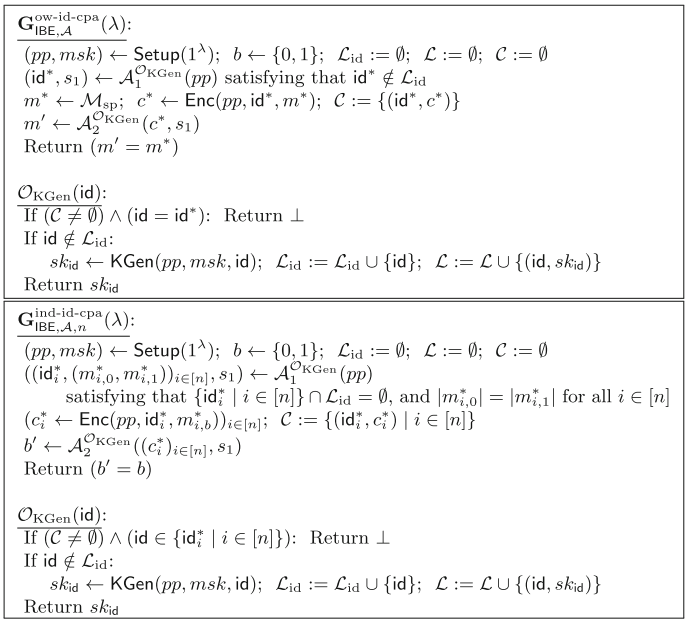

Fig. 2 Games for defining OW-ID-CPA security (in Definition 1) and IND-ID-CPA security (in Definition 2) of IBE

Definition 2 (*IND-ID-CPA*) We say that an IBE scheme $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-ID-CPA secure, if for any polynomially bounded $n > 0$ and any PPT adversary $\mathcal{A}$,

$$\mathbf{Adv}^{\text{ind-id-cpa}}_{\mathsf{IBE},\mathcal{A},n}(\lambda) := \Pr[\mathbf{G}^{\text{ind-id-cpa}}_{\mathsf{IBE},\mathcal{A},n}(\lambda) = 1]$$

is negligible, where $\mathbf{G}^{\text{ind-id-cpa}}_{\mathsf{IBE},\mathcal{A},n}(\lambda)$ is defined in Fig. 2.

3 RSO security in the multi-challenge setting for IBE

In this section, we introduce simulation-based receiver selective opening security in the multi-challenge setting for IBE, which we call SIM-ID-RSO$_k$-CPA/CCA security ($k \geq 1$).

Definition 3 (*SIM-ID-RSO$_k$-CPA/CCA security*) We say that an IBE scheme $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is SIM-ID-RSO$_k$-ATK secure (ATK $\in \{\text{CPA}, \text{CCA}\}$), if for any polynomially bounded $n > 0$ and any PPT adversary $\mathcal{A}$, there exists a PPT simulator $\mathcal{S}$, such that for any PPT distinguisher $\mathcal{D}$,

$$\begin{aligned}\mathbf{Adv}^{\text{rso}_k\text{-atk}}_{\mathsf{IBE},\mathcal{A},\mathcal{S},\mathcal{D},n}(\lambda) := |&\Pr[\mathcal{D}(\mathbf{G}^{\text{rso}_k\text{-atk-real}}_{\mathsf{IBE},\mathcal{A},n}(\lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathbf{G}^{\text{rso}_k\text{-atk-ideal}}_{\mathsf{IBE},\mathcal{S},n}(\lambda)) = 1]|\end{aligned}$$

is negligible, where $\mathbf{G}^{\text{rso}_k\text{-atk-real}}_{\mathsf{IBE},\mathcal{A},n}(\lambda)$ and $\mathbf{G}^{\text{rso}_k\text{-atk-ideal}}_{\mathsf{IBE},\mathcal{S},n}(\lambda)$ are both defined in Fig. 3, and atk $\in \{\text{cpa}, \text{cca}\}$.

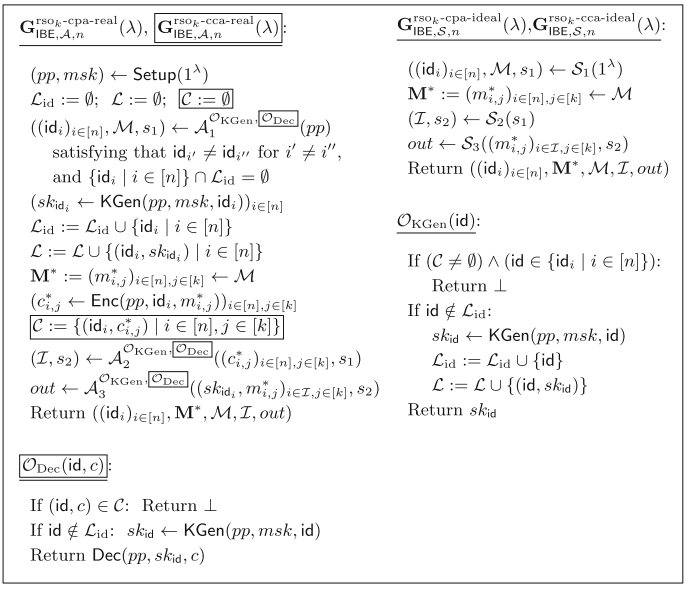

Fig. 3 Games for defining SIM-ID-RSO$_k$-CPA security and SIM-ID-RSO$_k$-CCA security (in Definition 3) for IBE, where $\mathcal{I} \subset [n]$. Boxed code is only executed in the games specified by the game names in the same box style

In game $\mathbf{G}^{\text{rso}_k\text{-atk-real}}_{\mathsf{IBE},\mathcal{A},n}(\lambda)$ (atk $\in$ {cpa, cca}), we use $\mathcal{L}_{\text{id}}$ to denote the identities whose secret keys have been generated, and use $\mathcal{L}$ to denote these identities and the corresponding secret keys.

4 Lower bound for IBE with RSO$_k$ security

In this section, we show a lower bound for IBE with SIM-ID-RSO$_k$ ($k \geq 1$) security. The idea is inspired by the work of Yang et al. [42]. Generally speaking, we show that the conclusion of lower bound (proposed in [42]) on the secret key size of RSO$_k$ secure encryption scheme in the PKE setting also holds in the IBE setting, i.e., assuming that the secret key space is $\{0,1\}^{\ell_{\text{sk}}}$ and the message space is $\{0,1\}^{\ell_{\text{m}}}$ for some $\ell_{\text{sk}}, \ell_{\text{m}} \in \mathbb{N}$, an IBE scheme cannot be SIM-ID-RSO$_k$-CPA secure if the length of its secret key is not k times larger than the length of message.

Formally, we have the following theorem.

Theorem 1 *Let* $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an IBE scheme with secret key space* $\mathcal{SK}_{\text{sp}}$ *and message space* $\mathcal{M}_{\text{sp}}$*, where* $|\mathcal{SK}_{\text{sp}}| = 2^{\ell_{\text{sk}}}$ *and* $|\mathcal{M}_{\text{sp}}| = 2^{\ell_{\text{m}}}$*. If* $\ell_{\text{sk}} \leq \ell_{\text{m}}k - 1$*, then* IBE *is not SIM-ID-RSO$_k$-CPA secure in the non-programmable random oracle model.*

According to Theorem 1, even if IBE is SIM-ID-RSO-CCA secure, when $\ell_{\text{sk}} < \ell_{\text{m}}k$, it is *not* SIM-ID-RSO$_k$-CPA secure in the non-programmable random oracle model. In other words, in the IBE setting, RSO security in the single-challenge setting is not enough to guarantee RSO security in the multi-challenge setting.

Next, we turn to the proof of Theorem 1.

Proof Let $H : \{0,1\}^* \to \{0,1\}^{\ell_{\text{h}}}$ be a hash function, which will be modeled as a non-programmable random oracle in this proof. We write $\mathcal{PP}_{\text{sp}}$, $\mathcal{ID}_{\text{sp}}$ and $\mathcal{C}_{\text{sp}}$ to denote the public parameter space, the identity space and the ciphertext space of IBE respectively. Let $\ell_{\text{pp}} := \lceil \log |\mathcal{PP}_{\text{sp}}| \rceil$, and $\ell_{\text{c}} := \lceil \log |\mathcal{C}_{\text{sp}}| \rceil$, and $\kappa := \ell_{\text{pp}} + \ell_{\text{c}}k + 2$. Let $n := \ell_{\text{h}} + 1$.

Now, we construct a SIM-ID-RSO$_k$-CPA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ and a distinguisher $\mathcal{D}$ as shown in Fig. 4. Note that we require that $\mathcal{A}$ should not query oracle $\mathcal{O}_{\text{KGen}}$, so we omit oracle $\mathcal{O}_{\text{KGen}}$ in Fig. 4.

Correctness of IBE guarantees that

$$\Pr[\mathcal{D}(\mathbf{G}^{\text{rso}_k\text{-cpa-real}}_{\mathsf{IBE},\mathcal{A},n}(\lambda)) = 1] \leq \mathsf{negl}(\lambda). \tag{1}$$

For any fixed PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$, without loss of generality, we assume that $\mathcal{S}_2$ and $\mathcal{S}_3$ are both deterministic, i.e., the random coins are sampled by $\mathcal{S}_1$ and then given to $\mathcal{S}_2$ and $\mathcal{S}_3$ via states s_1 and s_2 respectively. Let

$$\delta_{\mathcal{S}} = \Pr[\mathcal{D}(\mathbf{G}^{\text{rso}_k\text{-cpa-ideal}}_{\mathsf{IBE},\mathcal{S},n}(\lambda)) = 1].$$

Following the idea of [42], if we can show that $\delta_{\mathcal{S}} \geq \frac{1}{2\kappa}$, then we obtain that

$$\begin{aligned}
&\mathbf{Adv}^{\text{rso}_k\text{-cca}}_{\mathsf{IBE},\mathcal{A},\mathcal{S},\mathcal{D},n}(\lambda) \\
&= |\Pr[\mathcal{D}(\mathbf{G}^{\text{rso}_k\text{-cca-real}}_{\mathsf{IBE},\mathcal{A},n}(\lambda)) = 1] - \Pr[\mathcal{D}(\mathbf{G}^{\text{rso}_k\text{-cca-ideal}}_{\mathsf{IBE},\mathcal{S},n}(\lambda)) = 1]| \\
&\geq \frac{1}{2\kappa} - \mathsf{negl}(\lambda),
\end{aligned}$$

which is non-negligible. So what remains is to prove $\delta_{\mathcal{S}} \geq \frac{1}{2\kappa}$.

$\underline{\mathcal{A}_1(pp):}$
$\mathcal{M} := \mathcal{U}_{\ell_{\mathrm{m}} nk}$; $(\mathsf{id}_i)_{i\in[n]} \leftarrow (\mathcal{ID}_{\mathrm{sp}})^n$ satisfying that $\mathsf{id}_{i'} \neq \mathsf{id}_{i''}$ for $i' \neq i''$
$s_1 := (pp, (\mathsf{id}_i)_{i\in[n]})$
Return $((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1)$

$\underline{\mathcal{A}_2((c^*_{i,j})_{i\in[n],j\in[k]}, s_1):}$
Parse $s_1 = (pp, (\mathsf{id}_i)_{i\in[n]})$
$t := H(pp, (\mathsf{id}_i, c^*_{i,j})_{i\in[n],j\in[k]}))$; $\mathcal{I} := \{i \mid (i \in [\ell_{\mathrm{h}}]) \wedge (t[i] = 1)\} \cup \{n\}$
$s_2 := (pp, (\mathsf{id}_i)_{i\in[n]}, (c^*_{i,j})_{i\in[n],j\in[k]})$
Return $(\mathcal{I}, s_2)$

$\underline{\mathcal{A}_3((sk_{\mathsf{id}_i}, m^*_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2):}$
Parse $s_2 = (pp, (\mathsf{id}_i)_{i\in[n]}, (c^*_{i,j})_{i\in[n],j\in[k]})$
$out := (pp, (c^*_{i,j})_{i\in[n],j\in[k]}, (sk_{\mathsf{id}_i})_{i\in\mathcal{I}})$
Return out

$\underline{\mathcal{D}((\mathsf{id}_i)_{i\in[n]}, (m^*_{i,j})_{i\in[n],j\in[k]}, \mathcal{M}, \mathcal{I}, out):}$
Parse $out = (pp, (c^*_{i,j})_{i\in[n],j\in[k]}, (sk_{\mathsf{id}_i})_{i\in\mathcal{I}})$
If $\mathcal{M} \neq \mathcal{U}_{\ell_{\mathrm{m}} nk}$: Return 1
$t := H(pp, (\mathsf{id}_i, c^*_{i,j})_{i\in[n],j\in[k]}))$; $\mathcal{I}' = \{i \mid (i \in [\ell_{\mathrm{h}}]) \wedge (t[i] = 1)\} \cup \{n\}$
If $\mathcal{I}' \neq \mathcal{I}$: Return 1
For $i \in \mathcal{I}, j \in [k]$:
  If $\mathsf{Dec}(pp, sk_{\mathsf{id}_i}, c^*_{i,j}) \neq m^*_{i,j}$: Return 1
Return 0

Fig. 4 Adversary $\mathcal{A}$ and distinguisher $\mathcal{D}$ in attacking SIM-ID-RSO$_k$-CPA security of IBE. We omit oracle $\mathcal{O}_{\mathrm{KGen}}$ here, since we require that $\mathcal{A}$ should not query $\mathcal{O}_{\mathrm{KGen}}$. $\mathcal{U}_{\ell_{\mathrm{m}} nk}$ denotes a uniform distribution over $\{0,1\}^{\ell_{\mathrm{m}} nk}$

$\underline{\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda):}$
$((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1) \leftarrow \mathcal{S}_1(1^\lambda)$; $(\mathcal{I}, s_2) \leftarrow \mathcal{S}_2(s_1)$; $r_{\mathcal{D}} \leftarrow \mathcal{R}_{\mathcal{D}}$
For $\theta \in [\kappa]$:
  $(m^{(\theta)}_{i,j})_{i\in[n],j\in[k]} \leftarrow \mathcal{M}$; $out^{(\theta)} \leftarrow \mathcal{S}_3((m^{(\theta)}_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2)$
  If $\mathcal{D}((\mathsf{id}_i)_{i\in[n]}, (m^{(\theta)}_{i,j})_{i\in[n],j\in[k]}, \mathcal{M}, \mathcal{I}, out) = 1$: Return 1
For $\theta \in \{2, \cdots, \kappa\}$:
  Parse $out^{(\theta)} = (pp^{(\theta)}, (c^{(\theta)}_{i,j})_{i\in[n],j\in[k]}, (sk^{(\theta)}_{\mathsf{id}_i})_{i\in\mathcal{I}})$
  If $(pp^{(\theta-1)}, (c^{(\theta-1)}_{i,j})_{i\in[n],j\in[k]}) \neq (pp^{(\theta)}, (c^{(\theta)}_{i,j})_{i\in[n],j\in[k]})$: Return 2
Return 0

Fig. 5 The auxiliary game $\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda)$

Let $r_{\mathcal{D}}$ denote the randomness of $\mathcal{D}$ (this randomness is used in the decryption algorithm of IBE). Consider an auxiliary game $\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda)$ as shown in Fig. 5.

