Tightly SIM-SO-CCA Secure Public Key Encryption from Standard Assumptions

Lin Lyu[1,2], Shengli Liu[1,2,3(✉)], Shuai Han[1,2,4], and Dawu Gu[5,1]

[1] Dept. of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
{lvlin,slliu,dalen17,dwgu}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Westone Cryptologic Research Center, Beijing 100070, China
[4] Karlsruhe Institute of Technology, Karlsruhe, Germany
[5] Shanghai Institute for Advanced Communication and Data Science,
Shanghai, China

Abstract. Selective opening security (SO security) is desirable for public key encryption (PKE) in a multi-user setting. In a selective opening attack, an adversary receives a number of ciphertexts for possibly correlated messages, then it opens a subset of them and gets the corresponding messages together with the randomnesses used in the encryptions. SO security aims at providing security for the unopened ciphertexts. Among the existing simulation-based, selective opening, chosen ciphertext secure (SIM-SO-CCA secure) PKEs, only one (Libert *et al.* Crypto'17) enjoys *tight* security, which is reduced to the Non-Uniform LWE assumption. However, their public key and ciphertext are not compact.

In this work, we focus on constructing PKE with *tight* SIM-SO-CCA security based on standard assumptions. We formalize security notions needed for key encapsulation mechanism (KEM) and show how to transform these securities into SIM-SO-CCA security of PKE through a tight security reduction, while the construction of PKE from KEM follows the general framework proposed by Liu and Paterson (PKC'15). We present two KEM constructions with tight securities based on the Matrix Decision Diffie-Hellman assumption. These KEMs in turn lead to two tightly SIM-SO-CCA secure PKE schemes. One of them enjoys not only tight security but also compact public key.

1 Introduction

Selective Opening Security. In the context of public key encryption (PKE), IND-CPA(CCA) security is widely believed to be the right security notion. However, multi-user settings enable more complicated attacks and the traditional IND-CPA(CCA) security may not be strong enough. Consider a scenario of N senders and one receiver. The senders encrypt N (possibly correlated) messages $\mathbf{m}_1, \cdots, \mathbf{m}_N$ under the receiver's public key pk using fresh randomnesses $\mathbf{r}_1, \cdots, \mathbf{r}_N$ to get ciphertexts $\mathbf{c}_1, \cdots, \mathbf{c}_N$, respectively, i.e., each sender i computes $\mathbf{c}_i = \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_i; \mathbf{r}_i)$. Upon receiving the ciphertexts $\mathbf{c}_1, \cdots, \mathbf{c}_N$, the adversary might be able to open a subset of them via implementing corruptions.

Namely, by corrupting a subset of users, say $I \subset [N]$, the adversary obtains the messages $\{\mathbf{m}_i\}_{i\in I}$ together with the randomnesses $\{\mathbf{r}_i\}_{i\in I}$. Such an attack is called selective opening attack (SOA). It is desirable that the unopened ciphertexts $\{\mathbf{c}_i\}_{i\in[N]\setminus I}$ still protect the privacy of $\{\mathbf{m}_i\}_{i\in[N]\setminus I}$, which is exactly what the SO security concerns.

The potential correlation between $\{\mathbf{m}_i\}_{i\in I}$ and $\{\mathbf{m}_i\}_{i\in[N]\setminus I}$ hinders the use of hybrid argument proof technique. Hence, traditional IND-CPA security may not imply SO security. To date, there exist two types of SO security formalizations: indistinguishability-based SO security (IND-SO, [1, 2]) and simulation-based SO security (SIM-SO, [1, 5]). According to whether the adversary has access to a decryption oracle, these securities are further classified into IND-SO-CPA, IND-SO-CCA, SIM-SO-CPA and SIM-SO-CCA.

Intuitively, IND-SO security requires that, given public key pk, ciphertexts $\{\mathbf{c}_i\}_{i\in[N]}$, the opened messages $\{\mathbf{m}_i\}_{i\in I}$ and randomnesses $\{\mathbf{r}_i\}_{i\in I}$ (together with a decryption oracle in the CCA case), the unopened messages $\{\mathbf{m}_i\}_{i\in[N]\setminus I}$ remain computationally indistinguishable from independently sampled messages conditioned on the already opened messages $\{\mathbf{m}_i\}_{i\in I}$. Accordingly, the IND-SO security usually requires the message distributions be *efficiently conditionally resamplable* [1, 10, 11] (and such security is referred to as *weak* IND-SO security in [2]), which limits its application scenarios.

On the other hand, SIM-SO security is conceptually similar to semantic security [9]. It requires that the output of the SO adversary can be simulated by a simulator which only takes the opened messages $\{\mathbf{m}_i\}_{i\in I}$ as its input after it assigns the corruption set I. Since there is no restriction on message distribution, SIM-SO security has an advantage over IND-SO security from an application point of view. SIM-SO security was also shown to be stronger than (weak) IND-SO security in [2]. However, as shown in [13], SIM-SO security turns out to be significantly harder to achieve.

Generally speaking, there are two approaches to achieve SIM-SO-CCA security. The first approach uses lossy trapdoor functions [22], All-But-N lossy trapdoor functions [10] or All-But-Many lossy trapdoor functions [11] to construct lossy encryption schemes. If this lossy encryption has an efficient opener, then the resulting PKE scheme can be proven to be SIM-SO-CCA secure as shown in [1]. A DCR-based scheme in [11] and a LWE-based scheme in [18] are the only two schemes known to have such an opener. The second approach uses extended hash proof system and cross-authentication codes (XACs) [6]. As pointed out in [14, 15], a stronger property of XAC is required to make this proof rigorous. Following this line of research, Liu and Paterson proposed a general framework for constructing SIM-SO-CCA PKE from a special kind of key encapsulation mechanism (KEM) in combination with a strengthened XAC [19].

Tight Security Reductions. Usually, the security of a cryptographic primitive is established on the hardness of some underlying mathematical problems through a *security reduction*. It shows that any successful probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ breaking the cryptographic primitive with

advantage $\epsilon_{\mathcal{A}}$ can be transformed into a successful PPT problem solver $\mathcal{B}$ for the underlying hard problem with advantage $\epsilon_{\mathcal{B}}$. The ideal case is $\epsilon_{\mathcal{A}} = \epsilon_{\mathcal{B}}$. However, most reductions suffer from a loss in the advantage, for example, $\epsilon_{\mathcal{A}} = L \cdot \epsilon_{\mathcal{B}}$ where L is called *security loss factor* of the reduction. Smaller L always indicates a better security level for a fixed security parameter. For a PKE scheme, L usually depends on λ (the security parameter) as well as Q_e (the number of challenge ciphertexts) and Q_d (the number of decryption queries). A security reduction for a PKE scheme is *tight* and the PKE scheme is called a *tightly secure* one [7, 12] if L depends only on the security parameter λ[6] (and is independent of both Q_e and Q_d). Note that for concrete settings, λ is much smaller than Q_e and Q_d (for example, $\lambda = 80$ and Q_e, Q_d can be as large as 2^{20} or even 2^{30} in some settings). Most reductions are not tight and it appears to be a non-trivial problem to construct tightly IND-CCA secure PKE schemes.

Among the existing SIM-SO-CCA secure PKEs, only one of them has a tight security reduction [18]. Very recently, Libert *et al.* [18] provide an all-but-many lossy trapdoor function with an efficient opener, leading to a tightly SIM-SO-CCA secure PKE based on the Non-Uniform LWE assumption. Note that, their construction relies on a specific tightly secure PRF which is computable in NC^1. So far, no construction of such a PRF based on standard LWE assumption is known, which is why their PKE has to rely on a non-standard assumption. Meanwhile, there is no PKE scheme enjoying both tight SIM-SO-CCA security and compact public key & ciphertext up to now.

1.1 Our Contribution

We explore how to construct tightly SIM-SO-CCA secure PKE based on standard assumptions. Following the KEM+XAC framework proposed in [19],

- we characterize stronger security notions needed for KEM and present a tightness preserving security reduction, which shows the PKE is tightly SIM-SO-CCA secure as long as the underlying KEM is tightly secure;
- we present two KEM instantiations and prove that their security can be tightly reduced to the Matrix Decision Diffie-Hellman (MDDH) assumption, thus leading to two tightly SIM-SO-CCA secure PKE schemes. One of them enjoys not only tight security but also compact public key.

1.2 Technique Overview

Roughly speaking, to prove the SIM-SO-CCA security of a PKE (see for Definition 1), for any PPT adversary, we need to construct a simulator and show that the adversary's outputs are indistinguishable with those of the simulator. Naturally, such a simulator can be realized simply by simulating the entire real SO-CCA environment, invoking the adversary and returning the adversary's

[6] According to [3, 8], such a security reduction is called an *almost* tight one and a security reduction is tight only if L is a constant.

outputs. However, due to lack of essential information like messages and randomnesses, the simulator is not able to provide a perfect environment directly. Therefore, both the PKE scheme and the simulator has to be carefully designed, so that the simulator is able to provide the adversary a *computational indistinguishable* environment. To this end, we have to solve two problems.

- The first problem is how the simulator prepares ciphertexts for the adversary without knowing the messages.
- The second problem is how the simulator prepares randomnesses for the adversary according to the opened messages $\{\mathbf{m}_i\}_{i\in I}$ that it receives later.

To solve the first problem, the simulator has to provide ciphertexts that are computational indistinguishable with real ciphertexts in the setting of selective opening (together with chosen-ciphertext attacks). As to the second problem, note that the adversary can always check the consistence between $\{\mathbf{m}_i\}_{i\in I}, \{\mathbf{c}_i\}_{i\in I}$ and the randomnesses by re-encryption. Therefore, the simulator should not only provide indistinguishable ciphertexts but also be able to explain these ciphertexts as encryptions of any designated messages.

Liu and Paterson [19] solved these two problems and proposed a general framework for constructing SIM-SO-CCA secure PKE with the help of KEM in combination with XAC. Their PKE construction encrypts message in a bitwise manner. Suppose the message $\mathbf{m}$ has bit length ℓ. If the i-th bit of $\mathbf{m}$ is 1 ($\mathbf{m}_i = 1$), a pair of encapsulation ψ_i and key γ_i is generated from KEM, i.e., $(\psi_i, \gamma_i) \leftarrow_{\$} \mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$. If $\mathbf{m}_i = 0$, a random pair is generated, i.e., $(\psi_i, \gamma_i) \leftarrow_{\$} \Psi \times \Gamma$. Then a tag T is generated to bind up $(\gamma_1, \cdots, \gamma_\ell)$ and $(\psi_1, \cdots, \psi_\ell)$ via XAC. And the final ciphertext is $C = (\psi_1, \cdots, \psi_\ell, T)$.

They construct a simulator in the following way.

• Without knowledge of the message, the simulator uses an encryption of 1^ℓ as the ciphertext. Thus the encryption involves ℓ encapsulated pairs $(\psi_i, \gamma_i) \leftarrow_{\$} \mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$. The simulator then saves all the randomnesses used in these encapsulations.

• When providing the randomnesses for the opened messages, the simulator checks the opened messages bit by bit. If a specific bit is 1, then the simulator outputs the original randomnesses and the simulation is perfect. Otherwise, the simulator views the encapsulated pair as a random pair. Then the simulator resamples randomnesses as if this pair is randomly chosen using these resampled randomnesses.

Thanks to the bit-wise encryption mode and the resampling property of spaces Ψ and Γ, an encapsulation pair (encrypting bit 1) can be easily explained as a random pair (encrypting bit 0). Therefore the second problem is solved.

To solve the first problem, one has to show that the encapsulated pairs and the random pairs are computationally indistinguishable. In [19], a special security named IND-tCCCA is formalized for KEM. This security guarantees that *one* encapsulated pair is computationally indistinguishable with *one* random pair even when a constrained decryption oracle is provided. With the help of IND-tCCCA security of KEM, the indistinguishability between the encryption of 1^ℓ

and the encryption of real messages are proved with ℓ hybrid arguments, each hybrid replacing only one encapsulated pair with one random pair.

To pursue tight security reduction, the ℓ hybrid arguments have to be avoided. To this end, we enhance the IND-tCCCA security and consider the pseudorandomness for *multiple* pairs even when a constrained decryption oracle is provided. This new security for KEM is formalized as mPR-CCCA security in Definition 5. Armed with this enhanced security, it is possible to replace the ℓ encapsulated pairs once for all in the security reduction from the SIM-SO-CCA security of PKE to the mPR-CCCA security of KEM. However, this gives rise to another problem. The SIM-SO-CCA adversary $\mathcal{A}$ may submit a fresh ciphertext which shares the same encapsulation ψ with some challenge encapsulation. In the security reduction, the adversary $\mathcal{B}$, who invokes $\mathcal{A}$ to attack the mPR-CCCA security of KEM, cannot ask its own decapsulation oracle to decapsulate ψ since ψ is already embedded in some challenge ciphertext for $\mathcal{A}$. To solve this problem, we define another security notion for KEM, namely, the Random Encapsulation Rejection (RER) security of KEM (cf. Definition 6). Equipped with the RER security of KEM and a security of XAC, $\mathcal{B}$ could simply set 0 as the decryption bit for ψ.

Although the enhancement from IND-tCCCA to mPR-CCCA is conceptually simple, finding an mPR-CCCA secure KEM instantiation with tight reduction to standard assumptions is highly non-trivial. Inspired by the recent work on constructing tightly IND-CCA secure PKE [7, 8], we are able to give two tightly mPR-CCCA & RER secure KEM instantiations, one of which also enjoys compact public key.

1.3 Instantiation Overview

We provide two KEM instantiations.

The first KEM instantiation is inspired by a recent work in Eurocrypt'16. In the work [7], Gay *et al.* proposed the first tightly multi-challenge IND-CCA secure PKE scheme based on the MDDH assumption. From their PKE construction, we extract a KEM and tightly prove its mPR-CCCA security & RER security based on the MDDH assumption.[7]

The second KEM instantiation is contained in a very recent work by Gay *et al.* [8] in Crypto'17. In [8], a qualified proof system (QPS) is proposed to construct multi-challenge IND-CCCA secure KEM, which can be used to obtain a tightly multi-challenge IND-CCA secure PKE scheme with help of an authenticated encryption scheme. Note that our mPR-CCCA security is stronger than multi-challenge IND-CCCA security. To achieve mPR-CCCA security, we formalize a so-called Pseudorandom Simulated Proof property for QPS. We prove that if QPS has this property, the KEM from QPS is mPR-CCCA secure. Finally, we show that the QPS in [8] possesses the pseudorandom simulated proof property.

[7] In [20], a PKE with tight SIM-SO-CCA security is constructed directly on the MDDH assumption. Our work unified their work by characterizing the mPR-CCCA security and RER security for KEM.