We postpone the proofs of the following three lemmas.

Lemma 1 $\Pr[\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda) = 0] \leq \frac{1}{4}$.

Lemma 2 $\Pr[\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda) = 1] \leq \kappa \cdot \delta_{\mathcal{S}}$.

Lemma 3 $\Pr[\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda) = 2] \leq \frac{1}{4}$.

Combining the three lemmas, we obtain

$$1 \leq \frac{1}{4} + \kappa \cdot \delta_{\mathcal{S}} + \frac{1}{4},$$

which implies

$$\delta_{\mathcal{S}} \geq \frac{1}{2\kappa},$$

finishing the proof of this theorem.

Now, we turn to the proofs of the above three lemmas.

Proof of Lemma 1 Assume that $\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda) = 0$. Then,

- we have $\mathcal{M} = \mathcal{U}_{\ell_{\mathrm{m}} nk}$ (for simplicity, we denote this event as evt_1), which means that for each $\theta \in [\kappa]$, $i \in [n]$ and $j \in [k]$, $m^{(\theta)}_{i,j}$ is uniformly random sampled from $\{0,1\}^{\ell_{\mathrm{m}}}$;
- we have $(pp^{(\theta-1)}, (c^{(\theta-1)}_{n,j})_{j\in[k]}) = (pp^{(\theta)}, (c^{(\theta)}_{n,j})_{j\in[k]})$ for all $\theta \in \{2,\ldots,\kappa\}$ (we denote this event as evt_2), so we can write pp as $pp^{(\theta)}$, and $(c_{n,j})_{j\in[k]}$ as $(c^{(\theta)}_{n,j})_{j\in[k]}$;
- we have $m^{(\theta)}_{n,j} = \mathsf{Dec}(pp, sk^{(\theta)}_{\mathsf{id}_n}, c_{n,j})$ for all $\theta \in [\kappa]$ and all $j \in [k]$ (we denote this event as evt_3).

Obviously, we derive

$$\Pr[\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda) = 0] \leq \Pr[\mathsf{evt}_1 \vee \mathsf{evt}_2 \vee \mathsf{evt}_3].$$

Note that if fixing any tuple $(pp, (c_{n,1},\ldots,c_{n,k}), (sk^{(1)}_{\mathsf{id}_n},\ldots,sk^{(\kappa)}_{\mathsf{id}_n}))$, which is not necessary the output of $\mathcal{S}_3$, then we have

$$\Pr[\forall\, \theta \in [\kappa],\ j \in [k],\ m^{(\theta)}_{n,j} = \mathsf{Dec}(pp, sk^{(\theta)}_{\mathsf{id}_n}, c_{n,j})] = \frac{1}{2^{\ell_{\mathrm{m}} k\kappa}},$$

where the probability is taken over the random choice of each $m^{(\theta)}_{n,j}$ ($\theta \in [\kappa]$, $j \in [k]$).

We also note that total number of possible $(pp, (c_{n,1},\ldots,c_{n,k}), (sk^{(1)}_{\mathsf{id}_n},\ldots,sk^{(\kappa)}_{\mathsf{id}_n}))$ is at most $2^{\ell_{\mathrm{pp}}+\ell_{\mathrm{c}}k+\ell_{\mathrm{sk}}\kappa} = 2^{(\ell_{\mathrm{sk}}+1)\kappa-2}$. Hence,

$$\begin{aligned}
&\Pr[\mathsf{evt}_1 \vee \mathsf{evt}_2 \vee \mathsf{evt}_3] \\
&\leq \Pr[\exists\ (pp, (c_{n,1},\ldots,c_{n,k}), (sk^{(1)}_{\mathsf{id}_n},\ldots,sk^{(\kappa)}_{\mathsf{id}_n})) : \\
&\quad \forall\, \theta \in [\kappa],\ j \in [k],\ m^{(\theta)}_{n,j} = \mathsf{Dec}(pp, sk^{(\theta)}_{\mathsf{id}_n}, c_{n,j})] \\
&\leq \frac{2^{(\ell_{\mathrm{sk}}+1)\kappa-2}}{2^{\ell_{\mathrm{m}} k\kappa}} \leq \frac{2^{((\ell_{\mathrm{m}}k-1)+1)\kappa-2}}{2^{\ell_{\mathrm{m}} k\kappa}} = \frac{1}{4}.
\end{aligned}$$

So we obtain

$$\Pr[\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda) = 0] \leq \Pr[\mathsf{evt}_1 \vee \mathsf{evt}_2 \vee \mathsf{evt}_3] \leq \frac{1}{4}.$$

□

Proof of Lemma 2 Note that the randomness of $\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda)$ comes from the randomness of $\mathcal{S}$ (i.e., $r_{\mathcal{S}} \leftarrow \mathcal{R}_{\mathcal{S}}$), $r_{\mathcal{D}} \leftarrow \mathcal{R}_{\mathcal{D}}$ and $(m^{(\theta)}_{i,j})_{i\in[n],j\in[k]} \leftarrow \mathcal{M}$. Let

$$\begin{aligned}
f(r_{\mathcal{D}}, r_{\mathcal{S}}) := \Pr[&((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1) = \mathcal{S}_1(1^{\lambda}; r_{\mathcal{S}});\ (\mathcal{I}, s_2) = \mathcal{S}_2(s_1); \\
&(m_{i,j})_{i\in[n],j\in[k]} \leftarrow \mathcal{M};\ out = \mathcal{S}_3((m_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2) : \\
&\mathcal{D}((\mathsf{id}_i)_{i\in[n]}, (m_{i,j})_{i\in[n],j\in[k]}, \mathcal{M}, \mathcal{I}, out) = 1]
\end{aligned}$$

where the probability is taken over the random choice of $(m_{i,j})_{i\in[n],j\in[k]}$. Letting $\mathbb{E}_{x\leftarrow X}(x) := \sum_x \Pr[X = x]\cdot x$ denote the expectation of the random variable X, we have

$$\begin{aligned}\Pr[\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda)=1] &= \mathbb{E}_{r_{\mathcal{D}}\leftarrow\mathcal{R}_{\mathcal{D}},r_{\mathcal{S}}\leftarrow\mathcal{R}_{\mathcal{S}}}(1-(1-f(r_{\mathcal{D}},r_{\mathcal{S}}))^{\kappa})\\ &\le \mathbb{E}_{r_{\mathcal{D}}\leftarrow\mathcal{R}_{\mathcal{D}},r_{\mathcal{S}}\leftarrow\mathcal{R}_{\mathcal{S}}}(\kappa\cdot f(r_{\mathcal{D}},r_{\mathcal{S}}))\\ &= \kappa\cdot\mathbb{E}_{r_{\mathcal{D}}\leftarrow\mathcal{R}_{\mathcal{D}},r_{\mathcal{S}}\leftarrow\mathcal{R}_{\mathcal{S}}}(f(r_{\mathcal{D}},r_{\mathcal{S}}))\\ &= \kappa\cdot\delta_{\mathcal{S}}.\end{aligned}$$

□

Proof of Lemma 3 Denote the number of queries to H by q_H. Since H is modeled as a non-programmable random oracle, the probability that there are two distinct queries x_1, x_2 satisfying that $H(x_1) = H(x_2)$ is less than $\frac{q_H^2}{2^{\ell_{\mathrm{h}}}}$, which is negligible.

If $\Pr[\mathbf{G}^{\mathrm{aux}}_{\mathsf{IBE},\mathcal{S},\mathcal{D},n,k,\kappa}(\lambda)=2] > \frac{1}{4}$, then via running this game, one can find $\theta\in\{2,\ldots,\kappa\}$ satisfying that

(1) $(pp^{(\theta-1)}, (c_{i,j}^{(\theta-1)})_{i\in[n],j\in[k]}) \neq (pp^{(\theta)}, (c_{i,j}^{(\theta)})_{i\in[n],j\in[k]})$;
(2) $H(pp^{(\theta-1)}, (\mathsf{id}_i, c_{i,j}^{(\theta-1)})_{i\in[n],j\in[k]}) = H(pp^{(\theta)}, (\mathsf{id}_i, c_{i,j}^{(\theta)})_{i\in[n],j\in[k]}) = (t[1],\ldots,t[h])$, where $t[i]=1$ if and only if $i\in\mathcal{I}$,

with probability greater than $\frac{1}{4}$, which is obviously non-negligible, contradicting the collision resistant property of H. □

Remark 1 As pointed out in [4] and restated in [42], impossibility result in the non-programmable random oracle model does not extend to that in the standard model naturally, since the adversary in the non-programmable random oracle model is allowed to query the random oracle and thus is stronger than an adversary in the standard model. So Theorem 1 does not rule out the achievability of SIM-RSO$_k$ secure IBE when $\ell_{\mathrm{sk}} \le \ell_{\mathrm{m}}k - 1$ in the standard model.

Similar to [42], the proof of Theorem 1 can be modified to achieve the same lower bound in the standard model, but only in the auxiliary-input model. More specifically, the modified proof is the same as the proof of Theorem 1, except that (1) H is a collision-resistant (CR) hash function, instead of being modeled as a non-programmable random oracle, and (2) the proof is given in the auxiliary input model, i.e., all participants, including $\mathcal{A}$, $\mathcal{D}$ and $\mathcal{S}$, are all given some common auxiliary input (a random key of the CR hash function) at the beginning.

5 Generic construction of SIM-ID-RSO$_k$-CCA secure IBE

In this section, we show a generic construction of IBE, achieving SIM-ID-RSO$_k$-CCA security, based on an IND-ID-CPA secure IBE scheme and a non-interactive zero-knowledge (NIZK) proof system, via the double encryption technique [36, 38].

Generic construction Let $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-ID-CPA secure IBE scheme with an identity space $\mathcal{ID}\times[k]\times\{0,1\}$ and a message space $\{0,1\}$, for some set $\mathcal{ID}$. Let $\mathsf{NIZK} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Verify})$ be a NIZK proof system for language

$$\begin{aligned}\{(pp, \mathsf{id}, (c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}) \mid \exists(\mathsf{m}_\eta, r_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}} \text{ s.t.}\\ (c_{\eta,\beta} = \mathsf{Enc}(pp, (\mathsf{id},\eta,\beta), \mathsf{m}_\eta; r_{\eta,\beta}))_{\eta\in[k],\beta\in\{0,1\}}\}.\end{aligned}$$

$\underline{\mathsf{Setup}'(1^\lambda):}$
$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$; $crs \leftarrow \mathsf{CRSGen}(1^\lambda)$; $PP := (pp, crs)$; $MSK := msk$
Return (PP, MSK)

$\underline{\mathsf{KGen}'(PP, MSK, id):}$
Parse $PP = (pp, crs)$, and $MSK = msk$
For $\eta \in [k]$:
  $\alpha_\eta \leftarrow \{0,1\}$; $sk_{\mathsf{id},\eta,\alpha_\eta} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}, \eta, \alpha_\eta))$
Return $SK_{\mathsf{id}} := ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$

$\underline{\mathsf{Enc}'(PP, \mathsf{id}, m):}$
Parse $PP = (pp, crs)$
$\mathsf{m}_1, \cdots, \mathsf{m}_k \leftarrow \{0,1\}$ satisfying $\mathsf{m}_1 \oplus \cdots \oplus \mathsf{m}_k = m$
For $\eta \in [k]$:
  For $\beta \in \{0,1\}$:
    $r_{\eta,\beta} \leftarrow \mathcal{R}_{\mathsf{Enc}}$; $c_{\eta,\beta} = \mathsf{Enc}(pp, (\mathsf{id}, \eta, \beta), \mathsf{m}_\eta; r_{\eta,\beta})$
$\pi \leftarrow \mathsf{Prove}(crs, (pp, \mathsf{id}, (c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}), ((\mathsf{m}_\eta, r_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}))$
Return $C := ((c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}, \pi)$

$\underline{\mathsf{Dec}'(PP, SK_{\mathsf{id}}, C):}$
Parse $PP = (pp, crs)$, $SK_{\mathsf{id}} = ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$, and $C = ((c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}, \pi)$
$x := (pp, \mathsf{id}, (c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}})$
If $\mathsf{Verify}(crs, x, \pi) = 0$: Return $\perp$
$(\mathsf{m}_\eta := \mathsf{Dec}(pp, sk_{\mathsf{id},\eta,\alpha_\eta}, c_{\eta,\alpha_\eta}))_{\eta\in[k]}$
Return $m := \mathsf{m}_1 \oplus \cdots \oplus \mathsf{m}_k$

Fig. 6 IBE scheme $\mathsf{IBE}' = (\mathsf{Setup}', \mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$

We construct an IBE scheme $\mathsf{IBE}' = (\mathsf{Setup}', \mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ with the identity space $\mathcal{ID}$ and the message space $\{0, 1\}$ as in Fig. 6.

The correctness of IBE' is straightforward guaranteed by the correctness of IBE and the completeness of NIZK. Now we turn to the security analysis. Formally, we have the following theorem.

Theorem 2 *If* IBE *is an IND-ID-CPA secure IBE scheme, and* NIZK *is a NIZK proof system satisfying unbounded zero-knowledge property and unbounded simulation soundness, then* IBE' *is SIM-ID-RSO$_k$-CCA secure.*

Proof For any polynomial $n > 0$, any PPT adversary $\mathcal{A}$ and any PPT distinguisher $\mathcal{D}$, we consider the following games $\mathbf{G}_0$–$\mathbf{G}_5$.

Game $\mathbf{G}_0$: This game is exactly the same as $\mathbf{G}^{\text{rso}_k\text{-cca-real}}_{\mathsf{IBE}',\mathcal{A},n}(\lambda)$. Specifically, the challenger interacts with $\mathcal{A}$ as follows.

(1) The challenger firstly computes $(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$ and $crs \leftarrow \mathsf{CRSGen}(1^\lambda)$, and sets $PP := (pp, crs)$ and $MSK := msk$. Then, it prepares five sets $\mathcal{L}_{\mathsf{id}} := \emptyset$, $\mathcal{L} := \emptyset$, $\mathcal{L}' := \emptyset$, $\mathcal{L}_{\mathsf{chal}} := \emptyset$ and $\mathcal{C} := \emptyset$, and sends PP to $\mathcal{A}_1$. The challenger answers $\mathcal{A}_1$'s oracle queries as follows:

- $\mathcal{O}_{\mathrm{KGen}}(\mathsf{id})$: Since $\mathcal{C} = \emptyset$, the challenger directly checks whether id belongs to $\mathcal{L}_{\mathsf{id}}$ or not.
  - If $\mathsf{id} \notin \mathcal{L}_{\mathsf{id}}$, then for each $\eta \in [k]$, it samples $\alpha_\eta \leftarrow \{0,1\}$, and generates $sk_{\mathsf{id},\eta,\alpha_\eta} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}, \eta, \alpha_\eta))$. Then it sets that $SK_{\mathsf{id}} := ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$, appends id (resp., $(\mathsf{id}, SK_{\mathsf{id}})$) to $\mathcal{L}_{\mathsf{id}}$ (resp., $\mathcal{L}$), and returns SK_{id} to $\mathcal{A}_1$.
  - If $\mathsf{id} \in \mathcal{L}_{\mathsf{id}}$, which means that there is some $(\mathsf{id}, SK_{\mathsf{id}}) \in \mathcal{L}$, then it returns SK_{id} to $\mathcal{A}_1$.
- $\mathcal{O}_{\mathrm{Dec}}(\mathsf{id}, C)$: The challenger checks whether id belongs to $\mathcal{L}_{\mathsf{id}}$.

– If $\mathsf{id} \notin \mathcal{L}_{\mathsf{id}}$, then for each $\eta \in [k]$, it samples $\alpha_\eta \leftarrow \{0,1\}$, and generates $sk_{\mathsf{id},\eta,\alpha_\eta} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}, \eta, \alpha_\eta))$. Then it sets that $SK_{\mathsf{id}} := ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$.
– If $\mathsf{id} \in \mathcal{L}_{\mathsf{id}}$, which means that there is some $(\mathsf{id}, SK_{\mathsf{id}}) \in \mathcal{L}$, then it retrieves SK_{id} from $\mathcal{L}$.

The challenger parse $SK_{\mathsf{id}} = ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$ and $C = ((c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}, \pi)$, and sets that $x := (pp, \mathsf{id}, (c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}})$. If $\mathsf{Verify}(crs, x, \pi) = 0$, it returns $\perp$ to $\mathcal{A}_1$; otherwise, it computes $\mathsf{m}_\eta := \mathsf{Dec}(pp, sk_{\mathsf{id},\eta,\alpha_\eta}, c_{\eta,\alpha_\eta})$ for each $\eta \in [k]$, and then returns $m := \mathsf{m}_1 \oplus \cdots \oplus \mathsf{m}_k$ to $\mathcal{A}_1$.

(2) Receiving $((\mathsf{id}_i)_{i\in[n]}, \mathcal{M})$ from $\mathcal{A}_1$, the challenger proceeds as follows. For each $i \in [n]$, it firstly generates SK_{id_i}. In particular, for each $i \in [n]$ and each $\eta \in [k]$, it samples $\alpha_{i,\eta} \leftarrow \{0,1\}$, and computes $sk_{\mathsf{id}_i,\eta,\alpha_{i,\eta}} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}_i, \eta, \alpha_{i,\eta}))$. Then, for each $i \in [n]$, the challenger sets that $SK_{\mathsf{id}_i} := ((\alpha_{i,\eta}, sk_{\mathsf{id}_i,\eta,\alpha_{i,\eta}})_{\eta\in[k]}, \mathsf{id}_i)$, and appends id_i (resp., $(\mathsf{id}_i, SK_{\mathsf{id}_i})$) to $\mathcal{L}_{\mathsf{id}}$ (resp., $\mathcal{L}$). After that, it samples $(m^*_{i,j})_{i\in[n],j\in[k]} \leftarrow \mathcal{M}$. For each $i \in [n]$ and $j \in [k]$, the challenger generates a challenge ciphertext $C^*_{i,j}$ for $m^*_{i,j}$ as follows.

(a) It firstly samples $\mathsf{m}_{i,j,\eta} \leftarrow \{0,1\}$ for each $\eta \in [k]$ and $\eta \neq j$, and sets $\mathsf{m}_{i,j,j} := (\oplus_{\eta\in[k]\wedge\eta\neq j} \mathsf{m}_{i,j,\eta}) \oplus m^*_{i,j}$.
(b) For each $\eta \in [k]$ and each $\beta \in \{0,1\}$, the challenger samples $r_{i,j,\eta,\beta} \leftarrow \mathcal{R}_{\mathsf{Enc}}$, and computes $c_{i,j,\eta,\beta} = \mathsf{Enc}(pp, (\mathsf{id}_i, \eta, \beta), \mathsf{m}_{i,j,\eta}; r_{i,j,\eta,\beta})$.
(c) The challenger computes $\pi_{i,j} \leftarrow \mathsf{Prove}(crs, (pp, \mathsf{id}_i, (c_{i,j,\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}), ((\mathsf{m}_{i,j,\eta}, r_{i,j,\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}))$.
(d) It sets $C^*_{i,j} := ((c_{i,j,\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}, \pi_{i,j})$.

(3) The challenger appends $(\mathsf{id}_i, C^*_{i,j})$ to $\mathcal{C}$ for each $i \in [n]$ and each $j \in [k]$, and returns $(C^*_{i,j})_{i\in[n],j\in[k]}$ to $\mathcal{A}_2$. Then, it answers $\mathcal{A}_2$'s oracle queries as follows:

- $\mathcal{O}_{\mathrm{KGen}}(\mathsf{id})$: If $\mathsf{id} \in \{\mathsf{id}_i \mid i \in [n]\}$, the challenger returns $\perp$ to $\mathcal{A}_2$ directly. Otherwise, it answers this query as that in Step (1).
- $\mathcal{O}_{\mathrm{Dec}}(\mathsf{id}, C)$: The challenger firstly checks whether $(\mathsf{id}, C) \in \mathcal{C}$. If so, it returns $\perp$ to $\mathcal{A}_2$ directly. Otherwise, it checks whether id belongs to $\mathcal{L}_{\mathsf{id}}$ or not.
  – If $\mathsf{id} \notin \mathcal{L}_{\mathsf{id}}$, then for each $\eta \in [k]$, it samples $\alpha_\eta \leftarrow \{0,1\}$, and generates $sk_{\mathsf{id},\eta,\alpha_\eta} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}, \eta, \alpha_\eta))$. Then it sets that $SK_{\mathsf{id}} := ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$.
  – If $\mathsf{id} \in \mathcal{L}_{\mathsf{id}}$, which means that there is some $(\mathsf{id}, SK_{\mathsf{id}}) \in \mathcal{L}$, then it retrieves SK_{id} from $\mathcal{L}$.

  The challenger parse $SK_{\mathsf{id}} = ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$ and $C = ((c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}, \pi)$, and sets that $x := (pp, \mathsf{id}, (c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}})$. If $\mathsf{Verify}(crs, x, \pi) = 0$, it returns $\perp$ to $\mathcal{A}_2$; otherwise, it computes $\mathsf{m}_\eta := \mathsf{Dec}(pp, sk_{\mathsf{id},\eta,\alpha_\eta}, c_{\eta,\alpha_\eta})$ for each $\eta \in [k]$, and then returns $m := \mathsf{m}_1 \oplus \cdots \oplus \mathsf{m}_k$ to $\mathcal{A}_2$.

(4) Receiving $\mathcal{I} \subset [n]$ from A_2, the challenger returns $(SK_{\mathsf{id}_i}, m^*_{i,j})_{i\in[n],j\in[k]}$ to $\mathcal{A}_3$, and answers $\mathcal{A}_3$'s oracle queries exactly as in Step (3).
(5) Finally, upon receiving $\mathcal{A}_3$'s final output out, the challenger outputs $((\mathsf{id}_i)_{i\in[n]}, (m^*_{i,j})_{i\in[n],j\in[k]}, \mathcal{M}, \mathcal{I}, out)$.

Game $\mathbf{G}_1$: This game is the same as $\mathbf{G}_0$, except that when generating the CRS and the proofs, the challenger employs the simulator $\mathcal{S}^{(\mathrm{zk})} = (\mathcal{S}^{(\mathrm{zk})}_1, \mathcal{S}^{(\mathrm{zk})}_2)$ for the unbounded zero-knowledge property of NIZK, instead of generating them honestly. More specifically, in Step

(1), the challenger computes $(crs, td) \leftarrow \mathcal{S}_1^{(\mathrm{zk})}(1^\lambda)$; for each $i \in [n]$ and $j \in [k]$, in (c) of Step (2), the challenger computes $\pi_{i,j} \leftarrow \mathcal{S}_2^{(\mathrm{zk})}(crs, td, (pp, \mathsf{id}_i, (c_{i,j,\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}))$.

The unbounded zero-knowledge property of NIZK guarantees that

$$|\Pr[\mathcal{D}(\mathbf{G}_1) = 1] - \Pr[\mathcal{D}(\mathbf{G}_0) = 1]| \leq \mathbf{Adv}^{\mathrm{zk}}_{\mathsf{NIZK},\mathcal{A}_{\mathrm{zk}},\mathcal{S}^{(\mathrm{zk})}}(\lambda) \tag{2}$$

for some adversary $\mathcal{A}_{\mathrm{zk}}$.

Game $\mathbf{G}_2$: This game is the same as $\mathbf{G}_1$, except for the generation of the challenge ciphertexts. More specifically, for each $i \in [n]$ and $j \in [k]$, the challenger computes $c_{i,j,j,1\oplus\alpha_{i,j}} \leftarrow \mathsf{Enc}(pp, (\mathsf{id}_i, j, 1 \oplus \alpha_{i,j}), 1 \oplus \mathsf{m}_{i,j,j})$ instead of $c_{i,j,j,1\oplus\alpha_{i,j}} \leftarrow \mathsf{Enc}(pp, (\mathsf{id}_i, j, 1 \oplus \alpha_{i,j}), \mathsf{m}_{i,j,j})$ in (b) of Step (2). Note that this change can be seen as letting $\mathsf{m}'_{i,j,j} := \mathsf{m}_{i,j,j} \oplus \alpha_{i,j} = ((\oplus_{\eta\in[k]\wedge\eta\neq j}\mathsf{m}_{i,j,\eta}) \oplus m^*_{i,j}) \oplus \alpha_{i,j}$, the challenger computes

$$c_{i,j,j,0} \leftarrow \mathsf{Enc}(pp, (\mathsf{id}_i, j, 0), \mathsf{m}'_{i,j,j})$$
$$c_{i,j,j,1} \leftarrow \mathsf{Enc}(pp, (\mathsf{id}_i, j, 1), 1 \oplus \mathsf{m}'_{i,j,j})$$

for each $i \in [n]$ and $j \in [k]$ in (b) of Step (2).

Now we show that $\mathbf{G}_2$ and $\mathbf{G}_1$ are computationally indistinguishable. Note that in these two games, for each $i \in [n]$, (i) the challenger only uses $SK_{\mathsf{id}_i} = ((\alpha_{i,\eta}, sk_{\mathsf{id}_i,\eta,\alpha_{i,\eta}})_{\eta\in[k]}, \mathsf{id}_i)$ to answer $\mathcal{A}$'s decryption queries and selective opening queries, and (ii) if $\mathcal{A}_2$ or $\mathcal{A}_3$ submits id_i to $\mathcal{O}_{\mathrm{KGen}}$, the challenger will return $\perp$ as a response.[1] In other words, $sk_{\mathsf{id}_i,\eta,1\oplus\alpha_{i,\eta}}$ for any $i \in [n]$ and any $\eta \in [k]$ will never be used. So for all $i \in [n]$ and $\eta \in [k]$, $sk_{\mathsf{id}_i,\eta,1\oplus\alpha_{i,\eta}}$ is hidden from the view of $\mathcal{A}$. Thus, IND-ID-CPA security of IBE guarantees that

$$|\Pr[\mathcal{D}(\mathbf{G}_2) = 1] - \Pr[\mathcal{D}(\mathbf{G}_1) = 1]| \leq \mathbf{Adv}^{\text{ind-id-cpa}}_{\mathsf{IBE},\mathcal{A}_{\mathrm{ibe}},nk}(\lambda) \tag{3}$$

for some adversary $\mathcal{A}_{\mathrm{ibe}}$.

Game $\mathbf{G}_3$: This game is the same as $\mathbf{G}_2$, except for the generation of secret keys for the challenge identities $(\mathsf{id}_i)_{i\in[n]}$ in Step (2). More specifically, in this game, when generating a secret key for id_i ($i \in [n]$) in Step (2), besides generating $SK_{\mathsf{id}_i} = ((\alpha_{i,\eta}, sk_{\mathsf{id}_i,\eta,\alpha_{i,\eta}})_{\eta\in[k]}, \mathsf{id}_i)$, the challenger additionally proceeds as follows:

(i) Compute $sk_{\mathsf{id}_i,\eta,1\oplus\alpha_{i,\eta}} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}_i, \eta, 1 \oplus \alpha_{i,\eta}))$ for each $\eta \in [k]$, set that $SK'_{\mathsf{id}_i} := ((1 \oplus \alpha_{i,\eta}, sk_{\mathsf{id}_i,\eta,1\oplus\alpha_{i,\eta}})_{\eta\in[k]}, \mathsf{id}_i)$, and append $(\mathsf{id}_i, SK'_{\mathsf{id}_i})$ into $\mathcal{L}'$.

(ii) Append $(\mathsf{id}_i, (sk_{\mathsf{id}_i,\eta,0}, sk_{\mathsf{id}_i,\eta,1})_{\eta\in[k]})$ into $\mathcal{L}_{\mathrm{chal}}$.

Note that this change does not affect the view of $\mathcal{A}$. Hence,

$$\Pr[\mathcal{D}(\mathbf{G}_3) = 1] = \Pr[\mathcal{D}(\mathbf{G}_2) = 1]. \tag{4}$$

Game $\mathbf{G}_4$: This game is the same as $\mathbf{G}_3$, except for the way to answer $\mathcal{A}$'s key generation queries. More specifically, upon receiving a key generation query id to $\mathcal{O}_{\mathrm{KGen}}$, besides generating $SK_{\mathsf{id}} = ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$, the challenger additionally generates $SK'_{\mathsf{id}} = ((1 \oplus \alpha_\eta, sk_{\mathsf{id},\eta,1\oplus\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$ and appends $(\mathsf{id}, SK'_{\mathsf{id}})$ into $\mathcal{L}'$. Note that this is a conceptual change, so it does not affect the view of $\mathcal{A}$. Thus,

$$\Pr[\mathcal{D}(\mathbf{G}_4) = 1] = \Pr[\mathcal{D}(\mathbf{G}_3) = 1]. \tag{5}$$

We also note that in this and subsequent games, for any id and any $\eta \in [k]$, if $sk_{\mathsf{id},\eta,\alpha_\eta}$ has been stored in $\mathcal{L}$, and both $sk_{\mathsf{id},\eta,0}$ and $sk_{\mathsf{id},\eta,1}$ have been generated and can be retrieved.