Compared with the first instantiation, the public key of our second KEM instantiation has a constant number of group elements. The compactness of public key is in turn transferred to the PKE, resulting in the first tightly SIM-SO-CCA secure PKE based on standard assumptions together with a compact public key.

2 Preliminaries

We use λ to denote the security parameter in this work. Let ε be the empty string. For $n \in \mathbb{N}$, denote by $[n]$ the set $\{1, \cdots, n\}$. Denote by $s_1, \cdots, s_n \leftarrow_\$ S$ the process of picking n elements uniformly from set S. For a PPT algorithm $\mathcal{A}$, we use $y \leftarrow \mathcal{A}(x; r)$ to denote the process of running $\mathcal{A}$ on input x with randomness r and assigning the deterministic result to y. Let $\mathcal{R}_\mathcal{A}$ be the randomness space of $\mathcal{A}$, we use $y \leftarrow_\$ \mathcal{A}(x)$ to denote $y \leftarrow \mathcal{A}(x; r)$ where $r \leftarrow_\$ \mathcal{R}_\mathcal{A}$. We use $\mathbf{T}(\mathcal{A})$ to denote the running time of $\mathcal{A}$, which is a polynomial in λ if $\mathcal{A}$ is PPT.

We use boldface letters to denote vectors or matrices. For a vector $\mathbf{m}$ of finite dimension, $|\mathbf{m}|$ denotes the dimension of the vector and $\mathbf{m}_i$ denotes the i-th component of $\mathbf{m}$. For a set $I = \{i_1, i_2, \cdots, i_{|I|}\} \subseteq [|\mathbf{m}|]$, define $\mathbf{m}_I := (\mathbf{m}_{i_1}, \mathbf{m}_{i_2}, \cdots, \mathbf{m}_{i_{|I|}})$. For all matrix $\mathbf{A} \in \mathbb{Z}_q^{\ell \times k}$ with $\ell > k$, $\overline{\mathbf{A}} \in \mathbb{Z}_q^{k \times k}$ denotes the upper square matrix of $\mathbf{A}$ and $\underline{\mathbf{A}} \in \mathbb{Z}_q^{(\ell-k) \times k}$ denotes the lower $\ell - k$ rows of $\mathbf{A}$. By $\mathsf{span}(\mathbf{A}) := \{\mathbf{Ar} \mid \mathbf{r} \in \mathbb{Z}_q^k\}$, we denote the span of $\mathbf{A}$. By $\mathsf{Ker}(\mathbf{A}^\top)$, we denote the orthogonal space of $\mathsf{span}(\mathbf{A})$. For $\ell = k$, we define the *trace* of $\mathbf{A}$ as the sum of all diagonal elements of $\mathbf{A}$, i.e., $\mathsf{trace}(\mathbf{A}) := \sum_{i=1}^k \mathbf{A}_{i,i}$.

A function $f(\lambda)$ is *negligible*, if for every $c > 0$ there exists a λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$.

We use game-based security proof. The games are illustrated using pseudo-codes in figures. By a box in a figure, we denote that the codes in the box appears in a specific game. For example, $\boxed{\boxed{G_4}\ \boxed{G_5}}$ means that G_4 contains the codes in dash box, G_5 contains the codes in oval box, and both of them contain codes in square box. Moreover, we assume that the unboxed codes are contained in all games. We use the notation $\Pr_i[\mathsf{E}]$ to denote the probability that event E occurs in game G_i, and use the notation $G \Rightarrow 1$ to denote the event that game G returns 1. All variables in games are initialized to $\perp$. We use "$\square$" to denote the end of proof of lemmas and use "$\blacksquare$" to denote the end of proof of theorems.

Due to space limitations, we refer to the full version of this paper [21] for the definitions of collision resistant hash function, universal hash function, public key encryption, the MDDH assumption and its random self-reducibility property, together with leftover hash lemma.

2.1 Prime-order Groups

Let GGen be a PPT algorithm that on input 1^λ returns $\mathcal{G} = (\mathbb{G}, q, P)$, a description of an additive cyclic group $\mathbb{G}$ with a generator P of order q which is a λ-bit prime. For $a \in \mathbb{Z}_q$, define $[a] := aP \in \mathbb{G}$ as the *implicit representation* of a in $\mathbb{G}$. More generally, for a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_q^{n \times m}$, we define $[\mathbf{A}]$ as the implicit

representation of $\mathbf{A}$ in $\mathbb{G}$, i.e., $[\mathbf{A}] := (a_{ij}P) \in \mathbb{G}^{n\times m}$. Note that from $[a] \in \mathbb{G}$ it is generally hard to compute the value a (discrete logarithm problem is hard in $\mathbb{G}$). Obviously, given $[a], [b] \in \mathbb{G}$ and a scalar $x \in \mathbb{Z}$, one can efficiently compute $[ax] \in \mathbb{G}$ and $[a+b] \in \mathbb{G}$. Similarly, for $\mathbf{A} \in \mathbb{Z}_q^{m\times n}, \mathbf{B} \in \mathbb{Z}_q^{n\times t}$, given $\mathbf{A}, \mathbf{B}$ or $[\mathbf{A}], \mathbf{B}$ or $\mathbf{A}, [\mathbf{B}]$, one can efficiently compute $[\mathbf{AB}] \in \mathbb{G}^{m\times t}$.

2.2 Simulation-based, Selective-Opening CCA Security of PKE

Let $\mathbf{m}$ and $\mathbf{r}$ be two vectors of dimension $n := n(\lambda)$. Define $\mathsf{Enc}(\mathsf{pk}, \mathbf{m}; \mathbf{r}) := (\mathsf{Enc}(\mathsf{pk}, \mathbf{m}_1; \mathbf{r}_1), \cdots, \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_n; \mathbf{r}_n))$ where $\mathbf{r}_i$ is a fresh randomness used for the encryption of $\mathbf{m}_i$ for $i \in [n]$. Then we review the SIM-SO-CCA security definition in [6]. Let $\mathcal{M}$ denote an n-message sampler, which on input a string $\alpha \in \{0,1\}^*$ outputs a message vector $\mathbf{m}$ of dimension n, i.e., $\mathbf{m} = (\mathbf{m}_1, \cdots, \mathbf{m}_n)$. Let R be any PPT relation.

$\mathsf{Exp}^{\text{so-cca-real}}_{\mathsf{PKE},\mathcal{A},n,\mathcal{M},R}(\lambda)$:	$\mathsf{Exp}^{\text{so-cca-ideal}}_{\mathcal{S},n,\mathcal{M},R}(\lambda)$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$	$(\alpha, s_1) \leftarrow_\$ \mathcal{S}_1(1^\lambda)$
$(\alpha, a_1) \leftarrow_\$ \mathcal{A}_1^{\mathsf{Dec}(\cdot)}(\mathsf{pk})$	
$\mathbf{m} \leftarrow_\$ \mathcal{M}(\alpha), \mathbf{r} \leftarrow_\$ (\mathcal{R}_{\mathsf{Enc}})^n$	$\mathbf{m} \leftarrow_\$ \mathcal{M}(\alpha)$
$\mathbf{C} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{m}; \mathbf{r})$	$(I, s_2) \leftarrow_\$ \mathcal{S}_2(s_1, (1^{\lvert\mathbf{m}_i\rvert})_{i\in[n]})$
$(I, a_2) \leftarrow_\$ \mathcal{A}_2^{\mathsf{Dec}_{\notin\mathbf{C}}(\cdot)}(a_1, \mathbf{C})$	
$\hat{\mathbf{r}}_I \leftarrow \mathbf{r}_I$	$out_{\mathcal{S}} \leftarrow_\$ \mathcal{S}_3(s_2, \mathbf{m}_I)$
$out_{\mathcal{A}} \leftarrow_\$ \mathcal{A}_3^{\mathsf{Dec}_{\notin\mathbf{C}}(\cdot)}(a_2, \mathbf{m}_I, \hat{\mathbf{r}}_I)$	
Return $R(\mathbf{m}, I, out_{\mathcal{A}})$	Return $R(\mathbf{m}, I, out_{\mathcal{S}})$

Fig. 1. Experiments used in the definition of SIM-SO-CCA security of PKE

Definition 1 (SIM-SO-CCA Security). *A PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is simulation-based, selective-opening, chosen-ciphertext secure (SIM-SO-CCA secure) if for every PPT n-message sampler* $\mathcal{M}$*, every PPT relation* R*, every stateful PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$*, there is a stateful PPT simulator* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ *such that* $\mathsf{Adv}^{\text{so-cca}}_{\mathsf{PKE},\mathcal{A},\mathcal{S},n,\mathcal{M},R}(\lambda)$ *is negligible, where*

$$\mathsf{Adv}^{\text{so-cca}}_{\mathsf{PKE},\mathcal{A},\mathcal{S},n,\mathcal{M},R}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\text{so-cca-real}}_{\mathsf{PKE},\mathcal{A},n,\mathcal{M},R}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{so-cca-ideal}}_{\mathcal{S},n,\mathcal{M},R}(\lambda) = 1\right]\right|.$$

Experiments $\mathsf{Exp}^{\text{so-cca-real}}_{\mathsf{PKE},\mathcal{A},n,\mathcal{M},R}(\lambda)$ *and* $\mathsf{Exp}^{\text{so-cca-ideal}}_{\mathcal{S},n,\mathcal{M},R}(\lambda)$ *are defined in Figure 1. Here the restriction on* $\mathcal{A}$ *is that* $\mathcal{A}_2, \mathcal{A}_3$ *are not allowed to query the decryption oracle* $\mathsf{Dec}(\cdot)$ *with any challenge ciphertext* $\mathbf{C}_i \in \mathbf{C}$*.*

2.3 Efficiently Samplable and Explainable (ESE) Domain

A domain $\mathcal{D}$ is said to be *efficiently samplable and explainable* (ESE) [6] if there exist two PPT algorithms $(\mathsf{Sample}_{\mathcal{D}}, \mathsf{Sample}^{-1}_{\mathcal{D}})$ where $\mathsf{Sample}_{\mathcal{D}}(1^\lambda)$ outputs a

uniform element over $\mathcal{D}$ and $\mathsf{Sample}_{\mathcal{D}}^{-1}(x)$, on input $x \in \mathcal{D}$, outputs r that is uniformly distributed over the set $\{r \in \mathcal{R}_{\mathsf{Sample}_{\mathcal{D}}} \mid \mathsf{Sample}_{\mathcal{D}}(1^\lambda; r) = x\}$.

It was shown by Damgård and Nielsen in [4] that any dense subset of an efficiently samplable domain is ESE as long as the dense subset admits an efficient membership test.

2.4 Cross-Authentication Codes

The concept of XAC was first proposed by Fehr *et al.* in [6] and later adapted to strong XAC in [15] and strengthened XAC in [17].

Definition 2 (ℓ-Cross-Authentication Code, XAC). *An ℓ-cross-authentication code* XAC *(for $\ell \in \mathbb{N}$) consists of three PPT algorithms* $(\mathsf{XGen}, \mathsf{XAuth}, \mathsf{XVer})$ *and two associated spaces, the key space $\mathcal{XK}$ and the tag space $\mathcal{XT}$. The key generation algorithm* $\mathsf{XGen}(1^\lambda)$ *outputs a uniformly random key $K \in \mathcal{XK}$, the authentication algorithm* $\mathsf{XAuth}(K_1, \cdots, K_\ell)$ *takes ℓ keys $(K_1, \cdots, K_\ell) \in \mathcal{XK}^\ell$ as input and outputs a tag $T \in \mathcal{XT}$, and the verification algorithm* $\mathsf{XVer}(K, T)$ *outputs a decision bit.*
Correctness. $\mathsf{fail}_{\mathsf{XAC}}(\lambda) := \Pr[\mathsf{XVer}(K_i, \mathsf{XAuth}(K_1, \cdots, K_\ell)) \neq 1]$ *is negligible for all $i \in [\ell]$, where the probability is taken over $K_1, \cdots, K_\ell \leftarrow_\$ \mathcal{XK}$.*
Security against impersonation and substitution attacks. *Define* $\epsilon^{\mathrm{imp}}_{\mathsf{XAC}}(\lambda) := \max_{T'} \Pr[\mathsf{XVer}(K, T') = 1 \mid K \leftarrow_\$ \mathcal{XK}]$ *where* max *is over all $T' \in \mathcal{XT}$, and* $\epsilon^{\mathrm{sub}}_{\mathsf{XAC}}(\lambda) := \max_{i, K_{\neq i}, F} \Pr\left[\begin{array}{c|c} T' \neq T & K_i \leftarrow_\$ \mathcal{XK}, \\ \mathsf{XVer}(K_i, T') = 1 & T \leftarrow \mathsf{XAuth}(K_1, \cdots, K_\ell), \\ & T' \leftarrow F(T) \end{array}\right]$
where max *is over all $i \in [\ell]$, all $K_{\neq i} := (K_j)_{j \in [\ell \setminus i]} \in \mathcal{XK}^{\ell-1}$ and all (possibly randomized) functions $F : \mathcal{XT} \to \mathcal{XT}$. Then we say* XAC *is secure against impersonation and substitution attacks if both $\epsilon^{\mathrm{imp}}_{\mathsf{XAC}}(\lambda)$ and $\epsilon^{\mathrm{sub}}_{\mathsf{XAC}}(\lambda)$ are negligible.*

Definition 3 (Strong and semi-unique XACs). *An ℓ-cross-authentication code* XAC *is* strong *and* semi-unique *if it has the following two properties.*
Strongness [15]. *There exists a PPT algorithm* ReSamp*, which takes as input $T \in \mathcal{XT}$ and $i \in [\ell]$, with $K_1, \cdots, K_\ell \leftarrow_\$ \mathsf{XGen}(1^\lambda), T \leftarrow \mathsf{XAuth}(K_1, \cdots, K_\ell)$, and outputs $\hat{K}_i \in \mathcal{XK}$, denoted by $\hat{K}_i \leftarrow_\$ \mathsf{ReSamp}(T, i)$. Suppose for each fixed $(k_1, \cdots, k_{\ell-1}, t) \in (\mathcal{XK})^{\ell-1} \times \mathcal{XT}$, the statistical distance between $\hat{K}_i$ and K_i, conditioned on $(K_{\neq i}, T) = (k_1, \cdots, k_{\ell-1}, t)$, is bounded by $\delta(\lambda)$, i.e.,*

$$\frac{1}{2} \sum_{k \in \mathcal{XK}} \left| \begin{array}{c} \Pr[\hat{K}_i = k \mid (K_{\neq i}, T) = (k_1, \cdots, k_{\ell-1}, t)] \\ - \\ \Pr[K_i = k \mid (K_{\neq i}, T) = (k_1, \cdots, k_{\ell-1}, t)] \end{array} \right| \leq \delta(\lambda).$$

Then the code XAC *is said to be $\delta(\lambda)$-*strong *or* strong *if $\delta(\lambda)$ is negligible.*
Semi-Uniqueness [17]. *The code* XAC *is said to be* semi-unique *if $\mathcal{XK} = \mathcal{K}_x \times \mathcal{K}_y$, and given $T \in \mathcal{XT}$ and $K^x \in \mathcal{K}_x$, there exists at most one $K^y \in \mathcal{K}_y$ such that* $\mathsf{XVer}((K^x, K^y), T) = 1$.