[1] Note that $(\mathsf{id}_i)_{i\in[n]}$ are specified by $\mathcal{A}_1$, and they are required to satisfy that $\{\mathsf{id}_i \mid i \in [n]\} \cap \mathcal{L}_{\mathsf{id}} = \emptyset$. So $\mathcal{A}_1$ cannot obtain secret keys for $(\mathsf{id}_i)_{i\in[n]}$ via querying $\mathcal{O}_{\mathrm{KGen}}$.

Game $\mathbf{G}_5$: This game is the same as $\mathbf{G}_4$, except for the way to answer $\mathcal{A}$'s decryption queries. More precisely, for any decryption query (id, C), the challenger will use $sk_{\mathsf{id},\eta,0}$ instead of $sk_{\mathsf{id},\eta,\alpha_\eta}$ to answer this query.

The unbounded simulation soundness of NIZK guarantees that

$$|\Pr[\mathcal{D}(\mathbf{G}_5) = 1] - \Pr[\mathcal{D}(\mathbf{G}_4) = 1]| \leq \mathbf{Adv}^{\text{sound}}_{\mathsf{NIZK},\mathcal{A}'_{\text{zk}},\mathcal{S}^{(\text{zk})}}(\lambda) \tag{6}$$

for some adversary $\mathcal{A}'_{\text{zk}}$.

Game $\mathbf{G}_6$: This game is the same as $\mathbf{G}_5$, except for the way to generate the challenge ciphertexts and answer the selective opening query. More specifically,

- When generating the challenge ciphertexts, the challenger samples $\mathsf{m}'_{i,j,j} \leftarrow \{0,1\}$ instead of setting $\mathsf{m}'_{i,j,j} := ((\oplus_{\eta\in[k]\wedge\eta\neq j}\mathsf{m}_{i,j,\eta}) \oplus m^*_{i,j}) \oplus \alpha_{i,j}$ for each $i \in [n]$ and $j \in [k]$ in (a) of Step (2).
- The challenger does not sample $\alpha_{i,j}$ for any $i \in [n]$ in Step (2) (note that both $sk_{\mathsf{id}_i,j,0}$ and $sk_{\mathsf{id}_i,j,1}$ for all $i \in [n]$ and all $j \in [k]$ have been generated and stroed in $\mathcal{L}_{\text{chal}}$ since the change introduced in game $\mathbf{G}_3$). Instead, when answering the selective opening query $\mathcal{I} \subset [n]$ in Step (4), the challenger sets that $\alpha_{i,j} := ((\oplus_{\eta\in[k]\wedge\eta\neq j}\mathsf{m}_{i,j,\eta}) \oplus m^*_{i,j}) \oplus \mathsf{m}'_{i,j,j}$ for all $i \in \mathcal{I}$ and $j \in [k]$.

The only difference between $\mathbf{G}_6$: and $\mathbf{G}_5$ is the order of generations of $\mathsf{m}'_{i,j,j}$ and $\alpha_{i,j}$ for some i and some j, and it is easy to see that this difference does not affect $\mathcal{A}$'s view in these two games. Hence,

$$\Pr[\mathcal{D}(\mathbf{G}_6) = 1] = \Pr[\mathcal{D}(\mathbf{G}_5) = 1]. \tag{7}$$

Now, we construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$, which simulates $\mathbf{G}_6$ perfectly for $\mathcal{A}$, as shown in Fig. 7. Obviously,

$$\mathbf{G}^{\text{rso}_k\text{-cca-ideal}}_{\mathsf{IBE},\mathcal{S},n}(\lambda) = \mathbf{G}_6. \tag{8}$$

Note that $\mathbf{G}^{\text{rso}_k\text{-cca-real}}_{\mathsf{IBE}',\mathcal{A},n}(\lambda) = \mathbf{G}_0$, so combining Eqs. (2)–(8), we obtain that

$$\begin{aligned}
&\mathbf{Adv}^{\text{rso}_k\text{-cca}}_{\mathsf{IBE}',\mathcal{A},\mathcal{S},\mathcal{D},n}(\lambda)\\
&= |\Pr[\mathcal{D}(\mathbf{G}^{\text{rso}_k\text{-cca-real}}_{\mathsf{IBE}',\mathcal{A},n}(\lambda)) = 1] - \Pr[\mathcal{D}(\mathbf{G}^{\text{rso}_k\text{-cca-ideal}}_{\mathsf{IBE}',\mathcal{S},n}(\lambda)) = 1]|\\
&= |\Pr[\mathcal{D}(\mathbf{G}_0) = 1] - \Pr[\mathcal{D}(\mathbf{G}_6) = 1]|\\
&\leq \mathbf{Adv}^{\text{zk}}_{\mathsf{NIZK},\mathcal{A}_{\text{zk}},\mathcal{S}^{(\text{zk})}}(\lambda) + \mathbf{Adv}^{\text{ind-id-cpa}}_{\mathsf{IBE},\mathcal{A}_{\text{ibe}},n}(\lambda) + \mathbf{Adv}^{\text{sound}}_{\mathsf{NIZK},\mathcal{A}'_{\text{zk}},\mathcal{S}^{(\text{zk})}}(\lambda),
\end{aligned}$$

for some adversaries $\mathcal{A}_{\text{zk}}$, $\mathcal{A}_{\text{ibe}}$ and $\mathcal{A}'_{\text{zk}}$. □

Instantiations Note that the NIZK proof system satisfying unbounded zero-knowledge property and unbounded simulation soundness can be constructed based on an ordinary NIZK proof system (i.e., a NIZK proof system satisfying standard zero-knowledge property and standard soundness) via the transformation of [39] without any additional assumption. Hence, when plugging a DLIN-based (resp., LWE-based) IND-ID-CPA secure IBE scheme [40] (resp., [1]) and a DLIN-based (resp., LWE-based) ordinary NIZK proof system [12] (resp., [37]) into our generic construction, we can obtain a concrete DLIN-based (resp., LWE-based) SIM-ID-RSO$_k$-CCA secure IBE scheme.

$\underline{\mathcal{S}_1(1^\lambda):}$
$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$; $(crs, td) \leftarrow \mathcal{S}_1^{(\mathrm{zk})}(1^\lambda)$; $PP := (pp, crs)$; $MSK := msk$
$\mathcal{L}_{\mathrm{id}} := \emptyset$; $\mathcal{L} := \emptyset$; $\mathcal{L}' := \emptyset$; $\mathcal{L}_{\mathrm{chal}} := \emptyset$; $\mathcal{C} := \emptyset$; $((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathrm{KGen}}, \mathcal{O}_{\mathrm{Dec}}}(PP)$
For $i \in [n]$:
  For $\eta \in [k]$:
    $sk_{\mathsf{id}_i,\eta,0} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}_i, \eta, 0))$; $sk_{\mathsf{id}_i,\eta,1} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}_i, \eta, 1))$
$\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}_i \mid i \in [n]\}$; $\mathcal{L}_{\mathrm{chal}} := \mathcal{L}_{\mathrm{chal}} \cup \{(\mathsf{id}_i, (sk_{\mathsf{id}_i,\eta,0}, sk_{\mathsf{id}_i,\eta,1})_{\eta\in[k]}) \mid i \in [n]\}$
$s_1' := (PP, (\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, td, (sk_{\mathsf{id}_i,\eta,\beta})_{i\in[n],\eta\in[k],\beta\in\{0,1\}}, \mathcal{L}_{\mathrm{id}}, \mathcal{L}, \mathcal{L}', \mathcal{L}_{\mathrm{chal}}, \mathcal{C}, s_1)$
Return $((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1')$

$\underline{\mathcal{S}_2(s_1'):}$
For $i \in [n]$:
  For $j \in [k]$:
    For $(\eta \in [k]) \wedge (\eta \neq j)$:
      $\mathsf{m}_{i,j,\eta} \leftarrow \{0,1\}$; $c_{i,j,\eta,0} \leftarrow \mathsf{Enc}(pp, (\mathsf{id}_i, \eta, 0), \mathsf{m}_{i,j,\eta})$; $c_{i,j,\eta,1} \leftarrow \mathsf{Enc}(pp, (\mathsf{id}_i, \eta, 1), \mathsf{m}_{i,j,\eta})$
    $\mathsf{m}_{i,j,j} \leftarrow \{0,1\}$; $c_{i,j,j,0} \leftarrow \mathsf{Enc}(pp, (\mathsf{id}_i, \eta, 0), \mathsf{m}_{i,j,j})$; $c_{i,j,j,1} \leftarrow \mathsf{Enc}(pp, (\mathsf{id}_i, \eta, 1), 1 \oplus \mathsf{m}_{i,j,j})$
    $\pi_{i,j} \leftarrow \mathcal{S}_2^{(\mathrm{zk})}(crs, td, (pp, \mathsf{id}_i, (c_{i,j,\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}))$; $C^*_{i,j} := ((c_{i,j,\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}, \pi_{i,j})$
$\mathcal{C} := \{(\mathsf{id}_i, C^*_{i,j}) \mid i \in [n], j \in [k]\}$; $(\mathcal{I}, s_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathrm{KGen}}, \mathcal{O}_{\mathrm{Dec}}}((C^*_{i,j})_{i\in[n],j\in[k]}, s_1)$
$s_2' := (PP, (\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, td, (sk_{\mathsf{id}_i,\eta,\beta})_{i\in[n],\eta\in[k],\beta\in\{0,1\}}, \mathcal{L}_{\mathrm{id}}, \mathcal{L}, \mathcal{L}', \mathcal{L}_{\mathrm{chal}}, \mathcal{C}, (C^*_{i,j})_{i\in[n],j\in[k]}, s_2)$
Return $(\mathcal{I}, s_2')$

$\underline{\mathcal{S}_3((m^*_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2'):}$
For $i \in \mathcal{I}$:
  For $j \in [k]$:
    $\alpha_{i,j} := (\oplus_{\eta\in[k]} \mathsf{m}_{i,j,\eta}) \oplus m^*_{i,j}$
  $SK_{\mathsf{id}_i} := ((\alpha_{i,\eta}, sk_{\mathsf{id}_i,\eta,\alpha_{i,\eta}})_{\eta\in[k]}, \mathsf{id}_i)$
$out \leftarrow \mathcal{A}_3^{\mathcal{O}_{\mathrm{KGen}}, \mathcal{O}_{\mathrm{Dec}}}((SK_{\mathsf{id}_i}, m^*_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2)$
Return out

$\underline{\mathcal{O}_{\mathrm{KGen}}(\mathsf{id}):}$
If $(\mathcal{C} \neq \emptyset) \wedge (\mathsf{id} \in \{\mathsf{id}_i \mid i \in [n]\})$: Return $\perp$
If $\mathsf{id} \notin \mathcal{L}_{\mathrm{id}}$:
  For $\eta \in [k]$:
    $\alpha_\eta \leftarrow \{0,1\}$; $sk_{\mathsf{id},\eta,\alpha_\eta} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}, \eta, \alpha_\eta))$
  $SK_{\mathsf{id}} := ((\alpha_\eta, sk_{\mathsf{id},\eta,\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$; $SK'_{\mathsf{id}} = ((1 \oplus \alpha_\eta, sk_{\mathsf{id},\eta,1\oplus\alpha_\eta})_{\eta\in[k]}, \mathsf{id})$
  $\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}, SK_{\mathsf{id}})\}$; $\mathcal{L}' := \mathcal{L}' \cup \{(\mathsf{id}, SK'_{\mathsf{id}})\}$
Return SK_{id}

$\underline{\mathcal{O}_{\mathrm{Dec}}(\mathsf{id}, C):}$
Parse $C = ((c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}, \pi)$
If $\mathsf{id} \notin \mathcal{L}_{\mathrm{id}}$: $(sk_{\mathsf{id},\eta,0} \leftarrow \mathsf{KGen}(pp, msk, (\mathsf{id}, \eta, 0)))_{\eta\in[k]}$
If $\mathsf{Verify}(crs, (pp, \mathsf{id}, (c_{\eta,\beta})_{\eta\in[k],\beta\in\{0,1\}}), \pi) = 0$: Return $\perp$
$(\mathsf{m}_\eta := \mathsf{Dec}(pp, sk_{\mathsf{id},\eta,0}, c_{\eta,0}))_{\eta\in[k]}$
Return $m := \mathsf{m}_1 \oplus \cdots \oplus \mathsf{m}_k$

Fig. 7 Simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ in the proof of Theorem 2

6 Practical SIM-ID-RSO$_k$-CCA secure IBE scheme

Although we have proposed a generic IBE construction achieving SIM-ID-RSO$_k$-CCA security in Sect. 5, it is inefficient in practical applications. In this section, we show a practical IBE construction achieving SIM-ID-RSO$_k$-CCA security in the random oracle model. More specifically, as shown in [41], the Fujisaki–Okamoto transformation [10] can be applied to

$\underline{\mathsf{FOSetup}(1^\lambda):}$
$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda);\ \ PP := pp;\ \ MSK := msk$
Return (PP, MSK)

$\underline{\mathsf{FOKGen}(PP, MSK, id):}$
Parse $PP = pp$, and $MSK = msk$
$sk_{\mathsf{id}} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id})$
Return $SK_{\mathsf{id}} := sk_{\mathsf{id}}$

$\underline{\mathsf{FOEnc}(PP, \mathsf{id}, m):}$
Parse $PP = pp$
$r \leftarrow \mathcal{R}_{\mathsf{FOEnc}};\ \ c := m \oplus G(r);\ \ h := H(r, c);\ \ e := \mathsf{Enc}(pp, \mathsf{id}, r; h)$
Return $C := (e, c)$

$\underline{\mathsf{FODec}(PP, SK_{\mathsf{id}}, C):}$
Parse $PP = pp$, $SK_{\mathsf{id}} = sk_{\mathsf{id}}$, and $C = (e, c)$
$\hat{r} := \mathsf{Dec}(pp, sk_{\mathsf{id}}, e)$
If $\hat{r} \notin \mathcal{R}_{\mathsf{FOEnc}}$: Return $\perp$
$\hat{h} := H(\hat{r}, c)$
If $e \neq \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$: Return $\perp$
Return $m := c \oplus G(\hat{r})$

Fig. 8 IBE scheme FOIBE = (FOSetup, FOKGen, FOEnc, FODec)

the IBE case, generally converting an IBE scheme with OW-ID-CPA security (and high min-entropy of ciphertexts) into an IBE scheme with IND-ID-CCA security in the random oracle model. Now we show that via the Fujisaki–Okamoto transformation [11], the obtained IBE scheme actually achieves SIM-ID-RSO$_k$-CCA security.

Firstly, we introduce a property of IBE in the following definition, which is extended directly from γ-spread PKE [10, 11].

Definition 4 (γ-*spread*) Let IBE = (Setup, KGen, Enc, Dec) be an IBE scheme with identity space $\mathcal{ID}$, message space $\mathcal{M}_{\mathsf{sp}}$, ciphertext space $\mathcal{CT}_{\mathsf{sp}}$ and randomness space $\mathcal{R}_{\mathsf{Enc}}$. For any pp generated by Setup, any $\mathsf{id} \in \mathcal{ID}$, any $m \in \mathcal{M}_{\mathsf{sp}}$ and any $c \in \mathcal{CT}_{\mathsf{sp}}$,

$$\Pr[r \leftarrow \mathcal{R}_{\mathsf{Enc}} : c = \mathsf{Enc}(pp, \mathsf{id}, m; r)] \leq 2^{-\gamma}.$$

Let IBE = (Setup, KGen, Enc, Dec) be an IBE scheme with an identity space $\mathcal{ID}$, a message space $\mathcal{M}_{\mathsf{sp}}$ and a randomness space $\mathcal{R}_{\mathsf{Enc}}$. Consider the IBE scheme FOIBE = (FOSetup, FOKGen, FOEnc, FODec) as shown in Fig. 8, with a message space $\mathcal{M}_{\mathsf{FO}} := \{0, 1\}^\ell$, for some $\ell \in \mathbb{N}$, and a randomness space $\mathcal{R}_{\mathsf{FOEnc}} := \mathcal{M}_{\mathsf{sp}}$. Note that the underlying $G : \mathcal{R}_{\mathsf{FOEnc}} \to \{0, 1\}^\ell$ and $H : \mathcal{R}_{\mathsf{FOEnc}} \times \{0, 1\}^\ell \to \mathcal{R}_{\mathsf{Enc}}$ in Fig. 8. are both hash functions, which will be modeled as random oracles in the security proof.

The correctness of FOIBE is obviously guaranteed by the correctness of IBE. For security, we have the following theorem.

Theorem 3 *If* IBE *is an OW-ID-CPA secure, γ-spread IBE scheme, and both G and H are modeled as random oracles, then* FOIBE *is SIM-ID-RSO$_k$-CCA secure in the random oracle model.*

Before going into the formal proof, we firstly show an intuition of why FOIBE achieves SIM-ID-RSO$_k$-CCA security. For a normally generated ciphertext $C = (e, c)$, we have $c = m \oplus G(r)$ and $e = \mathsf{Enc}(pp, \mathsf{id}, r; H(r, c))$, where $r \leftarrow \mathcal{R}_{\mathsf{FOEnc}}$. So in the ciphertext C, only c contains the information about the message m, and m is concealed by $G(r)$. Note that as long as r has never been queried to the random oracle $\mathcal{O}_{\mathrm{G}}$, the adversary $\mathcal{A}$ has no information about m (since $\mathcal{O}_{\mathrm{G}}(r)$ is uniformly distributed from $\mathcal{A}$'s point of view).

Furthermore, note that r is the input "plaintext" of the underlying encryption algorithm Enc, so the one-wayness of IBE guarantees that $\mathcal{A}$ cannot find out the value of r, which implies that c is uniformly distributed from $\mathcal{A}$'s point of view. Hence, if r has never been queried to the random oracles, then both c and $h = H(r, c)$ are uniformly distributed, and $e = \mathsf{Enc}(pp, \mathsf{id}, r; h)$ is independent of m. In this case, in the proof, the challenge ciphertexts can be generated firstly without knowing the challenge messages, and then when answering the selective opening query, the challenge ciphertexts and the corresponding challenge messages are correlated via programming the random oracles. That's how we deal with the selective opening query in the proof. For the decryption query, via the technique of the Fujisaki–Okamoto transformation [11], the properties of the random oracles implies that the decryption oracle can be simulated without the secret keys (i.e., the modification introduced in the following game $\mathbf{G}_1$).

The formal proof is as follows.

Proof For any polynomial $n > 0$, any PPT adversary $\mathcal{A}$ and any PPT distinguisher $\mathcal{D}$, let q_d (resp. q_r) denote the total number of decryption queries (resp. random-oracle queries) made by $\mathcal{A}$. Without loss of generality, we require that the challenger samples the random coins $(r_{i,j} \leftarrow \mathcal{R}_{\mathsf{FOEnc}})_{i\in[n],\, j\in[k]}$ before sending the public parameter PP to $\mathcal{A}$. We also assume that $\mathcal{A}$ will not repeat identical queries to the same oracles.

Since both G and H are modeled as random oracles, we assume that the challenger maintains lists L_G and L_H, which are both empty sets at the beginning, and employs them to keep track of the issued calls (either by the game or $\mathcal{A}$) of $\mathcal{O}_G(t)$ and $\mathcal{O}_H(u_1, u_2)$, respectively. Specifically, for a query t submitted to $\mathcal{O}_G$, $\mathcal{O}_G$ returns g_t if there is an entry $(t, g_t) \in L_G$, otherwise it samples $g_t \leftarrow \{0,1\}^\ell$, adds (t, g_t) to L_G, and returns g_t; similarly, for a query (u_1, u_2) submitted to $\mathcal{O}_H$, $\mathcal{O}_H$ returns h_u if there is an entry $((u_1, u_2), h_u) \in L_H$, otherwise it samples $h_u \leftarrow \mathcal{R}_{\mathsf{Enc}}$, adds $((u_1, u_2), h_u)$ to L_H, and returns h_u.

We proceed in a series of games.

Game $\mathbf{G}_0$: $\mathbf{G}_0$ (as shown in Fig. 9) is the real game $\mathbf{G}^{\mathrm{rso}_k\text{-cca-real}}_{\mathsf{FOIBE},\mathcal{A},n}(\lambda)$, i.e.,

$$\mathbf{G}_0 = \mathbf{G}^{\mathrm{rso}_k\text{-cca-real}}_{\mathsf{FOIBE},\mathcal{A},n}(\lambda). \tag{9}$$

Game $\mathbf{G}_1$: Game $\mathbf{G}_1$ is the same as $\mathbf{G}_0$, except that we change the procedure of the decryption oracle such that the decryption queries can be answered without the secret keys. Specifically, as shown in Fig. 9, for a decryption query $(\mathsf{id}, (e, c))$ in $\mathbf{G}_1$, instead of decrypting e with sk_{id} to obtain $\hat{r}$ and querying $(\hat{r}, c)$ to $\mathcal{O}_H$, the decryption oracle returns $\perp$ directly if $\mathcal{A}$ did not subimt some tuple $(\hat{r}, c)$ to $\mathcal{O}_H$ such that $e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \mathcal{O}_H(\hat{r}, c))$.

We note that for any decryption query $(\mathsf{id}, (e, c)) \notin \mathcal{C}$, if there exists $((\hat{r}, c), \hat{h}) \in L_H$ such that $e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$, then obviously the decryption oracle in game G_1 and that in G_0 will return the same message as a response. Let evt_0 denote the event that in game G_0, $\mathcal{A}$ submits a decryption query $(\mathsf{id}, (e, c)) \notin \mathcal{C}$ such that (i) "$\sharp\, ((\hat{r}, c), \hat{h}) \in L_H$ s.t. $e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$", and (ii) the decryption oracle does not return $\perp$. Note that $\mathbf{G}_1$ is the same as $\mathbf{G}_0$, except that evt_0 occurs. Thus, we have $|\Pr[\mathcal{D}(\mathbf{G}_1) = 1] - \Pr[\mathcal{D}(\mathbf{G}_0) = 1]| \le \Pr[\mathsf{evt}_0]$. The fact that evt_0 occurs in $\mathbf{G}_0$ implies that for $r' := \mathsf{Dec}(pp, sk_{\mathsf{id}}, e)$, $\mathcal{O}_{\mathrm{H}}(r', c)$ is uniformly and independently sampled from $\mathcal{R}_{\mathsf{Enc}}$. Since IBE is γ-spread, $\Pr[e = \mathsf{Enc}(pp, \mathsf{id}, r'; \mathcal{O}_{\mathrm{H}}(r', c))] \le 2^{-\gamma}$. Hence, we obtain that

$$|\Pr[\mathcal{D}(\mathbf{G}_1) = 1] - \Pr[\mathcal{D}(\mathbf{G}_0) = 1]| \le \Pr[\mathsf{evt}_0] \le q_d \cdot 2^{-\gamma}. \tag{10}$$

Game $\mathbf{G}_2$: Game $\mathbf{G}_2$ is the same as $\mathbf{G}_1$, except that the challenger aborts this game (with output $\perp$) as long as $\mathsf{AbortEARLY}$ occurs. Concretely, as long as $\mathcal{A}_1$ submits a random-oracle query t to $\mathcal{O}_{\mathrm{G}}$ such that $t \in \{r_{i,j} \mid i \in [n],\, j \in [k]\}$, or a a random-oracle query (u_1, u_2) to

Games $\boxed{\mathbf{G}_0}$, $\boxed{\mathbf{G}_1\text{-}\mathbf{G}_3}$, $\boxed{\mathbf{G}_0\text{-}\mathbf{G}_2}$, $\boxed{\mathbf{G}_2\text{-}\mathbf{G}_3}$, $\boxed{\mathbf{G}_3}$

$(PP, MSK) := (pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$; $\mathcal{L}_{\mathrm{id}} := \emptyset$; $\mathcal{L} := \emptyset$; $\mathcal{C} := \emptyset$; $\mathcal{I} := \emptyset$; $(r_{i,j} \leftarrow \mathcal{R}_{\mathsf{FOEnc}})_{i\in[n],j\in[k]}$

$((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}(PP)$; $\mathsf{EvtChal} := \mathrm{true}$; $\mathbf{M}^* := (m^*_{i,j})_{i\in[n],j\in[k]} \leftarrow \mathcal{M}$

$(sk_{\mathsf{id}_i} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id}_i))_{i\in[n]}$; $\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}_i \mid i \in [n]\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}_i, sk_{\mathsf{id}_i}) \mid i \in [n]\}$

For $i = 1$ to n:

For $j = 1$ to k:

$\boxed{c_{i,j} = m^*_{i,j} \oplus \mathcal{O}_{\mathrm{G}}(r_{i,j})}$; $h_{i,j} = \mathcal{O}_{\mathrm{H}}(r_{i,j}, c_{i,j})$ $\boxed{c_{i,j} \leftarrow \{0,1\}^\ell}$; $\boxed{h_{i,j} \leftarrow \mathcal{R}_{\mathsf{Enc}}}$

$e_{i,j} \leftarrow \mathsf{Enc}(pp, \mathsf{id}_i, r_{i,j}; h_{i,j})$; $C^*_{i,j} := (e_{i,j}, c_{i,j})$

$\mathcal{C} := \{(\mathsf{id}_i, C^*_{i,j}) \mid i \in [n], j \in [k]\}$; $(\mathcal{I}', s_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}((C^*_{i,j})_{i\in[n],j\in[k]}, s_1)$; $\mathcal{I} := \mathcal{I}'$

$\boxed{L_G := L_G \cup \{(r_{i,j}, c_{i,j} \oplus m^*_{i,j}) \mid i \in \mathcal{I}, j \in [k]\}}$; $\boxed{L_H := L_H \cup \{((r_{i,j}, c_{i,j}), h_{i,j}) \mid i \in \mathcal{I}, j \in [k]\}}$

$out \leftarrow \mathcal{A}_3^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}((sk_{\mathsf{id}_i}, m^*_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2)$; Return $((\mathsf{id}_i)_{i\in[n]}, \mathbf{M}^*, \mathcal{M}, \mathcal{I}, out)$

$\underline{\mathcal{O}_{\mathrm{G}}(t)}$:

If $(t, \cdot) \notin L_G$:

If $t \in \{r_{i,j} \mid i \in [n], j \in [k]\}$:

If $\neg\mathsf{EvtChal}$: $\mathsf{AbortEARLY} := \mathrm{true}$

Else:

Let (i, j) s.t. $t = r_{i,j}$

$L_G := L_G \cup \{(t, c_{i,j} \oplus m^*_{i,j})\}$

Else:

$g_t \leftarrow \{0,1\}^\ell$; $L_G := L_G \cup \{(t, g_t)\}$

Else: Let g_t s.t. $(t, g_t) \in L_G$

Return g_t

$\underline{\mathcal{O}_{\mathrm{H}}(u_1, u_2)}$:

If $((u_1, u_2), \cdot) \notin L_H$:

If $u_1 \in \{r_{i,j} \mid i \in [n], j \in [k]\}$:

If $\neg\mathsf{EvtChal}$: $\mathsf{AbortEARLY} := \mathrm{true}$

Else:

Let (i, j) s.t. $u_1 = r_{i,j}$

If $u_2 = c_{i,j}$:

$L_H := L_H \cup \{((u_1, u_2), h_{i,j})\}$

Else:

$h_u \leftarrow \mathcal{R}_{\mathsf{Enc}}$; $L_H := L_H \cup \{((u_1, u_2), h_u)\}$

Else: Let h_u s.t. $((u_1, u_2), h_u) \in L_H$

Return h_u

$\underline{\mathcal{O}_{\mathrm{KGen}}(\mathsf{id})}$:

If $(\mathcal{C} \neq \emptyset) \wedge (\mathsf{id} \in \{\mathsf{id}_i \mid i \in [n]\})$: Return $\perp$

If $\mathsf{id} \notin \mathcal{L}_{\mathrm{id}}$: $sk_{\mathsf{id}} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id})$; $\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}, sk_{\mathsf{id}})\}$

Return sk_{id}

$\underline{\boxed{\mathcal{O}_{\mathrm{Dec}}(\mathsf{id}, (e, c))}}$:

If $(\mathsf{id}, (e, c)) \in \mathcal{C}$: Return $\perp$

If $\mathsf{id} \notin \mathcal{L}_{\mathrm{id}}$: $sk_{\mathsf{id}} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id})$

$\hat{r} := \mathsf{Dec}(pp, sk_{\mathsf{id}}, e)$

If $\hat{r} \notin \mathcal{R}_{\mathsf{FOEnc}}$: Return $\perp$

$\hat{h} := \mathcal{O}_{\mathrm{H}}(\hat{r}, c)$

If $e \neq \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$: Return $\perp$

Return $m := c \oplus \mathcal{O}_{\mathrm{G}}(\hat{r})$

$\underline{\boxed{\mathcal{O}_{\mathrm{Dec}}(\mathsf{id}, (e, c))}}$:

If $(\mathsf{id}, (e, c)) \in \mathcal{C}$: Return $\perp$

If $\nexists\ ((\hat{r}, c), \hat{h}) \in L_H$ s.t. $e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$: Return $\perp$

Let $\hat{r}$ s.t. $((\hat{r}, c), \hat{h}) \in L_H \wedge e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$; Return $m := c \oplus \mathcal{O}_{\mathrm{G}}(\hat{r})$

Fig. 9 Games $\mathbf{G}_0$–$\mathbf{G}_3$ in the proof of Theorem 3. Boxed code is only executed in the games specified by the game names in the same box style

$\mathcal{O}_{\mathrm{H}}$ such that $u_1 \in \{r_{i,j} \mid i \in [n], j \in [k]\}$, then AbortEARLY is set true, which means that this game is aborted with output $\perp$. The details are shown in Fig. 9.