See the full version [21] for a concrete XAC instantiation by Fehr *et al.* in [6].

3 Key Encapsulation Mechanism

In this section, we recall the definition of key encapsulation mechanism and formalize two new security notions for it.

Definition 4 (Key Encapsulation Mechanism). *A* KEM KEM *is a tuple of PPT algorithms* $(\mathsf{KGen}, \mathsf{KEnc}, \mathsf{KDec})$ *such that,* $\mathsf{KGen}(1^\lambda)$ *generates a (public, secret) key pair* $(\mathsf{pk}_{\mathsf{kem}}, \mathsf{sk}_{\mathsf{kem}})$*;* $\mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$ *returns an encapsulation* $\psi \in \Psi$ *and a key* $\gamma \in \Gamma$*, where* Ψ *is the encapsulation space and* Γ *is the key space;* $\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi)$ *deterministically decapsulates* ψ *with* $\mathsf{sk}_{\mathsf{kem}}$ *to get* $\gamma \in \Gamma$ *or* $\perp$*.*

We say KEM *is* perfectly correct *if for all* λ, $\Pr[\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi) = \gamma] = 1$, *where* $(\mathsf{pk}_{\mathsf{kem}}, \mathsf{sk}_{\mathsf{kem}}) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)$ *and* $(\psi, \gamma) \leftarrow_{\$} \mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$.

3.1 mPR-CCCA Security for KEM

We formalize a new security notion for KEM, namely mPR-CCCA. Roughly speaking, mPR-CCCA security guarantees pseudorandomness of multiple (ψ, γ) pairs outputted by KEnc even if a constrained decapsulation oracle is provided.

Definition 5 (mPR-CCCA Security for KEM). *Let* $\mathcal{A}$ *be an adversary and* $b \in \{0, 1\}$ *be a bit. Let* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{KEnc}, \mathsf{KDec})$ *be a KEM with encapsulation space* Ψ *and key space* Γ*. Define the experiment* $\mathsf{Exp}^{\text{mpr-ccca-}b}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ *in Figure 2.*

$\underline{\mathsf{Exp}^{\text{mpr-ccca-}b}_{\mathsf{KEM},\mathcal{A}}(\lambda)\text{: } //b \in \{0,1\}}$	$\underline{\mathcal{O}_{\text{enc}}():}$	$\underline{\mathcal{O}_{\text{dec}}(\mathsf{pred}, \psi):}$
$(\mathsf{pk}_{\mathsf{kem}}, \mathsf{sk}_{\mathsf{kem}}) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)$	$(\psi_0, \gamma_0) \leftarrow_{\$} \Psi \times \Gamma$	$\gamma \leftarrow \mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi)$
$b' \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{enc}}(), \mathcal{O}_{\text{dec}}(\cdot,\cdot)}(\mathsf{pk}_{\mathsf{kem}})$	$(\psi_1, \gamma_1) \leftarrow_{\$} \mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$	Return $\begin{cases} \gamma & \text{If } \begin{pmatrix} \psi \notin \boldsymbol{\psi}_{\text{enc}} \wedge \\ \mathsf{pred}(\gamma) = 1 \end{pmatrix} \\ \perp & \text{Otherwise} \end{cases}$
Return b'	$\boldsymbol{\psi}_{\text{enc}} \leftarrow \boldsymbol{\psi}_{\text{enc}} \cup \{\psi_b\}$	
	Return (ψ_b, γ_b)	

Fig. 2. Experiment used in the definition of mPR-CCCA security of KEM

In $\mathsf{Exp}^{\text{mpr-ccca-}b}_{\mathsf{KEM},\mathcal{A}}(\lambda)$, $\mathsf{pred} : \Gamma \cup \{\perp\} \to \{0, 1\}$ *denotes a PPT predicate and* $\mathsf{pred}(\perp) := 0$*. Let* Q_{dec} *be the total number of decapsulation queries made by* $\mathcal{A}$*, which is independent of the environment without loss of generality. The* uncertainty *of* $\mathcal{A}$ *is defined as* $\mathit{uncert}_{\mathcal{A}}(\lambda) := \frac{1}{Q_{dec}} \sum_{i=1}^{Q_{dec}} \Pr_{\gamma \leftarrow_{\$} \Gamma}[\mathsf{pred}_i(\gamma) = 1]$*, where* pred_i *is the predicate in the* i*-th* $\mathcal{O}_{dec}$ *query.*

We say KEM *has* multi-encapsulation pseudorandom security against constrained CCA adversaries (mPR-CCCA security) *if for each PPT adversary* $\mathcal{A}$ *with negligible uncertainty* $\mathit{uncert}_{\mathcal{A}}(\lambda)$*, the advantage* $\mathsf{Adv}^{\text{mpr-ccca}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ *is negligible, where* $\mathsf{Adv}^{\text{mpr-ccca}}_{\mathsf{KEM},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Exp}^{\text{mpr-ccca-0}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{mpr-ccca-1}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = 1\right] \right|$.

Note that the afore-defined mPR-CCCA security implies multi-challenge IND-CCCA security defined in [8].

3.2 RER Security of KEM

We define Random Encapsulation Rejection security for KEM which requires the decapsulation of a random encapsulation is rejected overwhelmingly.

Definition 6 (Random Encapsulation Rejection Security for KEM). *Let* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{KEnc}, \mathsf{KDec})$ *be a KEM with encapsulation space* Ψ *and key space* Γ*. Let* $\mathcal{A}$ *be a stateful adversary and* $b \in \{0,1\}$ *be a bit. Define the following experiment* $\mathsf{Exp}^{\text{rer-}b}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ *in Figure 3.*

$\underline{\mathsf{Exp}^{\text{rer-}b}_{\mathsf{KEM},\mathcal{A}}(\lambda)}$: //$b \in \{0,1\}$	$\underline{\mathcal{O}_{\text{cha}}(\mathsf{pred}, \psi)}$:
$(\mathsf{pk}_{\mathsf{kem}}, \mathsf{sk}_{\mathsf{kem}}) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)$	If $\psi \notin \boldsymbol{\psi}_{\text{ran}}$:
$\boldsymbol{\psi}_{\text{ran}} \leftarrow \emptyset$	Return $\mathsf{pred}(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi))$
$(st, 1^n) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{cha}}(\cdot,\cdot)}(\mathsf{pk}_{\mathsf{kem}})$	If $b = 1$:
$\boldsymbol{\psi}_{\text{ran}} = \{\psi_1, \cdots, \psi_n\} \leftarrow_{\$} \Psi^n$	Return $\mathsf{pred}(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi))$
$b' \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{cha}}(\cdot,\cdot)}(st, \boldsymbol{\psi}_{\text{ran}})$	Else:
Return b'	Return 0

Fig. 3. Experiment used in the definition of RER property of KEM

In $\mathsf{Exp}^{\text{rer-}b}_{\mathsf{KEM},\mathcal{A}}(\lambda)$, $\mathsf{pred} : \Gamma \cup \{\perp\} \to \{0,1\}$ *denotes a PPT predicate and* $\mathsf{pred}(\perp) := 0$*. Let* Q_{cha} *be the total number of* $\mathcal{O}_{cha}$ *queries made by* $\mathcal{A}$*, which is independent of the environment without loss of generality. The* uncertainty *of* $\mathcal{A}$ *is defined as* $uncert_{\mathcal{A}}(\lambda) := \frac{1}{Q_{cha}} \sum_{i=1}^{Q_{cha}} \Pr_{\gamma \leftarrow_{\$} \Gamma}[\mathsf{pred}_i(\gamma) = 1]$*, where* pred_i *is the predicate in the* i*-th* $\mathcal{O}_{cha}$ *query.*

We say KEM *has* Random Encapsulation Rejection security (RER security) *if for each PPT adversary* $\mathcal{A}$ *with negligible uncertainty* $uncert_{\mathcal{A}}(\lambda)$*, the advantage*

$$\mathsf{Adv}^{\text{rer}}_{\mathsf{KEM},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\text{rer-0}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{rer-1}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = 1\right]\right| \text{ is negligible.}$$

4 SIM-SO-CCA Secure PKE from KEM

4.1 PKE Construction

In Figure 4, we recall the general framework for constructing SIM-SO-CCA secure PKE proposed in [19]. A small difference from [19] is that we make use of hash function H_1 to convert the key space of KEM to the key space of XAC.

Ingredients. This construction uses the following ingredients.

- KEM=$(\mathsf{KGen}, \mathsf{KEnc}, \mathsf{KDec})$ with key space Γ & ESE encapsulation space Ψ.
- $(\ell+1)$-XAC XAC with ESE key space $\mathcal{XK} = \mathcal{K}_x \times \mathcal{K}_y$.
- Hash function $\mathsf{H}_1 : \Gamma \to \mathcal{XK}$ generated by hash function generator $\mathcal{H}_1(1^\lambda)$.
- Hash function $\mathsf{H}_2 : \Psi^\ell \to \mathcal{K}_y$ generated by hash function generator $\mathcal{H}_2(1^\lambda)$.

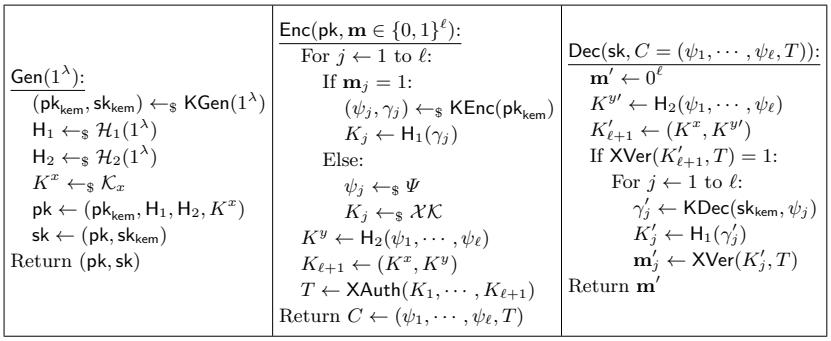

Fig. 4. Construction of $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

4.2 Tight Security Proof of PKE

In this subsection, we prove the SIM-SO-CCA security of PKE with tight reduction to the security of KEM. We state our main result in the following theorem.

Theorem 1. *Suppose the KEM* KEM *is mPR-CCCA and RER secure, the* $(\ell+1)$*-cross-authentication code* XAC *is* $\delta(\lambda)$*-strong, semi-unique, and secure against impersonation and substitution attacks;* $\mathcal{H}_1$ *is universal;* $\mathcal{H}_2$ *outputs collision resistant function. Then the PKE scheme* PKE *constructed in Figure 4 is SIM-SO-CCA secure. More precisely, for each PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ *against* PKE *in the SIM-SO-CCA real experiment, for each PPT* n*-message sampler* $\mathcal{M}$*, and each PPT relation* R*, we can construct a stateful PPT simulator* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ *for the SIM-SO-CCA ideal experiment and PPT adversaries* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *with* $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{B}_3) \leq \mathbf{T}(\mathcal{A}) + Q_{\mathrm{dec}} \cdot \mathsf{poly}(\lambda)$*, such that*

$$\begin{aligned}\mathsf{Adv}^{\text{so-cca}}_{\mathsf{PKE},\mathcal{A},\mathcal{S},n,\mathcal{M},R}(\lambda) \leq\ & \mathsf{Adv}^{\text{mpr-ccca}}_{\mathsf{KEM},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\text{rer}}_{\mathsf{KEM},\mathcal{B}_3}(\lambda) + \ell \cdot Q_{\mathrm{dec}} \cdot \epsilon^{\text{sub}}_{\mathsf{XAC}}(\lambda) \\ & + 2\mathsf{Adv}^{\text{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + (n\ell) \cdot (\delta(\lambda) + \Delta), \end{aligned} \tag{1}$$

where Q_{dec} *denotes the total number of* $\mathcal{A}$*'s decryption oracle queries,* $\mathsf{poly}(\lambda)$ *is a polynomial independent of* $\mathbf{T}(\mathcal{A})$ *and* $\Delta = \frac{1}{2} \cdot \sqrt{|\mathcal{XK}|/|\Gamma|}$.

Remark. If we instantiate the construction with the information-theoretically secure XAC in [6] and choose proper set $\mathcal{XK}$ and Γ, then $\Delta, \delta(\lambda)$, $\epsilon^{\text{imp}}_{\mathsf{XAC}}(\lambda)$ and $\epsilon^{\text{sub}}_{\mathsf{XAC}}(\lambda)$ are all exponentially small in λ. Then (1) turns out to be

$$\mathsf{Adv}^{\text{so-cca}}_{\mathsf{PKE},\mathcal{A},\mathcal{S},n,\mathcal{M},R}(\lambda) \leq \mathsf{Adv}^{\text{mpr-ccca}}_{\mathsf{KEM},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\text{rer}}_{\mathsf{KEM},\mathcal{B}_3}(\lambda) + 2\mathsf{Adv}^{\text{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + 2^{-\Omega(\lambda)}.$$

If the underlying KEM has tight mPR-CCCA security and RER security, then our PKE turns out to be tightly SIM-SO-CCA secure.

Proof of Theorem 1. For each PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, we can construct a stateful PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ as shown in Figure 5.

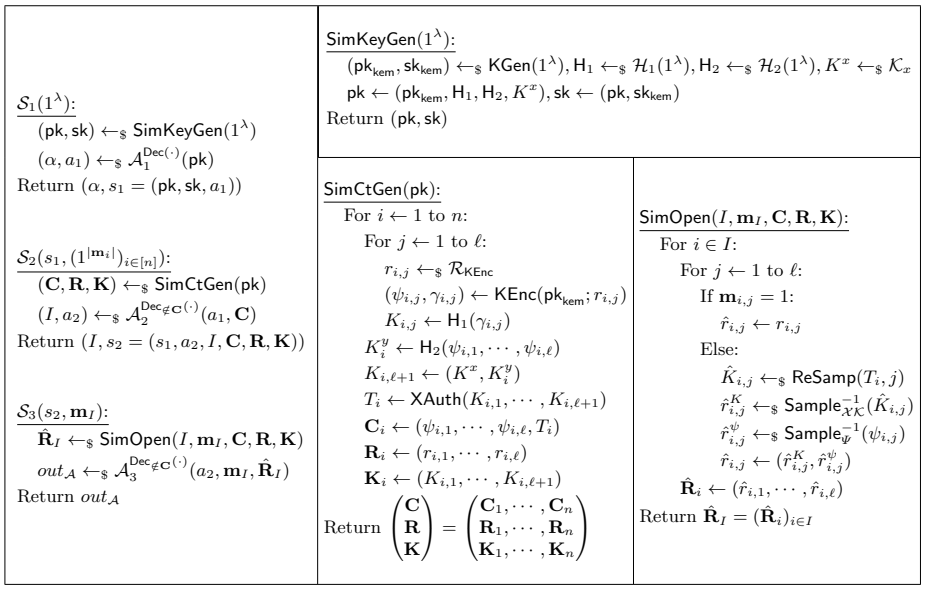

Fig. 5. Construction of simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ for $\mathsf{Exp}^{\text{so-cca-ideal}}_{\mathcal{S},n,\mathcal{M},R}(\lambda)$.