Note that $r_{i,j}$ for all $i \in [n]$ and $j \in [k]$ is uniformly random distributed from $\mathcal{A}_1$'s point of view when obtaining PP. We also note that when $\mathcal{A}_1$ queries the random oracles, EvtChal is not set true. Hence,

$$\begin{aligned} &|\Pr[\mathcal{D}(\mathbf{G}_2) = 1] - \Pr[\mathcal{D}(\mathbf{G}_1) = 1]| \\ &\leq \Pr[\mathsf{AbortEARLY}] \leq \sum_{\theta=1}^{q_r} \frac{nk}{|\mathcal{R}_{\mathsf{FOEnc}}| - (\theta - 1)} \leq \frac{nkq_r}{|\mathcal{R}_{\mathsf{FOEnc}}| - q_r}. \end{aligned} \tag{11}$$

Game $\mathbf{G}_3$: Game $\mathbf{G}_3$ is the same as $\mathbf{G}_2$, except that (i) during the generation of the challenge ciphertexts, for all $i \in [n]$ and $j \in [k]$, the procedures "$c_{i,j} = m^*_{i,j} \oplus \mathcal{O}_{\mathrm{G}}(r_{i,j})$, $h_{i,j} = \mathcal{O}_{\mathrm{H}}(r_{i,j}, c_{i,j})$" are replaced with "$c_{i,j} \leftarrow \{0,1\}^{\ell}$, $h_{i,j} \leftarrow \mathcal{R}_{\mathsf{Enc}}$", instead of querying $\mathcal{O}_{\mathrm{G}}$ and $\mathcal{O}_{\mathrm{H}}$, and (ii) $\mathcal{O}_{\mathrm{G}}(r_{i,j}) = c_{i,j} \oplus m^*_{i,j}$ and $\mathcal{O}_{\mathrm{H}}(r_{i,j}, c_{i,j}) = h_{i,j}$ are programmed only when $i \in \mathcal{I}$ or $r_{i,j}$ (resp., $(r_{i,j}, c_{i,j})$) is submitted to $\mathcal{O}_{\mathrm{G}}$ (resp., $\mathcal{O}_{\mathrm{H}}$). The details are shown in Fig. 9.

Note that during the generation of the challenge ciphertexts, both $c_{i,j}$ and $h_{i,j}$ are uniformly distributed for all $i \in [n]$ and $j \in [k]$, since the modification introduced in game $\mathbf{G}_2$. So "$c_{i,j} = m^*_{i,j} \oplus \mathcal{O}_{\mathrm{G}}(r_{i,j})$, $h_{i,j} = \mathcal{O}_{\mathrm{H}}(r_{i,j}, c_{i,j})$" can be replaced with "$c_{i,j} \leftarrow \{0,1\}^{\ell}$, $h_{i,j} \leftarrow \mathcal{R}_{\mathsf{Enc}}$". The additional procedures for answering $\mathcal{A}$'s random-oracle queries and opening queries are introduced to ensure that for all $i \in [n]$ and $j \in [k]$, $\mathcal{O}_{\mathrm{G}}(r_{i,j})$ and $\mathcal{O}_{\mathrm{H}}(r_{i,j}, c_{i,j})$ are programmed consistently. Hence, from $\mathcal{A}$'s point of view, these two games are identical, i.e.,

$$\Pr[\mathcal{D}(\mathbf{G}_3) = 1] - \Pr[\mathcal{D}(\mathbf{G}_2) = 1] = 0. \tag{12}$$

For convenience, we rewrite game $\mathbf{G}_3$ in Fig. 10, removing the replaced procedures.

Game $\mathbf{G}_4$: Game $\mathbf{G}_4$ is the same as $\mathbf{G}_3$, except that the challenger does not generate the secret keys corresponding to the challenge identities (i.e., $(\mathsf{id}_i)_{i \in [n]}$), until the identities are submitted to $\mathcal{O}_{\mathrm{KGen}}$ or submitted for the selective opening query. The details are shown in Fig. 10.

Note that (i) $\mathcal{O}_{\mathrm{Dec}}$ can answer the decryption queries without any secret key because of the modification introduced in $\mathbf{G}_1$, and (ii) in $\mathbf{G}_3$, for any $\mathsf{id} \in \mathcal{ID}$, sk_{id} has never been used or given to $\mathcal{A}$ until the identity id is submitted to $\mathcal{O}_{\mathrm{KGen}}$ or included in $\mathcal{I}$ as the selective opening query. Therefore, from $\mathcal{A}$'s point of view, $\mathbf{G}_4$ and $\mathbf{G}_3$ are identical, i.e.,

$$\Pr[\mathcal{D}(\mathbf{G}_4) = 1] - \Pr[\mathcal{D}(\mathbf{G}_3) = 1] = 0. \tag{13}$$

Game $\mathbf{G}_5$: In this game, a new abort condition is added (as shown in Fig. 10). Specifically, if $\mathcal{A}_2$ or $\mathcal{A}_3$ submits a random-oracle query $r_{i,j}$ (resp., $(r_{i,j}, c_{i,j})$) to $\mathcal{O}_{\mathrm{G}}$ (resp., $\mathcal{O}_{\mathrm{H}}$) satisfying that $(r_{i,j}, \cdot) \notin L_G \wedge i \notin \mathcal{I}$ (resp., $((r_{i,j}, c_{i,j}), \cdot) \notin L_H \wedge i \notin \mathcal{I}$), then AbortHASH is set true, which means that this game is aborted with output $\perp$. We present the following lemma with a postponed proof. □

Lemma 4 $|\Pr[\mathcal{D}(\mathbf{G}_5) = 1] - \Pr[\mathcal{D}(\mathbf{G}_4) = 1]| \leq nkq_r \cdot \mathbf{Adv}^{\text{ow-id-cpa}}_{\mathsf{IBE},\widetilde{\mathcal{A}}}(\lambda)$ *for some PPT adversary* $\widetilde{\mathcal{A}}$.

Now, we construct a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$, which simulates $\mathbf{G}_5$ perfectly for $\mathcal{A}$, as shown in Fig. 11. Obviously,

$$\mathbf{G}^{\mathrm{rso}_k\text{-cca-ideal}}_{\mathsf{FOIBE},\mathcal{S},n}(\lambda) = \mathbf{G}_5. \tag{14}$$

Games $\boxed{\mathbf{G}_3}$, $\boxed{\mathbf{G}_3\text{-}\mathbf{G}_4}$, $\boxed{\mathbf{G}_4\text{-}\mathbf{G}_5}$, $\boxed{\mathbf{G}_5}$

$(PP, MSK) := (pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$; $\mathcal{L}_{\mathrm{id}} := \emptyset$; $\mathcal{L} := \emptyset$; $\mathcal{C} := \emptyset$; $\mathcal{I} := \emptyset$; $(r_{i,j} \leftarrow \mathcal{R}_{\mathsf{FOEnc}})_{i\in[n],j\in[k]}$

$((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}(PP)$; $\mathsf{EvtChal} := \mathrm{true}$; $\mathbf{M}^* := (m^*_{i,j})_{i\in[n],j\in[k]} \leftarrow \mathcal{M}$

$\boxed{(sk_{\mathsf{id}_i} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id}_i))_{i\in[n]};\ \mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}_i \mid i \in [n]\};\ \mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}_i, sk_{\mathsf{id}_i}) \mid i \in [n]\}}$

For $i = 1$ to n:

  For $j = 1$ to k:

    $c_{i,j} \leftarrow \{0,1\}^\ell$; $h_{i,j} \leftarrow \mathcal{R}_{\mathsf{Enc}}$; $e_{i,j} \leftarrow \mathsf{Enc}(pp, \mathsf{id}_i, r_{i,j}; h_{i,j})$; $C^*_{i,j} := (e_{i,j}, c_{i,j})$

$\mathcal{C} := \{(\mathsf{id}_i, C^*_{i,j}) \mid i \in [n], j \in [k]\}$; $(\mathcal{I}', s_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}((C^*_{i,j})_{i\in[n],j\in[k]}, s_1)$; $\mathcal{I} := \mathcal{I}'$

$L_G := L_G \cup \{(r_{i,j}, c_{i,j} \oplus m^*_{i,j}) \mid i \in \mathcal{I}, j \in [k]\}$; $L_H := L_H \cup \{((r_{i,j}, c_{i,j}), h_{i,j}) \mid i \in \mathcal{I}, j \in [k]\}$

For $i \in \mathcal{I}$:

  If $\mathsf{id}_i \notin \mathcal{L}_{\mathrm{id}}$: $sk_{\mathsf{id}_i} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id}_i)$; $\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}_i\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}_i, sk_{\mathsf{id}_i})\}$

$out \leftarrow \mathcal{A}_3^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}((sk_{\mathsf{id}_i}, m^*_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2)$; Return $((\mathsf{id}_i)_{i\in[n]}, \mathbf{M}^*, \mathcal{M}, \mathcal{I}, out)$

$\underline{\mathcal{O}_{\mathrm{G}}(t):}$

If $(t, \cdot) \notin L_G$:

  If $t \in \{r_{i,j} \mid i \in [n], j \in [k]\}$:

    If $\neg\mathsf{EvtChal}$: $\mathsf{AbortEARLY} := \mathrm{true}$

    Else:

      Let (i, j) s.t. $t = r_{i,j}$

      $\boxed{\text{If } i \notin \mathcal{I}\text{: } \mathsf{AbortHASH} := \mathrm{true}}$

      $\boxed{L_G := L_G \cup \{(t, c_{i,j} \oplus m^*_{i,j})\}}$

  Else: $g_t \leftarrow \{0,1\}^\ell$; $L_G := L_G \cup \{(t, g_t)\}$

Else: Let g_t s.t. $(t, g_t) \in L_G$

Return g_t

$\underline{\mathcal{O}_{\mathrm{H}}(u_1, u_2):}$

If $((u_1, u_2), \cdot) \notin L_H$:

  If $u_1 \in \{r_{i,j} \mid i \in [n], j \in [k]\}$:

    If $\neg\mathsf{EvtChal}$: $\mathsf{AbortEARLY} := \mathrm{true}$

    Else:

      Let (i, j) s.t. $u_1 = r_{i,j}$

      If $u_2 = c_{i,j}$:

        $\boxed{\text{If } i \notin \mathcal{I}\text{: } \mathsf{AbortHASH} := \mathrm{true}}$

        $\boxed{L_H := L_H \cup \{((u_1, u_2), h_{i,j})\}}$

  Else: $h_u \leftarrow \mathcal{R}_{\mathsf{Enc}}$; $L_H := L_H \cup \{((u_1, u_2), h_u)\}$

Else: Let h_u s.t. $((u_1, u_2), h_u) \in L_H$

Return h_u

$\underline{\mathcal{O}_{\mathrm{KGen}}(\mathsf{id}):}$

If $(\mathcal{C} \neq \emptyset) \wedge (\mathsf{id} \in \{\mathsf{id}_i \mid i \in [n]\})$: Return $\perp$

If $\mathsf{id} \notin \mathcal{L}_{\mathrm{id}}$: $sk_{\mathsf{id}} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id})$; $\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}, sk_{\mathsf{id}})\}$

Return sk_{id}

$\underline{\mathcal{O}_{\mathrm{Dec}}(\mathsf{id}, (e, c)):}$

If $(\mathsf{id}, (e, c)) \in \mathcal{C}$: Return $\perp$

If $\nexists\ ((\hat{r}, c), \hat{h}) \in L_H$ s.t. $e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$: Return $\perp$

Let $\hat{r}$ s.t. $((\hat{r}, c), \hat{h}) \in L_H \wedge e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$; Return $m := c \oplus \mathcal{O}_{\mathrm{G}}(\hat{r})$

Fig. 10 Games $\mathbf{G}_3$–$\mathbf{G}_5$ in the proof of Theorem 3. Boxed code is only executed in the games specified by the game names in the same box style

Hence, combining Eqs. (9)–(14) and Lemma 4, we obtain that

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{rso}_k\text{-cca}}_{\mathsf{FOIBE},\mathcal{A},\mathcal{S},\mathcal{D},n}(\lambda) \\
&= |\Pr[\mathcal{D}(\mathbf{G}^{\mathrm{rso}_k\text{-cca-real}}_{\mathsf{FOIBE},\mathcal{A},n}(\lambda)) = 1] - \Pr[\mathcal{D}(\mathbf{G}^{\mathrm{rso}_k\text{-cca-ideal}}_{\mathsf{FOIBE},\mathcal{S},n}(\lambda)) = 1]| \\
&= |\Pr[\mathcal{D}(\mathbf{G}_0) = 1] - \Pr[\mathcal{D}(\mathbf{G}_5) = 1]| \\
&\leq q_d \cdot 2^{-\gamma} + \frac{nkq_r}{|\mathcal{R}_{\mathsf{FOEnc}}| - q_r} + nkq_r \cdot \mathbf{Adv}^{\mathrm{ow\text{-}id\text{-}cpa}}_{\mathsf{IBE},\widetilde{\mathcal{A}}}(\lambda),
\end{aligned}$$

for some adversary $\widetilde{\mathcal{A}}$.

$\underline{\mathcal{S}_1(1^\lambda):}$
$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$; $PP := pp$; $MSK := msk$; $\mathcal{L}_{\mathrm{id}} := \emptyset$; $\mathcal{L} := \emptyset$; $\mathcal{C} := \emptyset$; $\mathcal{I} := \emptyset$
$(r_{i,j} \leftarrow \mathcal{R}_{\mathsf{FOEnc}})_{i\in[n],j\in[k]}$; $((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}(PP)$
$\mathsf{EvtChal} := \mathrm{true}$; $(sk_{\mathsf{id}_i} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id}_i))_{i\in[n]}$
$\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}_i \mid i \in [n]\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}_i, sk_{\mathsf{id}_i}) \mid i \in [n]\}$
$s_1' := (PP, (\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, (sk_{\mathsf{id}_i})_{i\in[n]}, (r_{i,j})_{i\in[n],j\in[k]}, \mathcal{I}, \mathcal{L}_{\mathrm{id}}, \mathcal{L}, \mathcal{C}, s_1)$
Return $((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1')$

$\underline{\mathcal{S}_2(s_1'):}$
For $i = 1$ to n:
  For $j = 1$ to k:
    $c_{i,j} \leftarrow \{0,1\}^\ell$; $h_{i,j} \leftarrow \mathcal{R}_{\mathsf{Enc}}$; $e_{i,j} \leftarrow \mathsf{Enc}(pp, \mathsf{id}_i, r_{i,j}; h_{i,j})$; $C^*_{i,j} := (e_{i,j}, c_{i,j})$
$\mathcal{C} := \{(\mathsf{id}_i, C^*_{i,j}) \mid i \in [n], j \in [k]\}$; $(\mathcal{I}', s_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}((C^*_{i,j})_{i\in[n],j\in[k]}, s_1)$; $\mathcal{I} := \mathcal{I}'$
$s_2' := (PP, (\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, (sk_{\mathsf{id}_i})_{i\in[n]}, (r_{i,j})_{i\in[n],j\in[k]}, (C^*_{i,j})_{i\in[n],j\in[k]}, \mathcal{I}, \mathcal{L}_{\mathrm{id}}, \mathcal{L}, \mathcal{C}, s_2)$
Return $(\mathcal{I}, s_2')$

$\underline{\mathcal{S}_3((m^*_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2'):}$
$L_G := L_G \cup \{(r_{i,j}, c_{i,j} \oplus m^*_{i,j}) \mid i \in \mathcal{I}, j \in [k]\}$; $L_H := L_H \cup \{((r_{i,j}, c_{i,j}), h_{i,j}) \mid i \in \mathcal{I}, j \in [k]\}$
$out \leftarrow \mathcal{A}_3^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}((sk_{\mathsf{id}_i}, m^*_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2)$
Return out

$\underline{\mathcal{O}_{\mathrm{G}}(t):}$
If $(t, \cdot) \notin L_G$:
  If $t \in \{r_{i,j} \mid i \in [n], j \in [k]\}$:
    If $\neg\mathsf{EvtChal}$: $\mathsf{AbortEARLY} := \mathrm{true}$
    Else:
      Let (i, j) s.t. $t = r_{i,j}$
      If $i \notin \mathcal{I}$: $\mathsf{AbortHASH} := \mathrm{true}$
  Else: $g_t \leftarrow \{0,1\}^\ell$; $L_G := L_G \cup \{(t, g_t)\}$
Else: Let g_t s.t. $(t, g_t) \in L_G$
Return g_t

$\underline{\mathcal{O}_{\mathrm{H}}(u_1, u_2):}$
If $((u_1, u_2), \cdot) \notin L_H$:
  If $u_1 \in \{r_{i,j} \mid i \in [n], j \in [k]\}$:
    If $\neg\mathsf{EvtChal}$: $\mathsf{AbortEARLY} := \mathrm{true}$
    Else:
      Let (i, j) s.t. $u_1 = r_{i,j}$
      If $u_2 = c_{i,j}$:
        If $i \notin \mathcal{I}$: $\mathsf{AbortHASH} := \mathrm{true}$
  Else: $h_u \leftarrow \mathcal{R}_{\mathsf{Enc}}$; $L_H := L_H \cup \{((u_1, u_2), h_u)\}$
Else: Let h_u s.t. $((u_1, u_2), h_u) \in L_H$
Return h_u

$\underline{\mathcal{O}_{\mathrm{KGen}}(\mathsf{id}):}$
If $(\mathcal{C} \neq \emptyset) \wedge (\mathsf{id} \in \{\mathsf{id}_i \mid i \in [n]\})$: Return $\perp$
If $\mathsf{id} \notin \mathcal{L}_{\mathrm{id}}$: $sk_{\mathsf{id}} \leftarrow \mathsf{KGen}(pp, msk, \mathsf{id})$; $\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}, sk_{\mathsf{id}})\}$
Return sk_{id}

$\underline{\mathcal{O}_{\mathrm{Dec}}(\mathsf{id}, (e, c)):}$
If $(\mathsf{id}, (e, c)) \in \mathcal{C}$: Return $\perp$
If $\nexists\ ((\hat{r}, c), \hat{h}) \in L_H$ s.t. $e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$: Return $\perp$
Let $\hat{r}$ s.t. $((\hat{r}, c), \hat{h}) \in L_H \wedge e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$; Return $m := c \oplus \mathcal{O}_{\mathrm{G}}(\hat{r})$

Fig. 11 Simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ in the proof of Theorem 3

We catch up with the proof of Lemma 4.

Proof of Lemma 4 We note that $\mathbf{G}_5$ and $\mathbf{G}_4$ proceed identically until event AbortHASH is set true. Thus, we have $|\Pr[\mathcal{D}(\mathbf{G}_5) = 1] - \Pr[\mathcal{D}(\mathbf{G}_4) = 1]| \leq \Pr[\mathsf{AbortHASH}]$.

So what remains is to compute $\Pr[\mathsf{AbortHASH}]$.

Without loss of generality, we assume that for any tuple (r, c), before querying oracle $\mathcal{O}_{\mathrm{H}}$ on (r, c), $\mathcal{A}$ will query $\mathcal{O}_{\mathrm{G}}$ on r firstly. With this assumption, to answer $\mathcal{A}$'s decryption queries, the challenger does not need to "access to" $\mathcal{O}_{\mathrm{G}}$, as shown in Fig. 12.

$\widetilde{\mathcal{A}}_1^{\widetilde{O}_{\mathrm{KGen}}}(pp)$:
$\widetilde{m} := \bot$; $\widetilde{i} \leftarrow [n]$; $\widetilde{j} \leftarrow [k]$; $\widetilde{\theta} \leftarrow [q_r]$; $PP := pp$; $\mathcal{L}_{\mathrm{id}} := \emptyset$; $\mathcal{L} := \emptyset$; $\mathcal{C} := \emptyset$; $\mathcal{I} := \emptyset$; Count $:= 0$
$(r_{i,j} \leftarrow \mathcal{R}_{\mathsf{FOEnc}})_{(i,j)\in([n]\times[k])\setminus\{(\widetilde{i},\widetilde{j})\}}$; $((\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, s_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}(PP)$
$\mathsf{id}^* := \mathsf{id}_{\widetilde{i}}$; $\widetilde{s}_1 := (PP, (\mathsf{id}_i)_{i\in[n]}, \mathcal{M}, \widetilde{i}, \widetilde{j}, (r_{i,j})_{(i,j)\in([n]\times[k])\setminus\{(\widetilde{i},\widetilde{j})\}}, \mathcal{L}_{\mathrm{id}}, \mathcal{L}, \mathcal{C}, s_1)$; Return $(\mathsf{id}^*, \widetilde{s}_1)$

$\widetilde{\mathcal{A}}_2^{\widetilde{O}_{\mathrm{KGen}}}(\widetilde{c}^*, \widetilde{s}_1)$:
EvtChal $:=$ true; $\mathbf{M}^* := (m^*_{i,j})_{i\in[n],j\in[k]} \leftarrow \mathcal{M}$
For $(i,j) \in ([n]\times[k]) \setminus \{(\widetilde{i},\widetilde{j})\}$:
$c_{i,j} \leftarrow \{0,1\}^\ell$; $h_{i,j} \leftarrow \mathcal{R}_{\mathsf{Enc}}$; $e_{i,j} \leftarrow \mathsf{Enc}(pp, \mathsf{id}_i, r_{i,j}; h_{i,j})$; $C^*_{i,j} := (e_{i,j}, c_{i,j})$
$c_{\widetilde{i},\widetilde{j}} \leftarrow \{0,1\}^\ell$; $e_{\widetilde{i},\widetilde{j}} := \widetilde{c}^*$; $C^*_{\widetilde{i},\widetilde{j}} := (e_{\widetilde{i},\widetilde{j}}, c_{\widetilde{i},\widetilde{j}})$
$\mathcal{C} := \{(\mathsf{id}_i, C^*_{i,j}) \mid i \in [n], j \in [k]\}$; $(\mathcal{I}', s_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}((C^*_{i,j})_{i\in[n],j\in[k]}, s_1)$; $\mathcal{I} := \mathcal{I}'$
If $\widetilde{i} \in \mathcal{I}$: Abort-Return-I $:=$ true
$L_G := L_G \cup \{(r_{i,j}, c_{i,j} \oplus m^*_{i,j}) \mid i \in \mathcal{I}, j \in [k]\}$; $L_H := L_H \cup \{((r_{i,j}, c_{i,j}), h_{i,j}) \mid i \in \mathcal{I}, j \in [k]\}$
For $i \in \mathcal{I}$:
If $\mathsf{id}_i \notin \mathcal{L}_{\mathrm{id}}$: $sk_{\mathsf{id}_i} \leftarrow \widetilde{O}_{\mathrm{KGen}}(\mathsf{id}_i)$; $\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}_i\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}_i, sk_{\mathsf{id}_i})\}$
$out \leftarrow \mathcal{A}_3^{\mathcal{O}_{\mathrm{KGen}},\mathcal{O}_{\mathrm{Dec}},\mathcal{O}_{\mathrm{G}},\mathcal{O}_{\mathrm{H}}}((sk_{\mathsf{id}_i}, m^*_{i,j})_{i\in\mathcal{I},j\in[k]}, s_2)$; Return $\widetilde{m}$

$\mathcal{O}_{\mathrm{G}}(t)$:
Count $:=$ Count $+ 1$
If Count $= \widetilde{\theta}$: $\widetilde{m} := t$; Abort-Return-II $:=$ true
If $(t,\cdot) \notin L_G$:
If $t \in \{r_{i,j} \mid (i,j) \in ([n]\times[k]) \setminus \{(\widetilde{i},\widetilde{j})\}\}$:
If $\neg$EvtChal: AbortEARLY $:=$ true
Else:
Let (i,j) s.t. $t = r_{i,j}$
If $i \notin \mathcal{I}$: AbortHASH $:=$ true
Else: $g_t \leftarrow \{0,1\}^\ell$; $L_G := L_G \cup \{(t, g_t)\}$
Else: Let g_t s.t. $(t, g_t) \in L_G$
Return g_t

$\mathcal{O}_{\mathrm{H}}(u_1, u_2)$:
Count $:=$ Count $+ 1$
If Count $= \widetilde{\theta}$: $\widetilde{m} := u_1$; Abort-Return-II $:=$ true
If $((u_1,u_2),\cdot) \notin L_H$:
If $u_1 \in \{r_{i,j} \mid i \in [n], j \in [k]\}$:
If $\neg$EvtChal: AbortEARLY $:=$ true
Else:
Let (i,j) s.t. $u_1 = r_{i,j}$
If $u_2 = c_{i,j}$:
If $i \notin \mathcal{I}$: AbortHASH $:=$ true
Else: $h_u \leftarrow \mathcal{R}_{\mathsf{Enc}}$; $L_H := L_H \cup \{((u_1,u_2), h_u)\}$
Else: Let h_u s.t. $((u_1,u_2), h_u) \in L_H$
Return h_u

$\mathcal{O}_{\mathrm{KGen}}(\mathsf{id})$:
If $(\mathcal{C} \neq \emptyset) \wedge (\mathsf{id} \in \{\mathsf{id}_i \mid i \in [n]\})$: Return $\bot$
If $\mathsf{id} \notin \mathcal{L}_{\mathrm{id}}$: $sk_{\mathsf{id}} \leftarrow \widetilde{O}_{\mathrm{KGen}}(pp, msk, \mathsf{id})$; $\mathcal{L}_{\mathrm{id}} := \mathcal{L}_{\mathrm{id}} \cup \{\mathsf{id}\}$; $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{id}, sk_{\mathsf{id}})\}$
Return sk_{id}

$\mathcal{O}_{\mathrm{Dec}}(\mathsf{id}, (e,c))$:
If $(\mathsf{id}, (e,c)) \in \mathcal{C}$: Return $\bot$
If $\nexists\ ((\hat{r},c), \hat{h}) \in L_H$ s.t. $e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$: Return $\bot$
Let $\hat{r}$ s.t. $((\hat{r},c), \hat{h}) \in L_H \wedge e = \mathsf{Enc}(pp, \mathsf{id}, \hat{r}; \hat{h})$; Let g_t s.t. $(\hat{r}, g_t) \in L_G$; Return $m := c \oplus g_t$

Fig. 12 Adversary $\widetilde{\mathcal{A}} = (\widetilde{\mathcal{A}}_1, \widetilde{\mathcal{A}}_2)$ attacking IBE in the proof of Lemma 4. Note that without loss of generality, we assume that for any tuple (id, r, c), before querying oracle $\mathcal{O}_{\mathrm{H}}$ on (r, c), $\mathcal{A}$ will query $\mathcal{O}_{\mathrm{G}}$ on r firstly

Now, we construct an OW-ID-CPA adversary $\widetilde{\mathcal{A}}$, attacking IBE, from $\mathcal{A}$ in Fig. 12. We introduce a special event Abort-Return (in Fig. 12), and require that when Abort-Return is set true, $\widetilde{\mathcal{A}}$ immediately terminate the simulation and returns the current $\widetilde{m}$ as its final output.

For any $i' \in [n]$, let $\mathsf{AbortHASH}_{i'}$ denote the event that $\mathsf{AbortHASH}$ occurs for the first time for $i = i'$. Thus,

$$\Pr[\mathsf{AbortHASH}] \leq \sum_{i=1}^{n} \Pr[\mathsf{AbortHASH}_i]. \tag{15}$$

For any $j' \in [k]$, let $\mathsf{AbortHASH}_{i,j'}$ denote the event that $\mathsf{AbortHASH}_i$ occurs for the first time at $\mathcal{A}$'s random-oracle query $r_{i,j'}$ to $\mathcal{O}_{\mathrm{G}}$ or random-oracle query $(r_{i,j'}, c_{i,j'})$ to $\mathcal{O}_{\mathrm{H}}$. Thus,

$$\Pr[\mathsf{AbortHASH}_i] \leq \sum_{j=1}^{k} \Pr[\mathsf{AbortHASH}_{i,j}]. \tag{16}$$

Furthermore, for any $\theta' \in [q_r]$, let $\mathsf{AbortHASH}_{i,j}^{(\theta')}$ denote the event that $\mathsf{AbortHASH}_{i,j}$ occurs for the first time at $\mathcal{A}$'s θ'-th random oracle query. Thus,

$$\Pr[\mathsf{AbortHASH}_{i,j}] \leq \sum_{\theta=1}^{q_r} \Pr[\mathsf{AbortHASH}_{i,j}^{(\theta)}]. \tag{17}$$

For $\widetilde{i} \leftarrow [n]$, $\widetilde{j} \leftarrow [k]$ and $\widetilde{\theta} \leftarrow [q_r]$, we claim that the probability that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ occurs in the game simulated by $\widetilde{\mathcal{A}}$ is equal to $\Pr[\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}]$ (i.e., the probability that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ occurs in $\mathbf{G}_4$). The reason is as follows.

(1) When ignoring the oracles $\mathcal{O}_{\mathrm{G}}$, $\mathcal{O}_{\mathrm{H}}$, ,$\mathcal{O}_{\mathrm{KGen}}$ and $\mathcal{O}_{\mathrm{Dec}}$, the game simulated by $\widetilde{\mathcal{A}}$ is the same as $\mathbf{G}_4$, except for the case that $\mathsf{Abort\text{-}Return\text{-}I}$ occurs. Note that in the game simulated by $\widetilde{\mathcal{A}}$, $\widetilde{\mathcal{A}}$ will terminate the simulation with output $\widetilde{m}$ as long as $\mathsf{Abort\text{-}Return\text{-}I}$ occurs. On the other hand, $\mathsf{Abort\text{-}Return\text{-}I}$ occurs if and only if $\widetilde{i} \in \mathcal{I}$, which suggests that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ will not occur. So $\widetilde{\mathcal{A}}$ can terminate the simulation when $\mathsf{Abort\text{-}Return\text{-}I}$ occurs without influencing the probability that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ occurs.