The differences between the real and the ideal experiments lie in two aspects. The first is how the challenge ciphertext vector is generated and the second is how the corrupted ciphertexts are opened. In other words, the algorithms $\mathsf{SimCtGen}$ and $\mathsf{SimOpen}$ used by the simulator differ from the real experiment. In the proof, we focus on these two algorithms and gradually change them through a series of games starting with game G_0 and ending with game G_9, with adjacent games being proved to be computationally indistinguishable. The full set of games are illustrated in Figure 6.

Game G_0. Game G_0 is exactly the ideal experiment $\mathsf{Exp}^{\text{so-cca-ideal}}_{\mathcal{S},n,\mathcal{M},R}(\lambda)$. Hence

$$\Pr\left[\mathsf{Exp}^{\text{so-cca-ideal}}_{\mathcal{S},n,\mathcal{M},R}(\lambda) = 1\right] = \Pr_0[G \Rightarrow 1]. \tag{2}$$

Game $G_0 - G_1$. The only difference between G_1 and G_0 is that a collision check for H_2 is added in G_1 and G_1 aborts if a collision is found. More precisely, we use a set $\mathcal{Q}$ to log all the (input, output) pairs for H_2 in algorithm $\mathsf{SimCtGen}$. Then in the Dec oracle, if there exists a usage of H_2 such that its output collides with some output in $\mathcal{Q}$ but with different inputs, then a collision for H_2 is found and the game G_1 aborts immediately. It is straightforward to build a PPT adversary $\mathcal{B}_1$ with $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{A}) + Q_{\text{dec}} \cdot \mathsf{poly}(\lambda)$, where $\mathsf{poly}(\lambda)$ is a polynomial independent of $\mathbf{T}(\mathcal{A})$, such that,

$$|\Pr_0[G \Rightarrow 1] - \Pr_1[G \Rightarrow 1]| \leq \mathsf{Adv}^{\text{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda). \tag{3}$$

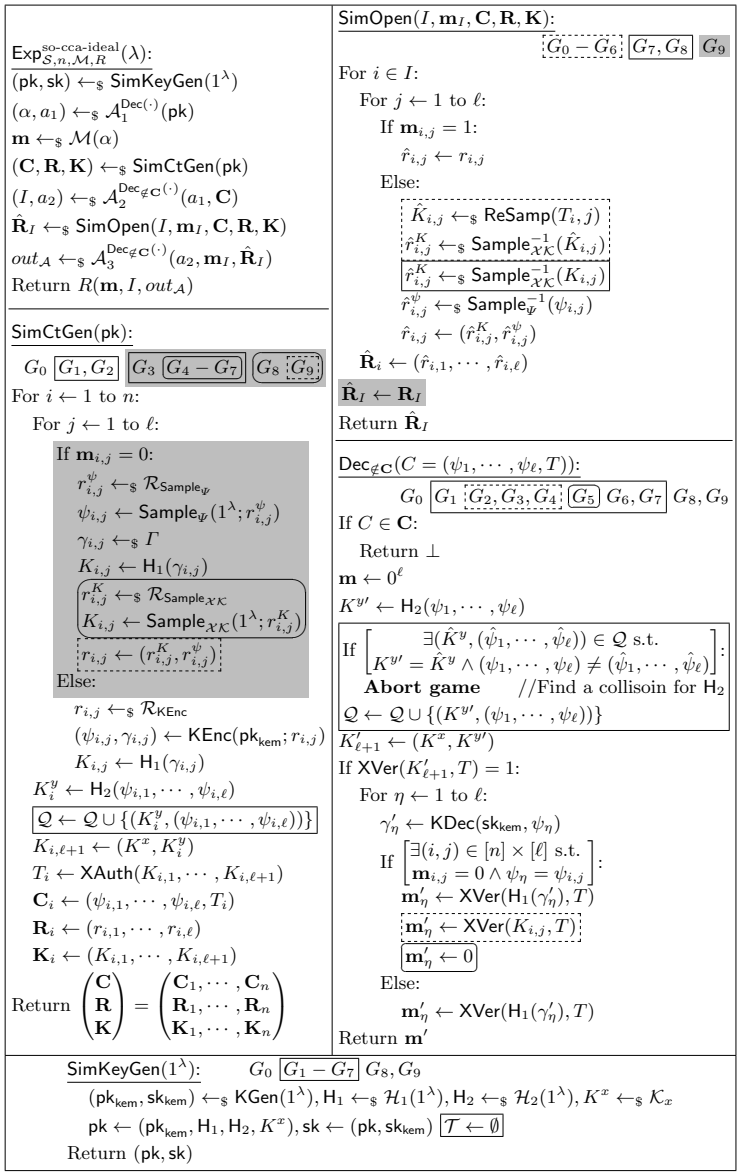

Fig. 6. Games $G_0 - G_9$ in the proof of Theorem 1.

Game $G_1 - G_2$. G_2 is essentially the same as G_1 except for one conceptual change in the Dec oracle. More precisely, for a $\mathsf{Dec}(C = (\psi_1, \cdots, \psi_\ell, T))$ query such that $\exists(i,j) \in [n] \times [\ell], \eta \in [\ell]$ s.t. $\mathbf{m}_{i,j} = 0 \wedge \psi_\eta = \psi_{i,j}$,

- in G_1, we proceed exactly the same as the decryption algorithm, i.e.,

set $\mathbf{m}'_\eta \leftarrow \mathsf{XVer}(\mathsf{H}_1(\gamma'_\eta), T)$ where $\gamma'_\eta = \mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)$;

- in G_2, we set $\mathbf{m}'_\eta \leftarrow \mathsf{XVer}(K_{i,j}, T)$.

Since $\psi_\eta = \psi_{i,j}$, $\gamma'_\eta = \mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)$ and $(\psi_{i,j}, \gamma_{i,j})$ is the output of $\mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$, we have that $\gamma'_\eta = \gamma_{i,j}$ due to the perfect correctness of KEM. Then $K_{i,j} = \mathsf{H}_1(\gamma_{i,j}) = \mathsf{H}_1(\gamma'_\eta)$. Thus the difference between G_1 and G_2 is only conceptual, and it follows

$$\Pr_1[G \Rightarrow 1] = \Pr_2[G \Rightarrow 1]. \tag{4}$$

Game $G_2 - G_3$. G_3 is almost the same as G_2 except for one change in the $\mathsf{SimCtGen}$ algorithm.

- In G_2, all $(\psi_{i,j}, \gamma_{i,j})$ pairs are the output of $\mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$.
- In G_3, for $\mathbf{m}_{i,j} = 1$, $(\psi_{i,j}, \gamma_{i,j})$ pairs are the output of $\mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$; for $\mathbf{m}_{i,j} = 0$, $(\psi_{i,j}, \gamma_{i,j})$ pairs are uniformly selected from $\Psi \times \Gamma$.

We will reduce the indistinguishability between game G_2 and G_3 to the mPR-CCCA security of KEM. Given $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, we can build a PPT adversary $\mathcal{B}_2$ with $\mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{A})$ and uncertainty $uncert_{\mathcal{B}_2}(\lambda) \leq \epsilon^{\mathrm{imp}}_{\mathsf{XAC}}(\lambda) + \Delta$ such that

$$|\Pr_2[G \Rightarrow 1] - \Pr_3[G \Rightarrow 1]| \leq \mathsf{Adv}^{\text{mpr-ccca}}_{\mathsf{KEM}, \mathcal{B}_2}(\lambda). \tag{5}$$

On input $\mathsf{pk}_{\mathsf{kem}}$, $\mathcal{B}_2$ selects $\mathsf{H}_1, \mathsf{H}_2$ and K^x itself and embeds $\mathsf{pk}_{\mathsf{kem}}$ in $\mathsf{pk} = (\mathsf{pk}_{\mathsf{kem}}, \mathsf{H}_1, \mathsf{H}_2, K^x)$. In the first phase, $\mathcal{B}_2$ calls $\mathcal{A}_1^{\mathsf{Dec}(\cdot)}(\mathsf{pk})$. To respond the decryption query $\mathsf{Dec}(C = (\psi_1, \cdots, \psi_\ell, T))$ submitted by $\mathcal{A}$, $\mathcal{B}_2$ simulates Dec until it needs to call $\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)$ to decapsulate ψ_η. Since $\mathcal{B}_2$ does not possess $\mathsf{sk}_{\mathsf{kem}}$ relative to $\mathsf{pk}_{\mathsf{kem}}$, $\mathcal{B}_2$ is not able to invoke KDec itself. Then $\mathcal{B}_2$ submits a $\mathcal{O}_{\mathrm{dec}}(\mathsf{pred}, \psi_\eta)$ query to its own oracle $\mathcal{O}_{\mathrm{dec}}$ where $\mathsf{pred}(\cdot) := \mathsf{XVer}(\mathsf{H}_1(\cdot), T)$. Clearly, this predicate is a PPT one. If the response of $\mathcal{O}_{\mathrm{dec}}$ is $\perp$, $\mathcal{B}_2$ sets $\mathbf{m}'_\eta$ to 0. Otherwise $\mathcal{B}_2$ sets $\mathbf{m}'_\eta$ to 1.

Case 1: $\mathcal{O}_{\mathrm{dec}}(\mathsf{XVer}(\mathsf{H}_1(\cdot), T), \psi_\eta) = \perp$. This happens if and only if

$$\psi_\eta \in \boldsymbol{\psi}_{\mathrm{enc}} \vee \mathsf{XVer}(\mathsf{H}_1(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)), T) = 0.$$

In the first phase, $\mathcal{B}_2$ has not submitted any $\mathcal{O}_{\mathrm{enc}}$ query yet and $\boldsymbol{\psi}_{\mathrm{enc}}$ is empty. So $\psi_\eta \notin \boldsymbol{\psi}_{\mathrm{enc}}$. In this case, $\mathcal{O}_{\mathrm{dec}}(\mathsf{XVer}(\mathsf{H}_1(\cdot), T), \psi_\eta) = \perp$ if and only if

$$\mathsf{XVer}(\mathsf{H}_1(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)), T) = 0.$$

Therefore $\mathcal{B}_2$ perfectly simulates the Dec oracle in $G_2(G_3)$ by setting $\mathbf{m}'_\eta \leftarrow 0$.

Case 2: $\mathcal{O}_{\mathrm{dec}}(\mathsf{XVer}(\mathsf{H}_1(\cdot), T), \psi_\eta) \neq \perp$. This happens if and only if

$$\psi_\eta \notin \boldsymbol{\psi}_{\mathrm{enc}} \wedge \mathsf{XVer}(\mathsf{H}_1(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)), T) = 1.$$

For the same reason as case 1, the condition $\psi_\eta \notin \boldsymbol{\psi}_{\mathrm{enc}}$ always holds. In this case, $\mathcal{O}_{\mathrm{dec}}(\mathsf{XVer}(\mathsf{H}_1(\cdot), T), \psi_\eta) \neq \perp$ if and only if $\mathsf{XVer}(\mathsf{H}_1(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)), T) = 1$. Therefore $\mathcal{B}_2$ perfectly simulates the Dec oracle in $G_2(G_3)$ by setting $\mathbf{m}'_\eta \leftarrow 1$.

In either case, $\mathcal{B}_2$ can perfectly simulate the Dec oracle for $\mathcal{A}_1$. At the end of this phase, $\mathcal{B}_2$ gets $\mathcal{A}_1$'s output (α, a_1). Then $\mathcal{B}_2$ calls $\mathbf{m} \leftarrow_{\$} \mathcal{M}(\alpha)$ and simulates algorithm $\mathsf{SimCtGen}(\mathsf{pk})$.

- If $\mathbf{m}_{i,j} = 1$, $\mathcal{B}_2$ proceeds just like game $G_2(G_3)$, i.e., $(\psi_{i,j}, \gamma_{i,j}) \leftarrow_{\$} \mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$ and set $K_{i,j} \leftarrow \mathsf{H}_1(\gamma_{i,j})$.

– If $\mathbf{m}_{i,j} = 0$, $\mathcal{B}_2$ submits an $\mathcal{O}_{\text{enc}}()$ query to its own oracle and gets the response (ψ, γ) (ψ is added into set $\boldsymbol{\psi}_{\text{enc}}$). Then $\mathcal{B}_2$ sets $(\psi_{i,j}, \gamma_{i,j}) \leftarrow (\psi, \gamma)$.
If $b = 1$, (ψ, γ) is the output of $\mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$, $\mathcal{B}_2$ perfectly simulates $\mathsf{SimCtGen}(\mathsf{pk})$ to generate challenge ciphertexts $\mathbf{C}$ in G_2.
If $b = 0$, (ψ, γ) is uniformly over $\Psi \times \Gamma$, $\mathcal{B}_2$ perfectly simulates $\mathsf{SimCtGen}(\mathsf{pk})$ to generate challenge ciphertexts $\mathbf{C}$ in G_3.

In the second phase, $\mathcal{B}_2$ calls $\mathcal{A}_2^{\mathsf{Dec}_{\notin \mathbf{C}}(\cdot)}(a_1, \mathbf{C})$ to get (I, a_2). Upon an decryption query $\mathsf{Dec}_{\notin \mathbf{C}}(C = (\psi_1, \cdots, \psi_\ell, T))$ submitted by $\mathcal{A}_2$, $\mathcal{B}_2$ responds almost in the same way as in the first phase, except that $\mathcal{B}_2$ has to deal with the case of $\exists \psi_\eta \in \boldsymbol{\psi}_{\text{enc}}$. This case does happen: even if $C = (\psi_1, \cdots, \psi_\ell, T) \notin \mathbf{C}$, it is still possible that $\exists \psi_\eta \in \{\psi_i\}_{i \in [\ell]}$ with $\psi_\eta \in \boldsymbol{\psi}_{\text{enc}}$. In this case, there is no chance for $\mathcal{B}_2$ to submit an $\mathcal{O}_{\text{dec}}(\mathsf{pred}, \psi_\eta)$ query for a useful response because the response will always be $\perp$. However, it does not matter. By the specification of $G_2(G_3)$, $\mathbf{m}'_\eta$ should be set to the output of $\mathsf{XVer}(K_{i,j}, T)$ which $\mathcal{B}_2$ can perfectly do.

Note that the execution of algorithm $\mathsf{SimOpen}$ in game $G_2(G_3)$ does not need all information about $\mathbf{R}$. Only those randomnesses with respect to $\mathbf{m}_{i,j} = 1$ are needed. Now that $\mathcal{B}_2$ does have $I, \mathbf{m}_I, \mathbf{C}, \mathbf{K}$ and part of $\mathbf{R}$ (for $\mathbf{m}_{i,j} = 1$), it can call $\mathsf{SimOpen}(I, \mathbf{m}_I, \mathbf{C}, \mathbf{R}, \mathbf{K})$ to get $\hat{\mathbf{R}}_I$.

In the third phase, $\mathcal{B}_2$ calls $\mathcal{A}_3^{\mathsf{Dec}_{\notin \mathbf{C}}(\cdot)}(a_2, \mathbf{m}_I, \hat{\mathbf{R}}_I)$ to get $out_{\mathcal{A}}$. The $\mathsf{Dec}_{\notin \mathbf{C}}$ query submitted by $\mathcal{A}$ in this phase is responded by $\mathcal{B}_2$ in the same way as in the second phase. Finally, $\mathcal{B}_2$ outputs $R(\mathbf{m}, I, out_{\mathcal{A}})$.

According to the above analysis, $\mathcal{B}_2$ perfectly simulates G_2 for $\mathcal{A}$ if $b = 1$ and perfectly simulates G_3 for $\mathcal{A}$ if $b = 0$. Moreover, for $\gamma \leftarrow_{\$} \Gamma$, $\mathsf{H}_1(\gamma)$ is Δ-close to uniform by leftover hash lemma since H_1 is universal. Then

$$\Pr_{\gamma \leftarrow_{\$} \Gamma}[\mathsf{pred}(\gamma) = 1] = \Pr_{\gamma \leftarrow_{\$} \Gamma}[\mathsf{XVer}(\mathsf{H}_1(\gamma), T) = 1] \leq \epsilon^{\text{imp}}_{\mathsf{XAC}}(\lambda) + \Delta.$$

By the definition of uncertainty, we have.

$$uncert_{\mathcal{B}_2}(\lambda) \leq \epsilon^{\text{imp}}_{\mathsf{XAC}}(\lambda) + \Delta. \tag{6}$$

Thus (5) follows.

Game $G_3 - G_4$. G_4 is almost the same as G_3 except for one change in the $\mathsf{SimCtGen}$ algorithm. In the $\mathsf{SimCtGen}$ algorithm, if $\mathbf{m}_{i,j} = 0$,

- in G_3, $K_{i,j} \leftarrow \mathsf{H}_1(\gamma_{i,j})$ for $\gamma_{i,j} \leftarrow_{\$} \Gamma$;
- in G_4, $K_{i,j}$ is uniformly selected from $\mathcal{XK}$.

Since H_1 is universal, by leftover hash lemma and a union bound, we have that

$$|\Pr_3[G \Rightarrow 1] - \Pr_4[G \Rightarrow 1]| \leq (n\ell) \cdot \Delta. \tag{7}$$

Game $G_4 - G_5$. G_5 is almost the same as G_4 except for one change in the Dec oracle. More precisely, to reply a $\mathsf{Dec}_{\notin \mathbf{C}}(C = (\psi_1, \cdots, \psi_\ell, T))$ query such that $\exists (i, j) \in [n] \times [\ell], \eta \in [\ell]$ s.t. $\mathbf{m}_{i,j} = 0 \wedge \psi_\eta = \psi_{i,j}$,

- in G_4, we set $\mathbf{m}'_\eta \leftarrow \mathsf{XVer}(K_{i,j}, T)$;
- in G_5, we set $\mathbf{m}'_\eta \leftarrow 0$ directly.

Suppose $\psi_\eta = \psi_{i,j} \in \mathbf{C}_i = (\psi_{i,1}, \cdots, \psi_{i,\ell}, T_i)$ where $T_i = \mathsf{XAuth}(K_{i,1}, \cdots, K_{i,\ell+1})$. There are two cases according to whether $T = T_i$.

Case 1: $T = T_i$. In this case, since $C \notin \mathbf{C}$, we have that $(\psi_1, \cdots, \psi_\ell) \neq (\psi_{i,1}, \cdots, \psi_{i,\ell})$. Note that $K_i^y = \mathsf{H}_2(\psi_{i,1}, \cdots, \psi_{i,\ell})$ and $K^{y\prime} = \mathsf{H}_2(\psi_1, \cdots, \psi_\ell)$. If $K_i^y = K^{y\prime}$, a collision for H_2 occurs, both G_4 and G_5 abort. Otherwise, we must have $K^{y\prime} \neq K_i^y$, hence $K'_{\ell+1} = (K^x, K^{y\prime}) \neq (K^x, K_i^y) = K_{i,\ell+1}$. Since XAC is semi-unique and $\mathsf{XVer}(K_{i,\ell+1}, T) = 1$, it holds that $\mathsf{XVer}(K'_{\ell+1}, T) \neq 1$ which implies that $\mathbf{m}'_\eta = 0$. In this case, the responses of $\mathsf{Dec}_{\notin \mathbf{C}}$ make no difference in G_4 and G_5.

Case 2: $T \neq T_i$. Note that all the information about $K_{i,j}$ is leaked to $\mathcal{A}$ only through T_i in game G_4. Thus, the probability that $\mathsf{XVer}(K_{i,j}, T) = 1$ for $T \neq T_i$ will be no more than $\epsilon^{\mathrm{sub}}_{\mathsf{XAC}}(\lambda)$.

By a union bound, we have that

$$|\Pr_4[G \Rightarrow 1] - \Pr_5[G \Rightarrow 1]| \leq \ell \cdot Q_{\mathrm{dec}} \cdot \epsilon^{\mathrm{sub}}_{\mathsf{XAC}}(\lambda). \tag{8}$$

Game $G_5 - G_6$. G_6 is almost the same as G_5 except for one change in the Dec oracle. More precisely, for a $\mathsf{Dec}(C = (\psi_1, \cdots, \psi_\ell, T))$ query such that $\exists (i,j) \in [n] \times [\ell]$ s.t. $\mathbf{m}_{i,j} = 0 \wedge \psi_\eta = \psi_{i,j}$ for any $\eta \in [\ell]$,

- in G_5, we set $\mathbf{m}'_\eta \leftarrow 0$ directly;
- in G_6, we proceed exactly the same as the decryption algorithm, i.e., setting $\mathbf{m}'_\eta \leftarrow \mathsf{XVer}(\mathsf{H}_1(\gamma'_\eta), T)$, where $\gamma'_\eta = \mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)$.

We will reduce the indistinguishability between game G_5 and G_6 to the RER security of KEM. More precisely, we can build a PPT adversary $\mathcal{B}_3$ with $\mathbf{T}(\mathcal{B}_3) \approx \mathbf{T}(\mathcal{A})$ and with uncertainty $uncert_{\mathcal{B}_3}(\lambda) \leq \epsilon^{\mathrm{imp}}_{\mathsf{XAC}}(\lambda) + \Delta$ such that

$$|\Pr_5[G \Rightarrow 1] - \Pr_6[G \Rightarrow 1]| \leq \mathsf{Adv}^{\mathrm{rer}}_{\mathsf{KEM},\mathcal{B}_3}(\lambda). \tag{9}$$

On input $\mathsf{pk}_{\mathsf{kem}}$, $\mathcal{B}_3$ selects $\mathsf{H}_1, \mathsf{H}_2$ and K^x itself and embeds $\mathsf{pk}_{\mathsf{kem}}$ in $\mathsf{pk} = (\mathsf{pk}_{\mathsf{kem}}, \mathsf{H}_1, \mathsf{H}_2, K^x)$. In the first phase, $\mathcal{B}_3$ calls $\mathcal{A}_1^{\mathsf{Dec}(\cdot)}(\mathsf{pk})$. To respond the decryption query $\mathsf{Dec}(C = (\psi_1, \cdots, \psi_\ell, T))$ submitted by $\mathcal{A}$, $\mathcal{B}_3$ simulates Dec until it needs to call $\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)$ to decapsulate ψ_η. Since $\mathcal{B}_3$ does not hold $\mathsf{sk}_{\mathsf{kem}}$ relative to $\mathsf{pk}_{\mathsf{kem}}$, $\mathcal{B}_3$ is not able to invoke KDec itself. Then $\mathcal{B}_3$ submits a $\mathcal{O}_{\mathrm{cha}}(\mathsf{pred}, \psi)$ query to its own oracle $\mathcal{O}_{\mathrm{cha}}$ where $\mathsf{pred}(\cdot) := \mathsf{XVer}(\mathsf{H}_1(\cdot), T)$ and $\psi = \psi_\eta$. Clearly, this predicate is a PPT one. Since $\boldsymbol{\psi}_{\mathrm{ran}}$ is empty set in this phase, the condition $\psi \notin \boldsymbol{\psi}_{\mathrm{ran}}$ will always hold and $\mathcal{B}_3$ will get a bit $\beta = \mathsf{pred}(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi)) = \mathsf{XVer}(\mathsf{H}_1(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)), T)$ in return. Then $\mathcal{B}_3$ sets $\mathbf{m}'_\eta \leftarrow \beta$ and perfectly simulates Dec for $\mathcal{A}$ in this phase.

At the end of this phase, $\mathcal{B}_3$ gets $\mathcal{A}$'s output (α, a_1). Then $\mathcal{B}_3$ calls $\mathbf{m} \leftarrow_{\$} \mathcal{M}(\alpha)$ and then simulates algorithm $\mathsf{SimCtGen}(\mathsf{pk})$ as follows. $\mathcal{B}_3$ first outputs $1^{n\ell}$ and get $\boldsymbol{\psi}_{\mathrm{ran}} = \{\psi_1^{\mathrm{ran}}, \cdots, \psi_{n\ell}^{\mathrm{ran}}\}$ which are $n\ell$ random encapsulations. During the generation of the challenge ciphertexts, $\mathcal{B}_3$ sets $(\psi_{i,j}, K_{i,j})$ according to $\mathbf{m}$.

- If $\mathbf{m}_{i,j} = 1$, $\mathcal{B}_3$ sets $(\psi_{i,j}, \gamma_{i,j}) \leftarrow_{\$} \mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$ and sets $K_{i,j} \leftarrow \mathsf{H}_1(\gamma_{i,j})$.
- If $\mathbf{m}_{i,j} = 0$, $\mathcal{B}_3$ sets $\psi_{i,j} \leftarrow \psi^{\mathrm{ran}}_{(i-1)\ell+j}$ and $K_{i,j} \leftarrow_{\$} \mathcal{XK}$. Since $(i,j) \in [n] \times [\ell]$, the subscript $(i-1)\ell + j \in \{1, \cdots, n\ell\}$ is well defined.

Then $\mathcal{B}_3$ proceeds just like algorithm $\mathsf{SimCtGen}(\mathsf{pk})$ in game $G_5(G_6)$.

In the second phase, $\mathcal{B}_3$ calls $\mathcal{A}_2^{\mathsf{Dec}_{\notin\mathbf{C}}(\cdot)}(a_1, \mathbf{C})$ to get (I, a_2). To respond the decryption query $\mathsf{Dec}_{\notin\mathbf{C}}(C = (\psi_1, \cdots, \psi_\ell, T))$ submitted by $\mathcal{A}$, $\mathcal{B}_3$ proceeds just like game $G_5(G_6)$. When a decapsulation of ψ_η is needed, $\mathcal{B}_3$ submits a $\mathcal{O}_{\mathrm{cha}}(\mathsf{pred}, \psi_\eta)$ query to its own oracle $\mathcal{O}_{\mathrm{cha}}$ where $\mathsf{pred}(\cdot) := \mathsf{XVer}(\mathsf{H}_1(\cdot), T)$. After that, $\mathcal{B}_3$ will get a bit β in return and $\mathcal{B}_3$ sets $\mathbf{m}'_\eta \leftarrow \beta$. Note that

- In case of $\psi_\eta \notin \boldsymbol{\psi}_{\mathrm{ran}}$, $\mathbf{m}'_\eta = \mathsf{XVer}(\mathsf{H}_1(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)), T)$, which is exactly how $\mathbf{m}'_\eta$ is computed in both game G_5 and G_6.
- In case of $\psi_\eta \in \boldsymbol{\psi}_{\mathrm{ran}}$, there must exist $(i, j) \in [n] \times [\ell]$ s.t. $\mathbf{m}_{i,j} = 0 \wedge \psi_\eta = \psi_{i,j}$. Thus $\mathbf{m}'_\eta = \mathsf{XVer}(\mathsf{H}_1(\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi_\eta)), T)$ if $b = 1$ and $\mathbf{m}'_\eta = 0$ if $b = 0$. The former case is exactly how $\mathbf{m}'_\eta$ is computed in game G_6 and the latter case is exactly how $\mathbf{m}'_\eta$ is computed in game G_5.

As a result, $\mathcal{B}_3$ perfectly simulates $\mathsf{Dec}_{\notin\mathbf{C}}$ in the second phase of game G_5 for $\mathcal{A}$ if $b = 0$ and perfectly simulates $\mathsf{Dec}_{\notin\mathbf{C}}$ in the second phase of game G_6 for $\mathcal{A}$ if $b = 1$. After $\mathcal{B}_3$ gets (I, a_2), $\mathcal{B}_3$ is able to call $\mathsf{SimOpen}(I, \mathbf{m}_I, \mathbf{C}, \mathbf{R}, \mathbf{K})$ to get $\hat{\mathbf{R}}_I$ for the similar reason as in the proof of $G_2 - G_3$.

In the third phase, $\mathcal{B}_3$ calls $\mathcal{A}_3^{\mathsf{Dec}_{\notin\mathbf{C}}(\cdot)}(a_2, \mathbf{m}_I, \hat{\mathbf{R}}_I)$ to get $out_{\mathcal{A}}$. The $\mathsf{Dec}_{\notin\mathbf{C}}$ query submitted by $\mathcal{A}$ in this phase is responded using the same way as in the second phase. Finally, $\mathcal{B}_3$ outputs $R(\mathbf{m}, I, out_{\mathcal{A}})$.

Thus $\mathcal{B}_3$ perfectly simulates G_6 for $\mathcal{A}$ if $b = 1$ and perfectly simulates G_5 for $\mathcal{A}$ if $b = 0$. Similar to (6), $uncert_{\mathcal{B}_3}(\lambda) \leq \epsilon^{\mathrm{imp}}_{\mathsf{XAC}}(\lambda) + \Delta$. Thus (9) follows.