(2) Both oracles $\mathcal{O}_{\mathrm{KGen}}$ and $\mathcal{O}_{\mathrm{Dec}}$ in the game simulated by $\widetilde{\mathcal{A}}$ are identical with that in $\mathbf{G}_4$.

(3) The only differences between the oracle $\mathcal{O}_{\mathrm{G}}$ simulated by $\widetilde{\mathcal{A}}$ (denoted as $\mathcal{O}_{\mathrm{G}}^{\widetilde{\mathcal{A}}}$) and the real $\mathcal{O}_{\mathrm{G}}$ in $\mathbf{G}_4$ are: (i) $\widetilde{\mathcal{A}}$ introduces $\mathsf{Abort\text{-}Return\text{-}II}$ and aborts when $\mathsf{Count} = \widetilde{\theta}$, and (ii) for a query t satisfying $(t, \cdot) \notin L_G$, $\widetilde{\mathcal{A}}$ checks whether $t \in \{r_{i,j} \mid (i,j) \in ([n] \times [k]) \setminus \{(\widetilde{i}, \widetilde{j})\}\}$ instead of checking whether $t \in \{r_{i,j} \mid i \in [n], j \in [k]\}$ in $\mathbf{G}_4$.

   (a) For (i), $\mathsf{Abort\text{-}Return\text{-}II}$ is set true when $\mathsf{Count} = \widetilde{\theta}$, suggesting that $\widetilde{\mathcal{A}}$ terminates the simulation with output $\widetilde{m}$ immediately. Note that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ focuses on $\mathcal{A}$'s $\widetilde{\theta}$-th random oracle query. Whatever happens when $\mathsf{Count} > \widetilde{\theta}$ will not affect $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$. So introducing $\mathsf{Abort\text{-}Return\text{-}II}$ will not influence the probability that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ occurs.

   (b) Now we analyze the case of (ii). For the θ-th query t satisfying $(t, \cdot) \notin L_G$ and $\theta < \widetilde{\theta}$,[2] if $t \notin \{r_{i,j} \mid i \in [n], j \in [k]\}$ or $t \in \{r_{i,j} \mid (i,j) \in ([n] \times [k]) \setminus \{(\widetilde{i}, \widetilde{j})\}\}$, obviously $\mathcal{O}_{\mathrm{G}}^{\widetilde{\mathcal{A}}}$ and $\mathcal{O}_{\mathrm{G}}$ both generate the response in the same way. On the other hand, if $t = r_{\widetilde{i},\widetilde{j}}$, then $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ will not occur, because the θ-th query $t = r_{\widetilde{i},\widetilde{j}}$

[2] Note that the case of $\theta \geq \widetilde{\theta}$ has been discussed in (a).

leads to $\mathsf{AbortHASH}$, where $\theta < \widetilde{\theta}$, and $\mathcal{A}$ are assumed to not repeat identical queries to the same oracles. So no matter what $\widetilde{\mathcal{A}}$ returns as the response of $\mathcal{O}_{\mathrm{G}}^{\widetilde{\mathcal{A}}}$, the response will not affect $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$.

Hence, the differences between $\mathcal{O}_{\mathrm{G}}^{\widetilde{\mathcal{A}}}$ (in the game simulated by $\widetilde{\mathcal{A}}$) and $\mathcal{O}_{\mathrm{G}}$ (in $\mathbf{G}_4$) does not influence the probability that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ occurs.

(4) The only differences between the oracle $\mathcal{O}_{\mathrm{H}}$ simulated by $\widetilde{\mathcal{A}}$ (denoted as $\mathcal{O}_{\mathrm{H}}^{\widetilde{\mathcal{A}}}$) and the real $\mathcal{O}_{\mathrm{H}}$ in $\mathbf{G}_4$ are: (i) $\widetilde{\mathcal{A}}$ introduces $\mathsf{Abort\text{-}Return\text{-}II}$ and aborts when $\mathsf{Count} = \widetilde{\theta}$, and (ii) for a query (u_1, u_2) satisfying $((u_1, u_2), \cdot) \notin L_H$, $\widetilde{\mathcal{A}}$ checks whether $u_1 \in \{r_{i,j} \mid (i, j) \in ([n] \times [k]) \setminus \{(\widetilde{i}, \widetilde{j})\}\}$ instead of checking whether $u_1 \in \{r_{i,j} \mid i \in [n], j \in [k]\}$ in $\mathbf{G}_4$.

(a) For (i), $\mathsf{Abort\text{-}Return\text{-}II}$ is set true when $\mathsf{Count} = \widetilde{\theta}$, suggesting that $\widetilde{\mathcal{A}}$ terminates the simulation with output $\widetilde{m}$ immediately. Note that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ focuses on $\mathcal{A}$'s $\widetilde{\theta}$-th random oracle query. Whatever happens when $\mathsf{Count} > \widetilde{\theta}$ will not affect $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$. So introducing $\mathsf{Abort\text{-}Return\text{-}II}$ will not influence the probability that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ occurs.

(b) Now we analyze the case of (ii). For the θ-th query (u_1, u_2) satisfying $((u_1, u_2), \cdot) \notin L_G$ and $\theta < \widetilde{\theta}$, if $u_1 \notin \{r_{i,j} \mid i \in [n], j \in [k]\}$ or $u_1 \in \{r_{i,j} \mid (i, j) \in ([n] \times [k]) \setminus \{(\widetilde{i}, \widetilde{j})\}\}$, obviously $\mathcal{O}_{\mathrm{H}}^{\widetilde{\mathcal{A}}}$ and $\mathcal{O}_{\mathrm{H}}$ both generate the response in the same way. On the other hand, if $u_1 = r_{\widetilde{i},\widetilde{j}}$, then there are two cases:

- Case 1: $u_2 \neq c_{\widetilde{i},\widetilde{j}}$. In this case, both $\mathcal{O}_{\mathrm{H}}^{\widetilde{\mathcal{A}}}$ and $\mathcal{O}_{\mathrm{H}}$ will generate the response in the same way: sampling $h_u \leftarrow \mathcal{R}_{\mathsf{Enc}}$ and adding $((u_1, u_2), h_u)$ to L_H.
- Case 2: $u_2 = c_{\widetilde{i},\widetilde{j}}$. In this case, $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ will not occur since the θ-th query $(u_1, u_2) = (r_{\widetilde{i},\widetilde{j}}, c_{\widetilde{i},\widetilde{j}})$ leads to $\mathsf{AbortHASH}$, where $\theta < \widetilde{\theta}$, and $\mathcal{A}$ are assumed to not repeat identical queries to the same oracles. So no matter what $\widetilde{\mathcal{A}}$ returns as the response of $\mathcal{O}_{\mathrm{H}}^{\widetilde{\mathcal{A}}}$, the response will not affect $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$.

Hence, the differences between $\mathcal{O}_{\mathrm{H}}^{\widetilde{\mathcal{A}}}$ (in the game simulated by $\widetilde{\mathcal{A}}$) and $\mathcal{O}_{\mathrm{H}}$ (in $\mathbf{G}_4$) does not influence the probability that $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ occurs.

Note that $\widetilde{\mathcal{A}}$ succeeds if and only if $\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}$ occurs. Hence,

$$\begin{aligned}
\mathbf{Adv}_{\mathsf{IBE},\widetilde{\mathcal{A}}}^{\text{ow-id-cpa}}(\lambda) &= \Pr[\mathsf{AbortHASH}_{\widetilde{i},\widetilde{j}}^{(\widetilde{\theta})}] \\
&= \frac{1}{nkq_r} \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{\theta=1}^{q_r} \Pr[\mathsf{AbortHASH}_{i,j}^{(\theta)}] \\
&\geq \frac{1}{nkq_r} \Pr[\mathsf{AbortHASH}].
\end{aligned}$$

□

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant Nos. 61922036, U2001205, 62106114), and Major Program of Guangdong Basic and Applied Research (Grant No. 2019B030302008).

Data availibility Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Agrawal S., Boneh D., Boyen X.: Efficient lattice (H)IBE in the standard model. In: EUROCRYPT 2010, pp. 553–572. Springer (2010).
2. Bellare M., Dowsley R., Waters B., Yilek S.: Standard security does not imply security against selective-opening. In: EUROCRYPT 2012, pp. 645–662. Springer (2012).
3. Bellare M., Hofheinz D., Yilek S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: EUROCRYPT 2009, pp. 1–35. Springer (2009).
4. Bellare M., O'Neill A.: Semantically-secure functional encryption: possibility results, impossibility results and the quest for a general definition. In: CANS 2013, pp. 218–234. Springer (2013).
5. Bellare M., Waters B., Yilek S.: Identity-based encryption secure against selective opening attack. In: TCC 2011, pp. 235–252. Springer (2011).
6. Bellare M., Yilek S.: Encryption schemes secure under selective opening attack. Cryptology ePrint Archive, Report 2009/101 (2009). https://eprint.iacr.org/2009/101.
7. Boneh D., Franklin M.: Identity-based encryption from the weil pairing. In: CRYPTO 2001, pp. 213–229. Springer (2001).
8. Boyen X., Li Q.: All-but-many lossy trapdoor functions from lattices and applications. In: CRYPTO 2017, pp. 298–331. Springer (2017).
9. Fehr S., Hofheinz D., Kiltz E., Wee H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: EUROCRYPT 2010, pp. 381–402. Springer (2010).
10. Fujisaki E., Okamoto T.: Secure integration of asymmetric and symmetric encryption schemes. In: CRYPTO 1999, pp. 537–554. Springer (1999).
11. Fujisaki E., Okamoto T.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptol. **26**(1), 80–101 (2013).
12. Groth J., Ostrovsky R., Sahai A.: Perfect non-interactive zero knowledge for np. In: EUROCRYPT 2006, pp. 339–358. Springer (2006).
13. Hara K., Kitagawa F., Matsuda T., Hanaoka G., Tanaka K.: Simulation-based receiver selective opening CCA secure PKE from standard computational assumptions. In: Security and Cryptography for Networks 2018, pp. 140–159. Springer (2018).
14. Hara K., Matsuda T., Tanaka K.: Receiver selective opening chosen ciphertext secure identity-based encryption. In: Proceedings of the 8th ACM on ASIA Public-Key Cryptography Workshop, pp. 51–59 (2021).
15. Hazay C., Patra A., Warinschi B.: Selective opening security for receivers. In: ASIACRYPT 2015, pp. 443–469. Springer (2015).
16. Hemenway B., Libert B., Ostrovsky R., Vergnaud D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: ASIACRYPT 2011, pp. 70–88. Springer (2011).
17. Heuer F., Jager T., Kiltz E., Schäge S.: On the selective opening security of practical public-key encryption schemes. In: PKC **2015**, 27–51 (2015).
18. Heuer F., Poettering B.: Selective opening security from simulatable data encapsulation. In: ASIACRYPT 2016, pp. 248–277. Springer (2016).
19. Hofheinz D.: All-but-many lossy trapdoor functions. In: EUROCRYPT 2012, pp. 209–227. Springer (2012).
20. Hofheinz D., Rao V., Wichs D.: Standard security does not imply indistinguishability under selective opening. In: TCC 2016, pp. 121–145. Springer (2016).
21. Hofheinz D., Rupp A.: Standard versus selective opening security: separation and equivalence results. In: TCC 2014, pp. 591–615. Springer (2014).
22. Huang Z., Lai J., Chen W., Au M.H., Peng Z., Li J.: Simulation-based selective opening security for receivers under chosen-ciphertext attacks. Des. Codes Cryptogr. **87**(6), 1345–1371 (2019).

23. Huang Z., Liu S., Mao X., Chen K.: Non-malleability under selective opening attacks: Implication and separation. In: ACNS 2015, pp. 87–104. Springer (2015).
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. Jia D., Libert B.: SO-CCA secure PKE from pairing based all-but-many lossy trapdoor functions. Des. Codes Cryptogr. **89**(5), 895–923 (2021).
26. Jia D., Liu Y., Li B.: IBE with tight security against selective opening and chosen-ciphertext attacks. Des. Codes Cryptogr. **88**, 1371–1400 (2020).
27. Jia D., Lu X., Li B.: Receiver selective opening security from indistinguishability obfuscation. In: INDOCRYPT 2016, pp. 393–410. Springer (2016).
28. Jia D., Lu X., Li B.: Constructions secure against receiver selective opening and chosen ciphertext attacks. In: CT-RSA 2017, pp. 417–431. Springer (2017).
29. Kitagawa F., Tanaka K.: Key dependent message security and receiver selective opening security for identity-based encryption. In: PKC 2018, pp. 32–61. Springer (2018).
30. Lai J., Deng R.H., Liu S., Weng J., Zhao Y.: Identity-based encryption secure against selective opening chosen-ciphertext attack. In: EUROCRYPT 2014, pp. 77–92. Springer (2014).
31. Lai J., Yang R., Huang Z., Weng J.: Simulation-based bi-selective opening security for public key encryption. In: ASIACRYPT 2021, pp. 456–482. Springer (2021).
32. 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, pp. 332–364. Springer (2017).
33. Liu S., Paterson K.G.: Simulation-based selective opening CCA security for PKE from key encapsulation mechanisms. In: PKC 2015, pp. 3–26. Springer (2015).
34. Lu Y., Hara K., Tanaka K.: Receiver selective opening CCA secure public key encryption from various assumptions. In: Provable and Practical Security 2020, pp. 213–233. Springer (2020).
35. Lyu L., Liu S., Han S., Gu D.: Tightly SIM-SO-CCA secure public key encryption from standard assumptions. In: PKC 2018, pp. 62–92. Springer (2018).
36. Naor M., Yung M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437. Citeseer (1990).
37. Peikert C., Shiehian S.: Noninteractive zero knowledge for np from (plain) learning with errors. In: CRYPTO 2019, pp. 89–114. Springer (2019).
38. Sahai A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS 1999, pp. 543–553. IEEE (1999).
39. Sahai A.: Simulation-sound non-interactive zero knowledge (2001). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.3.7990&rep=rep1&type=pdf.
40. Waters B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: CRYPTO 2009, pp. 619–636. Springer (2009).
41. Yang P., Kitagawa T., Hanaoka G., Zhang R., Matsuura K., Imai H.: Applying Fujisaki-Okamoto to identity-based encryption. In: Applied Algebra. Algebraic Algorithms and Error-Correcting Codes - AAECC 2006, pp. 183–192. Springer, Berlin (2006).
42. Yang R., Lai J., Huang Z., Au M.H., Xu Q., Susilo W.: Possibility and impossibility results for receiver selective opening secure PKE in the multi-challenge setting. In: ASIACRYPT 2020, pp. 191–220. Springer (2020).