Game $G_6 - G_7$. G_7 is almost the same as G_6 except for one change in the $\mathsf{SimOpen}$ algorithm. More precisely,

- in G_6, $\hat{r}^K_{i,j}$ is the output of $\mathsf{Sample}^{-1}_{\mathcal{XK}}(\hat{K}_{i,j})$ where $\hat{K}_{i,j} \leftarrow_{\$} \mathsf{ReSamp}(T_i, j)$;
- in G_7, $\hat{r}^K_{i,j}$ is the output of $\mathsf{Sample}^{-1}_{\mathcal{XK}}(K_{i,j})$ for the original $K_{i,j}$ generated in algorithm $\mathsf{SimCtGen}$.

In game G_6 and G_7, before the invocation of algorithm $\mathsf{SimOpen}$, only T_i leaks information about $K_{i,j}$ to $\mathcal{A}$ when $\mathbf{m}_{i,j} = 0$. Since XAC is $\delta(\lambda)$-strong, the statistical distance between the resampled $\hat{K}_{i,j} \leftarrow_{\$} \mathsf{ReSamp}(T_i, j)$ and the original $K_{i,j}$ is at most $\delta(\lambda)$. By a union bound, we have that

$$|\mathrm{Pr}_6[G \Rightarrow 1] - \mathrm{Pr}_7[G \Rightarrow 1]| \leq (n\ell) \cdot \delta(\lambda). \tag{10}$$

Game $G_7 - G_8$. G_8 is almost the same as G_7 except for the dropping of the collision check added in G_1. Similar to the proof of $G_0 - G_1$, we can show that

$$|\mathrm{Pr}_7[G \Rightarrow 1] - \mathrm{Pr}_8[G \Rightarrow 1]| \leq \mathsf{Adv}^{\mathrm{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda). \tag{11}$$

Game $G_8 - G_9$. G_9 is almost the same as G_8 except for one change in $\mathsf{SimOpen}$. More precisely,

- in G_8, the opened randomness is a "reverse sampled" randomness, i.e., $\hat{r}^K_{i,j} \leftarrow_{\$} \mathsf{Sample}^{-1}_{\mathcal{XK}}(K_{i,j})$ and $\hat{r}^\psi_{i,j} \leftarrow_{\$} \mathsf{Sample}^{-1}_{\Psi}(\psi_{i,j})$;
- in G_9, the opened randomness $(\hat{r}^K_{i,j}, \hat{r}^\psi_{i,j})$ is changed to be the original randomness used to sample $K_{i,j}$ and $\psi_{i,j}$, i.e., $(\hat{r}^K_{i,j}, \hat{r}^\psi_{i,j}) \leftarrow (r^K_{i,j}, r^\psi_{i,j})$.

This change is conceptual since Ψ and $\mathcal{XK}$ are ESE domains. Thus

$$\Pr_8[G \Rightarrow 1] = \Pr_9[G \Rightarrow 1]. \tag{12}$$

Game G_9. Game G_9 is exactly the real experiment $\mathsf{Exp}^{\text{so-cca-real}}_{\mathsf{PKE},\mathcal{A},n,\mathcal{M},R}(\lambda)$. Thus

$$\Pr_9[G \Rightarrow 1] = \Pr\left[\mathsf{Exp}^{\text{so-cca-real}}_{\mathsf{PKE},\mathcal{A},n,\mathcal{M},R}(\lambda) = 1\right]. \tag{13}$$

Finally, Theorem 1 follows from (2, 3, 4, 5, 7, 8, 9, 10, 11, 12) and (13). ■

5 Instantiations

We give two instantiations of KEM with mPR-CCCA security and RER security.

5.1 KEM from MDDH

We present a KEM which is extracted from the multi-challenge IND-CCA secure PKE proposed by Gay *et al.* in [7]. The KEM $\mathsf{KEM}_{\mathsf{mddh}} = (\mathsf{KGen}, \mathsf{KEnc}, \mathsf{KDec})$ is shown in Figure 7.

Suppose $\mathcal{G} = (\mathbb{G}, q, P) \leftarrow_{\$} \mathsf{GGen}(1^\lambda)$ and $\mathcal{H}$ is a hash generator outputting functions $\mathsf{H} : \mathbb{G}^k \to \{0,1\}^\lambda$. For a vector $\mathbf{y} \in \mathbb{Z}_q^{3k}$, we use $\overline{\mathbf{y}} \in \mathbb{Z}_q^k$ to denote the upper k components and $\underline{\mathbf{y}} \in \mathbb{Z}_q^{2k}$ to denote the lower $2k$ components.

$\underline{\mathsf{KGen}(1^\lambda):}$	$\underline{\mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}}):}$	$\underline{\mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi):}$
$\mathbf{M} \leftarrow_{\$} \mathcal{U}_{3k,k}, \mathsf{H} \leftarrow_{\$} \mathcal{H}(1^\lambda).$	$\mathbf{r} \leftarrow_{\$} \mathbb{Z}_q^k, [\mathbf{y}] \leftarrow [\mathbf{M}]\mathbf{r}$	$\psi = [\mathbf{y}]$
$\mathbf{k}_{1,0}, \cdots, \mathbf{k}_{\lambda,1} \leftarrow_{\$} \mathbb{Z}_q^{3k}$	$\tau \leftarrow \mathsf{H}([\overline{\mathbf{y}}])$	$\tau \leftarrow \mathsf{H}([\overline{\mathbf{y}}])$
$\mathsf{pk}_{\mathsf{kem}} \leftarrow \begin{pmatrix} \mathcal{G}, \mathsf{H}, [\mathbf{M}] \\ ([\mathbf{M}^\top \mathbf{k}_{j,\beta}])_{1\le j\le\lambda}^{0\le\beta\le 1} \end{pmatrix}$	$\gamma \leftarrow \mathbf{r}^\top \cdot \sum_{j=1}^{\lambda} [\mathbf{M}^\top \mathbf{k}_{j,\tau_j}]$	$\mathbf{k}_\tau \leftarrow \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
$\mathsf{sk}_{\mathsf{kem}} \leftarrow (\mathbf{k}_{j,\beta})_{1\le j\le\lambda, 0\le\beta\le 1}$	Return $(\psi \leftarrow [\mathbf{y}], \gamma)$	$\gamma \leftarrow [\mathbf{y}^\top] \cdot \mathbf{k}_\tau$
Return $(\mathsf{pk}_{\mathsf{kem}}, \mathsf{sk}_{\mathsf{kem}})$	$//\Psi = \mathbb{G}^{3k}, \Gamma = \mathbb{G}$	Return γ

Fig. 7. The KEM $\mathsf{KEM}_{\mathsf{mddh}} = (\mathsf{KGen}, \mathsf{KEnc}, \mathsf{KDec})$ extracted from [7].

Perfectly correctness of $\mathsf{KEM}_{\mathsf{mddh}}$ is straightforward. See the full version [21] for the proofs of its tight mPR-CCCA security and tight RER security.

5.2 KEM from Qualified Proof System with Compact Public Key

First we recall the definition of a *proof system* described in [8].

Definition 7 (Proof System). *Let $\mathcal{L} = \{\mathcal{L}_{\mathsf{pars}}\}$ be a family of languages indexed by public parameters* pars*, with $\mathcal{L}_{\mathsf{pars}} \subseteq \mathcal{X}_{\mathsf{pars}}$ and an efficiently computable witness relation $\mathcal{R}$. A* proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ *for $\mathcal{L}$ consists of a tuple of PPT algorithms.*

- $\mathsf{PGen}(\mathsf{pars})$. *It outputs a public key* ppk *and a secret key* psk.
- $\mathsf{PPrv}(\mathsf{ppk}, x, w)$. *On input a statement* $x \in \mathcal{L}$ *and a witness* w *with* $\mathcal{R}(x, w) = 1$, *it deterministically outputs a proof* $\Pi \in \mathbf{\Pi}$ *and a key* $K \in \mathcal{K}$.
- $\mathsf{PVer}(\mathsf{ppk}, \mathsf{psk}, x, \Pi)$. *On input* $\mathsf{ppk}, \mathsf{psk}, x \in \mathcal{X}$ *and* Π, *it deterministically outputs* $b \in \{0, 1\}$ *together with a key* $K \in \mathcal{K}$ *if* $b = 1$ *or* $\perp$ *if* $b = 0$.
- $\mathsf{PSim}(\mathsf{ppk}, \mathsf{psk}, x)$. *Given* $\mathsf{ppk}, \mathsf{psk}, x \in \mathcal{X}$, *it deterministically outputs a proof* Π *and a key* $K \in \mathcal{K}$.

Next we recall the definition of a qualified proof system.

Definition 8 (Qualified Proof System [8]). *Let* $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ *be a proof system for a family of languages* $\mathcal{L} = \mathcal{L}_{\mathsf{pars}}$. *Let* $\mathcal{L}^{\mathrm{snd}} = \{\mathcal{L}^{\mathrm{snd}}_{\mathsf{pars}}\}$ *be a family of languages, such that* $\mathcal{L}_{\mathsf{pars}} \subseteq \mathcal{L}^{\mathrm{snd}}_{\mathsf{pars}}$. *We say that* PS *is* $\mathcal{L}^{\mathrm{snd}}$-qualified, *if the following properties hold.*

- **Completeness:** *For all possible public parameters* pars, *for all statements* $x \in \mathcal{L}$ *and all witnesses* w *such that* $\mathcal{R}(x, w) = 1$, $\Pr[\mathsf{PVer}(\mathsf{ppk}, \mathsf{psk}, x, \Pi)] = 1$, *where* $(\mathsf{ppk}, \mathsf{psk}) \leftarrow_{\$} \mathsf{PGen}(\mathsf{pars})$ *and* $(\Pi, K) \leftarrow_{\$} \mathsf{PPrv}(\mathsf{ppk}, x, w)$.
- **Perfect zero-knowledge:** *For all possible public parameters* pars, *all key pairs* $(\mathsf{ppk}, \mathsf{psk})$ *in the output range of* $\mathsf{PGen}(\mathsf{pars})$, *all statements* $x \in \mathcal{L}$ *and all witnesses* w *with* $\mathcal{R}(x, w) = 1$, *we have* $\mathsf{PPrv}(\mathsf{ppk}, x, w) = \mathsf{PSim}(\mathsf{ppk}, \mathsf{psk}, x)$.
- **Unique of the proofs:** *For all possible public parameters* pars, *all key pairs* $(\mathsf{ppk}, \mathsf{psk})$ *in the output range of* $\mathsf{PGen}(\mathsf{pars})$ *and all statements* $x \in \mathcal{X}$, *there exists at most one* Π^* *such that* $\mathsf{PVer}(\mathsf{ppk}, \mathsf{psk}, x, \Pi^*) = 1$.
- **Constrained $\mathcal{L}^{\mathrm{snd}}$-Soundness:** *For any stateful PPT adversary* $\mathcal{A}$, *consider the soundness experiment in Figure 8 (where* PSim *and* PVer *are implicitly assumed to have access to* ppk*).*

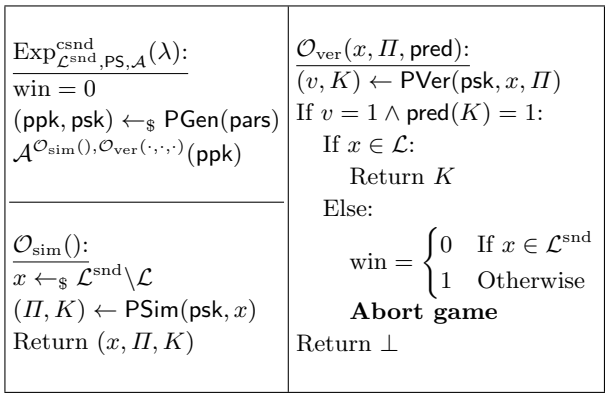

Fig. 8. Experiment used in the definition of constrained $\mathcal{L}^{\mathrm{snd}}$-soundness of PS.

Let Q_{ver} *be the total number of* $\mathcal{O}_{ver}$ *queries, which is independent of the environment without loss of generality. Let* $\mathsf{pred}_i : \mathcal{K} \cup \{\perp\} \to \{0, 1\}$ *be the predicate submitted by* $\mathcal{A}$ *in the* i*-th query, where* $\mathsf{pred}_i(\perp) = 0$ *for all* i. *The* uncertainty *of* $\mathcal{A}$ *is defined as*

$$uncert_{\mathcal{A}}(\lambda) := \frac{1}{Q_{ver}} \sum_{i=1}^{Q_{ver}} \Pr_{K \leftarrow_{\$} \mathcal{K}}[\mathsf{pred}_i(K) = 1].$$

We say constrained $\mathcal{L}^{\mathrm{snd}}$-soundness *holds for* PS *if for each PPT adversary* $\mathcal{A}$ *with negligible uncertainty,* $\mathsf{Adv}^{\mathrm{csnd}}_{\mathcal{L}^{\mathrm{snd}},\mathsf{PS},\mathcal{A}}(\lambda)$ *is negligible, where*

$$\mathsf{Adv}^{\mathrm{csnd}}_{\mathcal{L}^{\mathrm{snd}},\mathsf{PS},\mathcal{A}}(\lambda) := \Pr[\mathrm{win} = 1 \text{ in } \mathrm{Exp}^{\mathrm{csnd}}_{\mathcal{L}^{\mathrm{snd}},\mathsf{PS},\mathcal{A}}(\lambda)]$$

We omit the definition for $\mathcal{L}^{\mathrm{snd}}$-indistinguishability of two proof systems and the definition for $\widetilde{\mathcal{L}^{\mathrm{snd}}}$-extensibility of a proof system (See [8] and also our full version [21] for details). Here we define a new property for qualified proof system, which stresses that the simulated proof Π for a random $x \in \mathcal{L}^{\mathrm{snd}} \backslash \mathcal{L}$ is pseudorandom when providing verification oracle for only $x \in \mathcal{L}$.

Definition 9 (Pseudorandom Simulated Proof of Qualified Proof System). *Let* $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ *be a* $\mathcal{L}^{\mathrm{snd}}$*-qualified proof system for a family of languages* $\mathcal{L}$*. Let* $\mathcal{A}$ *be a stateful adversary and* $b \in \{0,1\}$ *be a bit. Define the following experiment* $\mathsf{Exp}^{\text{pr-proof-}b}_{\mathsf{PS},\mathcal{A}}(\lambda)$ *in Figure 9. We say* PS *has* pseudorandom simulated proof *if for each PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Exp}^{\text{pr-proof-0}}_{\mathsf{PS},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{pr-proof-1}}_{\mathsf{PS},\mathcal{A}}(\lambda) = 1\right] \right| \text{ is negl.}$$

$\mathrm{Exp}^{\text{pr-proof-}b}_{\mathsf{PS},\mathcal{A}}(\lambda)$://$b \in \{0,1\}$	$\mathcal{O}_{\mathrm{sim}}()$:	$\mathcal{O}_{\mathrm{ver}}(x, \Pi)$:
$(\mathsf{ppk}, \mathsf{psk}) \leftarrow_{\$} \mathsf{PGen}(\mathsf{pars})$	$x \leftarrow_{\$} \mathcal{L}^{\mathrm{snd}} \backslash \mathcal{L}$	$(v, K) \leftarrow \mathsf{PVer}(\mathsf{psk}, x, \Pi)$
$b' \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\mathrm{sim}}(), \mathcal{O}_{\mathrm{ver}}(\cdot,\cdot)}(\mathsf{ppk})$	$\Pi_0 \leftarrow_{\$} \boldsymbol{\Pi}$	If $x \notin \mathcal{L} \vee v = 0$:
Return b'	$(\Pi_1, K) \leftarrow \mathsf{PSim}(\mathsf{psk}, x)$	Return $\perp$
	Return (x, Π_b)	Return K

Fig. 9. Experiment used in the definition of pseudorandom simulated proof of PS.

The Qualified Proof System in [8]. First we explain how the public parameters pars are sampled. Fix some $k \in \mathbb{N}$, invoke $\mathcal{G} \leftarrow_{\$} \mathsf{GGen}(1^{\lambda})$ where $\mathcal{G} = (\mathbb{G}, q, P)$. Let $\mathcal{D}_{2k,k}$ be a fixed matrix distribution, we sample $\mathbf{A} \leftarrow_{\$} \mathcal{D}_{2k,k}$ and $\mathbf{A}_0 \leftarrow_{\$} \mathcal{U}_{2k,k}$ where $\overline{\mathbf{A}}$ and $\overline{\mathbf{A}}_0$ are both full rank. Additionally select $\mathbf{A}_1 \in \mathbb{Z}_q^{2k \times k}$ according to $\mathcal{U}_{2k,k}$ with the restriction $\overline{\mathbf{A}}_0 = \overline{\mathbf{A}}_1$. Let $\mathcal{H}_0$ and $\mathcal{H}_1$ be universal hash function generators returning functions $\mathsf{h}_0 : \mathbb{G}^{k^2+1} \to \mathbb{Z}_q^{k \times k}$ and $\mathsf{h}_1 : \mathbb{G}^{k+1} \to \mathbb{Z}_q^k$ respectively. Let $\mathsf{h}_0 \leftarrow_{\$} \mathcal{H}_0$ and $\mathsf{h}_1 \leftarrow_{\$} \mathcal{H}_1$. Let $\mathsf{pars} \leftarrow (k, \mathcal{G}, [\mathbf{A}], [\mathbf{A}_0], [\mathbf{A}_1], \mathsf{h}_0, \mathsf{h}_1)$ be the public parameters and we assume pars is an implicit input of all algorithms. The languages are defined as $\mathcal{L} := \mathsf{span}([\mathbf{A}])$, $\mathcal{L}^{\mathrm{snd}} := \mathsf{span}([\mathbf{A}]) \cup \mathsf{span}([\mathbf{A}_0])$ and $\widetilde{\mathcal{L}^{\mathrm{snd}}} := \mathsf{span}([\mathbf{A}]) \cup \mathsf{span}([\mathbf{A}_0]) \cup \mathsf{span}([\mathbf{A}_1])$.

The construction[8] of $\mathcal{L}^{\mathrm{snd}}$-qualified proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ in [8] is shown in Figure 10.

[8] This construction in Figure 10 is an updated version of [8] from a personal communication.

According to Theorem 1 of [8], PS is $\mathcal{L}^{\mathrm{snd}}$-qualified and $\widetilde{\mathcal{L}^{\mathrm{snd}}}$-extensible, both admitting tight security reductions to the MDDH assumption. More precisely, $\mathsf{Adv}^{\mathrm{csnd}}_{\mathcal{L}^{\mathrm{snd}},\mathsf{PS},\mathcal{A}}(\lambda), \mathsf{Adv}^{\mathrm{csnd}}_{\widetilde{\mathcal{L}^{\mathrm{snd}}},\widetilde{\mathsf{PS}},\mathcal{A}}(\lambda) \leq 2k \cdot \mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_{2k,k},\mathsf{GGen},\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$, $\mathsf{Adv}^{\text{PS-ind}}_{\mathcal{L}^{\mathrm{snd}}} \leq 2^{-\Omega(\lambda)}$.

$\underline{\mathsf{PGen}(\mathsf{pars})}$:
$\mathbf{K}_{\mathbf{X}} \leftarrow_{\$} \mathbb{Z}_q^{(k^2+1)\times 2k}$
$\mathbf{K}_{\mathbf{y}} \leftarrow_{\$} \mathbb{Z}_q^{(k+1)\times 2k}$
$[\mathbf{P}_{\mathbf{X}}] \leftarrow \mathbf{K}_{\mathbf{X}}[\mathbf{A}] \in \mathbb{G}^{(k^2+1)\times k}$
$[\mathbf{P}_{\mathbf{y}}] \leftarrow \mathbf{K}_{\mathbf{y}}[\mathbf{A}] \in \mathbb{G}^{(k+1)\times k}$
$\mathsf{ppk} \leftarrow ([\mathbf{P}_{\mathbf{X}}], [\mathbf{P}_{\mathbf{y}}])$
$\mathsf{psk} \leftarrow (\mathbf{K}_{\mathbf{X}}, \mathbf{K}_{\mathbf{y}})$
Return $(\mathsf{ppk}, \mathsf{psk})$

$\underline{\mathsf{PSim}(\mathsf{ppk}, \mathsf{psk}, [\mathbf{c}])}$:
$\mathbf{X} \leftarrow \mathsf{h}_0(\mathbf{K}_{\mathbf{X}}[\mathbf{c}])$
$\mathbf{y} \leftarrow \mathsf{h}_1(\mathbf{K}_{\mathbf{y}}[\mathbf{c}])$
$[\pi] \leftarrow [\overline{\mathbf{A}_0}] \cdot \mathbf{X} + [\overline{\mathbf{c}}] \cdot \mathbf{y}^\top$
$[\mathbf{K}] \leftarrow [\underline{\mathbf{A}_0}] \cdot \mathbf{X} + [\underline{\mathbf{c}}] \cdot \mathbf{y}^\top$
$[\kappa] \leftarrow \mathsf{trace}([\mathbf{K}])$
Return $([\pi], [\kappa])$

$\underline{\mathsf{PPrv}(\mathsf{ppk}, [\mathbf{c}], \mathbf{r})}$:
$\mathbf{X} \leftarrow \mathsf{h}_0([\mathbf{P}_{\mathbf{X}}]\mathbf{r}) \in \mathbb{Z}_q^{k\times k}$
$\mathbf{y} \leftarrow \mathsf{h}_1([\mathbf{P}_{\mathbf{y}}]\mathbf{r}) \in \mathbb{Z}_q^{k}$
$[\pi] \leftarrow [\overline{\mathbf{A}_0}] \cdot \mathbf{X} + [\overline{\mathbf{c}}] \cdot \mathbf{y}^\top \in \mathbb{G}^{k\times k}$
$[\mathbf{K}] \leftarrow [\underline{\mathbf{A}_0}] \cdot \mathbf{X} + [\underline{\mathbf{c}}] \cdot \mathbf{y}^\top \in \mathbb{G}^{k\times k}$
$[\kappa] \leftarrow \mathsf{trace}([\mathbf{K}]) \in \mathbb{G}$
Return $([\pi], [\kappa])$

$\underline{\mathsf{PVer}(\mathsf{ppk}, \mathsf{psk}, [\mathbf{c}], [\pi^*])}$:
$([\pi], [\kappa]) \leftarrow \mathsf{PSim}(\mathsf{ppk}, \mathsf{psk}, [\mathbf{c}])$
Return $\begin{cases} (1, [\kappa]) & \text{If } [\pi] = [\pi^*] \\ (0, \bot) & \text{Otherwise} \end{cases}$

Fig. 10. Construction of the $\mathcal{L}^{\mathrm{snd}}$-qualified proof system $\mathsf{PS} = (\mathsf{PGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ in [8].

We now prove that PS has pseudorandom simulated proof with Theorem 2.

Theorem 2. *The $\mathcal{L}^{\mathrm{snd}}$-qualified proof system PS in Figure 10 has pseudorandom simulated proof if $\mathcal{U}_k$-MDDH assumption holds. Specifically, for each PPT adversary $\mathcal{A}$, we can build a PPT adversary $\mathcal{B}$ with $\mathbf{T}(\mathcal{B}) \leq \mathbf{T}(\mathcal{A}) + (Q_{\mathrm{sim}} + Q_{\mathrm{ver}}) \cdot \mathsf{poly}(\lambda)$ such that the advantage*

$$\mathsf{Adv}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen},\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

where $Q_{\mathrm{sim}}(Q_{\mathrm{ver}})$ is the total number of $\mathcal{O}_{\mathrm{sim}}(\mathcal{O}_{\mathrm{ver}})$ queries made by $\mathcal{A}$ and $\mathsf{poly}(\lambda)$ is a polynomial independent of $\mathbf{T}(\mathcal{A})$.

Proof of Theorem 2.

For a fixed PPT adversary $\mathcal{A}$, consider an experiment $\mathsf{Exp}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda)$ which first uniformly selects $b \leftarrow_{\$} \{0,1\}$, then calls $\mathsf{Exp}^{\text{pr-proof-}b}_{\mathsf{PS},\mathcal{A}}(\lambda)$ and gets its output b'. It is straightforward that

$$\mathsf{Adv}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda) = 2\left|\Pr[b' = b \text{ in } \mathsf{Exp}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda)] - \tfrac{1}{2}\right|.$$

Now we rewrite $\mathsf{Exp}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda)$ in Figure 11 and make changes to it gradually through game G_0 to G_3. Games $G_0 - G_3$ are defined as follows.

$\mathrm{Exp}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda)$:$G_0$ $\boxed{G_1 - G_3}$	$\mathcal{O}_{\text{sim}}()$: G_0 $\boxed{G_1}$ $\boxed{G_2}$ $\boxed{G_3}$	$\mathcal{O}_{\text{ver}}([\mathbf{c}], \Pi^*)$: $G_0 - G_3$
$b \leftarrow_\$ \{0,1\}$	$\mathbf{r} \leftarrow_\$ \mathbb{Z}_q^k, [\mathbf{c}] \leftarrow [\mathbf{A}_0]\mathbf{r}$	$\mathbf{X} \leftarrow \mathsf{h}_0(\mathbf{K}_\mathbf{X}[\mathbf{c}])$
$\boxed{\mathbf{V} \leftarrow_\$ \mathbb{Z}_q^{(k^2+1)\times k}}$	$\Pi_0 \leftarrow_\$ \mathbb{G}^{k\times k}$	$\mathbf{y} \leftarrow \mathsf{h}_1(\mathbf{K}_\mathbf{y}[\mathbf{c}])$
$\mathbf{K}_\mathbf{X} \leftarrow_\$ \mathbb{Z}_q^{(k^2+1)\times 2k}$	$\mathbf{X} \leftarrow \mathsf{h}_0(\mathbf{K}_\mathbf{X}[\mathbf{c}])$	$\Pi \leftarrow [\overline{\mathbf{A}_0}] \cdot \mathbf{X} + [\overline{\mathbf{c}}] \cdot \mathbf{y}^\top$
$\mathbf{K}_\mathbf{y} \leftarrow_\$ \mathbb{Z}_q^{(k+1)\times 2k}$	$\boxed{\mathbf{X} \leftarrow \mathsf{h}_0([\mathbf{V}\mathbf{r}])}$	$[\mathbf{K}] \leftarrow [\underline{\mathbf{A}_0}] \cdot \mathbf{X} + [\underline{\mathbf{c}}] \cdot \mathbf{y}^\top$
$[\mathbf{P}_\mathbf{X}] \leftarrow \mathbf{K}_\mathbf{X}[\mathbf{A}]$	$\boxed{\mathbf{X} \leftarrow_\$ \mathbb{Z}_q^{k\times k}}$	$[\kappa] \leftarrow \mathsf{trace}([\mathbf{K}])$
$[\mathbf{P}_\mathbf{y}] \leftarrow \mathbf{K}_\mathbf{y}[\mathbf{A}]$	$\mathbf{y} \leftarrow \mathsf{h}_1(\mathbf{K}_\mathbf{y}[\mathbf{c}])$	If $\begin{bmatrix}[\mathbf{c}] \notin \mathsf{span}([\mathbf{A}]) \\ \vee \Pi \neq \Pi^*\end{bmatrix}$:
$\mathsf{ppk} \leftarrow ([\mathbf{P}_\mathbf{X}], [\mathbf{P}_\mathbf{y}])$	$\Pi_1 \leftarrow [\overline{\mathbf{A}_0}] \cdot \mathbf{X} + [\overline{\mathbf{c}}] \cdot \mathbf{y}^\top$	Return $\perp$
$b' \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\text{sim}}(),\mathcal{O}_{\text{ver}}(\cdot,\cdot)}(\mathsf{ppk})$	$\Pi_1 \leftarrow_\$ \mathbb{G}^{k\times k}$	Return $[\kappa]$
Return b'	Return $([\mathbf{c}], \Pi_b)$	

Fig. 11. Games $G_0 - G_3$ in the proof of Theorem 2.

Game G_0. This game is the same as $\mathsf{Exp}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda)$. Then

$$\mathsf{Adv}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{A}}(\lambda) = 2\left|\Pr_0[b' = b] - \frac{1}{2}\right|. \tag{14}$$

Game $G_0 - G_1$. G_1 is almost the same as G_0 except for the $\mathcal{O}_{\text{sim}}$ oracle.

- In G_0, $\mathbf{X} = \mathsf{h}_0(\mathbf{K}_\mathbf{X}[\mathbf{c}])$, where $[\mathbf{c}] = [\mathbf{A}_0]\mathbf{r}$ and $\mathbf{r} \leftarrow_\$ \mathbb{Z}_q^k$ for each $\mathcal{O}_{\text{sim}}$ query.
- In G_1, $\mathbf{X} = \mathsf{h}_0([\mathbf{V}\mathbf{r}])$, where (i) a fresh $\mathbf{r}$ is uniformly chosen from $\mathbb{Z}_q^k$ for each $\mathcal{O}_{\text{sim}}$ query; (ii) $\mathbf{V}$ is uniformly chosen from $\mathbb{Z}_q^{(k^2+1)\times k}$ beforehand but will be fixed for each $\mathcal{O}_{\text{sim}}$ query.

Define $\mathbf{U} := \mathbf{K}_\mathbf{X}\mathbf{A}_0$, so $(\mathbf{P}_\mathbf{X}|\mathbf{U}) = \mathbf{K}_\mathbf{X}(\mathbf{A}|\mathbf{A}_0)$. Note that, the square matrix $(\mathbf{A}|\mathbf{A}_0)$ is of full rank with probability $1 - 2^{-\Omega(\lambda)}$, then the entropy of $\mathbf{K}_\mathbf{X}$ is transferred to $(\mathbf{P}_\mathbf{X}|\mathbf{U})$ intactly. Recall that $\mathbf{K}_\mathbf{X}$ is uniform over $\mathbb{Z}_q^{(k^2+1)\times 2k}$. Therefore, $(\mathbf{P}_\mathbf{X}|\mathbf{U})$ is uniform over $\mathbb{Z}_q^{(k^2+1)\times 2k}$ as well. Consequently, $\mathbf{U}$ is uniformly distributed over $\mathbb{Z}_q^{(k^2+1)\times k}$ even conditioned on $\mathbf{P}_\mathbf{X}$.

In G_0, the $\mathcal{O}_{\text{ver}}$ oracle rejects all $[\mathbf{c}] \notin [\mathsf{span}(\mathbf{A})]$. Therefore, the information of $\mathbf{K}_\mathbf{X}$ leaked through $\mathcal{O}_{\text{ver}}$ is characterized by the public key $\mathbf{P}_\mathbf{X}$. Together with the fact that $[\mathbf{c}] = [\mathbf{A}_0]\mathbf{r}$ in $\mathcal{O}_{\text{sim}}$ of G_0 and G_1, the computation of $\mathbf{K}_\mathbf{X}[\mathbf{c}] = [\mathbf{K}_\mathbf{X}\mathbf{A}_0]\mathbf{r}$ in $\mathcal{O}_{\text{sim}}$ of G_0 can be replaced with $[\mathbf{V}]\mathbf{r}$ for $\mathbf{V} \leftarrow_\$ \mathbb{Z}_q^{(k^2+1)\times k}$ in G_1. Thus we have

$$|\Pr_0[b' = b] - \Pr_1[b' = b]| \leq 2^{-\Omega(\lambda)}. \tag{15}$$

Game $G_1 - G_2$. G_2 is the same as G_1 except for the $\mathcal{O}_{\text{sim}}$ oracle.

- In G_1, $\mathbf{X} = \mathsf{h}_0([\mathbf{V}\mathbf{r}])$ is computed with the same $\mathbf{V}$ but a fresh $\mathbf{r} \leftarrow_\$ \mathbb{Z}_q^k$.
- In G_2, $\mathbf{X}$ is uniformly selected from $\mathbb{Z}_q^{k\times k}$ for each $\mathcal{O}_{\text{sim}}$ oracle.

We will show that

$$|\Pr_1[b'=b]-\Pr_2[b'=b]| \le \mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen},\mathcal{B}}(\lambda)+2^{-\Omega(\lambda)}. \tag{16}$$

To prove (16), we define two intermediate games G_1' and G_1''.
G_1' is the same as G_1 except for the generation of $\mathbf{r}$ in $\mathcal{O}_{\mathrm{sim}}$. For each $\mathcal{O}_{\mathrm{sim}}$ query,
– in G_1, $\mathbf{r} \leftarrow_{\$} \mathbb{Z}_q^k$;
– in G_1', $\mathbf{r} \leftarrow \mathbf{W}\mathbf{s}$ with a fresh $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_q^k$ but the same $\mathbf{W}$, which is uniformly selected from $\mathbb{Z}_q^{k\times k}$ beforehand.

Since $\mathbf{W}$ is invertible with probability $1-2^{-\Omega(\lambda)}$, we have that

$$|\Pr_1[b'=b]-\Pr_{1'}[b'=b]| \le 2^{-\Omega(\lambda)}. \tag{17}$$

G_1'' is the same with G_1' except for the $\mathcal{O}_{\mathrm{sim}}$ oracle. For each $\mathcal{O}_{\mathrm{sim}}$ query,
– G_1' sets $[\mathbf{c}] \leftarrow \mathbf{A}_0[\mathbf{W}]\mathbf{s}$ and $\mathbf{X} \leftarrow \mathsf{h}_0([\mathbf{V}\mathbf{W}]\mathbf{s})$, where $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_q^k$;
– G_1'' sets $[\mathbf{c}] \leftarrow \mathbf{A}_0[\mathbf{r}]$ and $\mathbf{X} \leftarrow \mathsf{h}_0([\mathbf{u}])$, where $\mathbf{r} \leftarrow_{\$} \mathbb{Z}_q^k, \mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^{k^2+1}$.

Note that, with overwhelming probability, $[\mathbf{B}] = \left[\begin{smallmatrix}\mathbf{W}\\ \mathbf{V}\mathbf{W}\end{smallmatrix}\right]$ distributes uniformly over $\mathbb{G}^{(k^2+k+1)\times k}$. Then we can build an adversary $\mathcal{B}$ and show that

$$|\Pr_{1'}[b'=b]-\Pr_{1''}[b'=b]| \le \mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen},\mathcal{B}}(\lambda)+2^{-\Omega(\lambda)}. \tag{18}$$

To prove (18), we construct an adversary $\mathcal{B}'$ and show that

$$|\Pr_{1'}[b'=b]-\Pr_{1''}[b'=b]| \le \mathsf{Adv}^{Q_{\mathrm{sim}}\text{-mddh}}_{\mathcal{U}_{k^2+k+1,k},\mathsf{GGen},\mathcal{B}'}(\lambda). \tag{19}$$

Upon receiving a challenge $(\mathcal{G}, [\mathbf{B}] \in \mathbb{G}^{(k^2+k+1)\times k}, [\mathbf{H}] := ([\mathbf{h}_1|\cdots|\mathbf{h}_{Q_{\mathrm{sim}}}]) \in \mathbb{G}^{(k^2+k+1)\times Q_{\mathrm{sim}}})$ for the Q_{sim}-fold $\mathcal{U}_{k^2+k+1,k}$-MDDH problem, $\mathcal{B}'$ simulates game $G_1'(G_1'')$. In the simulation of the i-th $\mathcal{O}_{\mathrm{sim}}$ oracle query for $i \in [Q_{\mathrm{sim}}]$, $\mathcal{B}'$ embeds $[\overline{\mathbf{h}_i}]$ in $[\mathbf{c}]$ with $[\mathbf{c}] \leftarrow \mathbf{A}_0[\overline{\mathbf{h}_i}]$. Then $\mathcal{B}'$ embeds $[\underline{\mathbf{h}_i}]$ in $\mathbf{X}$ with $\mathbf{X} \leftarrow \mathsf{h}_0([\underline{\mathbf{h}_i}])$.

If $[\mathbf{h}_i]$ is uniformly chosen from $\mathsf{span}([\mathbf{B}])$ for all $i \in [Q_{\mathrm{sim}}]$, then $[\mathbf{h}_i] = \left[\begin{smallmatrix}\mathbf{W}\\ \mathbf{V}\mathbf{W}\end{smallmatrix}\right]\mathbf{s}_i$, $[\overline{\mathbf{h}_i}] = [\mathbf{W}]\mathbf{s}_i$ and $[\underline{\mathbf{h}_i}] = [\mathbf{V}\mathbf{W}]\mathbf{s}_i$ with $\mathbf{s}_i \leftarrow_{\$} \mathbb{Z}_q^k$. In this case, $\mathcal{B}'$ perfectly simulates G_1'. If $[\mathbf{h}_i]$ is uniformly chosen from $\mathbb{G}^{k^2+k+1}$ for all $i \in [Q_{\mathrm{sim}}]$, then both $[\overline{\mathbf{h}_i}]$ and $[\underline{\mathbf{h}_i}]$ are uniform. In this case, $\mathcal{B}'$ perfectly simulates G_1''.

From above, (19) follows. Then, (18) follows from (19) and the random self-reducibility property of the MDDH problem.

In G_1'', $\mathbf{X} \leftarrow \mathsf{h}_0([\mathbf{u}])$ for a uniform $\mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^{k^2+1}$. Since h_0 is universal, by leftover hash lemma and a union bound, we have that

$$|\Pr_{1''}[b'=b]-\Pr_2[b'=b]| \le \frac{Q_{\mathrm{sim}}}{2\sqrt{q}} = 2^{-\Omega(\lambda)}. \tag{20}$$

Then (16) follows from (17, 18) and (20).

Game $G_2 - G_3$. G_3 is the same as G_2 except for the $\mathcal{O}_{\mathrm{sim}}$ oracle.
For each $\mathcal{O}_{\mathrm{sim}}$ query,
• in G_2, $\Pi_1 = [\overline{\mathbf{A}_0}] \cdot \mathbf{X} + [\overline{\mathbf{c}}] \cdot \mathbf{y}^\top$ for $[\mathbf{c}] = [\mathbf{A}_0]\mathbf{r}$ and a fresh $\mathbf{X} \leftarrow_{\$} \mathbb{Z}_q^{k\times k}$;

• in G_3, Π_1 is uniformly selected from $\mathbb{G}^{k\times k}$.

Note that in G_2,

$$\Pi_1 = [\overline{\mathbf{A}_0}] \cdot \mathbf{X} + [\overline{\mathbf{c}}] \cdot \mathbf{y}^\top = [\overline{\mathbf{A}_0}](\mathbf{X} + \mathbf{r} \cdot \mathbf{y}^\top).$$

Due to the uniformness of $\mathbf{X}$, Π_1 has the same distribution as $[\overline{\mathbf{A}_0}]\mathbf{X}$. Since $\overline{\mathbf{A}_0}$ is an invertible matrix, $[\overline{\mathbf{A}_0}]\mathbf{X}$ is uniformly distributed over $\mathbb{G}^{k\times k}$. Thus we have

$$\Pr_2[b' = b] = \Pr_3[b' = b]. \tag{21}$$

Game G_3. In G_3, Π_0 distributes identically to Π_1 and

$$\Pr_3[b' = b] = \frac{1}{2}. \tag{22}$$

Finally, Theorem 2 follows from (14, 15, 16, 21) and (22). ■

$(\mathsf{pk}_{\mathsf{kem}}, \mathsf{sk}_{\mathsf{kem}}) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)$:
$(\mathsf{ppk}, \mathsf{psk}) \leftarrow_{\$} \mathsf{PGen}(\mathsf{pars})$
$\mathbf{k}_0, \mathbf{k}_1 \leftarrow_{\$} \mathbb{Z}_q^{2k}, \quad [\mathbf{p}_0^\top] \leftarrow \mathbf{k}_0^\top[\mathbf{A}] \in \mathbb{G}^{1\times k}, \quad [\mathbf{p}_1^\top] \leftarrow \mathbf{k}_1^\top[\mathbf{A}] \in \mathbb{G}^{1\times k}$
Return $\mathsf{pk}_{\mathsf{kem}} \leftarrow (\mathsf{ppk}, [\mathbf{p}_0^\top], [\mathbf{p}_1^\top]), \quad \mathsf{sk}_{\mathsf{kem}} \leftarrow (\mathsf{psk}, \mathbf{k}_0, \mathbf{k}_1)$

$(\psi, \gamma) \leftarrow_{\$} \mathsf{KEnc}(\mathsf{pk}_{\mathsf{kem}})$:
$\mathbf{r} \leftarrow_{\$} \mathbb{Z}_q^k, \quad [\mathbf{c}] \leftarrow [\mathbf{A}]\mathbf{r} \in \mathbb{G}^{2k}$
$(\Pi, [\kappa]) \leftarrow_{\$} \mathsf{PPrv}(\mathsf{ppk}, [\mathbf{c}], \mathbf{r})$
$\tau \leftarrow \mathsf{H}([\overline{\mathbf{c}}]) \in \{0,1\}^\lambda \subseteq \mathbb{Z}_q$
$\gamma \leftarrow ([\mathbf{p}_0^\top] + \tau[\mathbf{p}_1^\top]) \cdot \mathbf{r} + [\kappa] \in \mathbb{G}$
Return $(\psi \leftarrow ([\mathbf{c}], \Pi), \gamma)$
$//\Psi = \mathbb{G}^{2k} \times \mathbb{G}^{k\times k}, \quad \Gamma = \mathbb{G}$

$\gamma/\perp \leftarrow \mathsf{KDec}(\mathsf{sk}_{\mathsf{kem}}, \psi)$:
Parse $\psi = ([\mathbf{c}], \Pi)$
$(v \in \{0,1\}, [\kappa]) \leftarrow \mathsf{PVer}(\mathsf{psk}, [\mathbf{c}], \Pi)$
$\tau \leftarrow \mathsf{H}([\overline{\mathbf{c}}]) \in \{0,1\}^\lambda \subseteq \mathbb{Z}_q$
$\gamma \leftarrow (\mathbf{k}_0^\top + \tau\mathbf{k}_1^\top) \cdot [\mathbf{c}] + [\kappa] \in \mathbb{G}$
Return $\begin{cases} \gamma & \text{If } v = 1 \\ \perp & \text{Otherwise} \end{cases}$

Fig. 12. Construction of $\mathsf{KEM}_{\mathsf{qps}} = (\mathsf{KGen}, \mathsf{KEnc}, \mathsf{KDec})$ in [8]

KEM from Qualified Proof System. The construction of the qualified PS based KEM $\mathsf{KEM}_{\mathsf{qps}} = (\mathsf{KGen}, \mathsf{KEnc}, \mathsf{KDec})$ from [8] is shown in Figure 12. Suppose $\mathcal{H}$ is a hash generator outputting functions $\mathsf{H} : \mathbb{G}^k \to \{0,1\}^\lambda$. The parameters pars used in this construction are specified in Section 5.2.

Theorem 2 in [8] has shown that $\mathsf{KEM}_{\mathsf{qps}}$ is IND-CCCA secure. Now we prove that $\mathsf{KEM}_{\mathsf{qps}}$ is mPR-CCCA secure (through Theorem 3) and is RER secure (through Theorem 4), both admitting tight security reductions.

Theorem 3. *The KEM $\mathsf{KEM}_{\mathsf{qps}}$ in Figure 12 is mPR-CCCA secure if the $\mathcal{D}_{2k,k}$-MDDH assumption holds, $\mathcal{H}$ outputs collision-resistant hash function, PS is $\mathcal{L}^{snd}$-qualified, $\widetilde{\mathcal{L}^{snd}}$-extensible and has pseudorandom simulated proof. Specifically, for*

each PPT adversary $\mathcal{A}$ with negligible uncertainty $uncert_{\mathcal{A}}(\lambda)$, we can build PPT adversaries $\mathcal{B}_1, \cdots, \mathcal{B}_7$ with $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_7) \leq \mathbf{T}(\mathcal{A}) + (Q_{\mathrm{enc}} + Q_{\mathrm{dec}}) \cdot \mathsf{poly}(\lambda)$ and $uncert_{\mathcal{B}_4}(\lambda) = uncert_{\mathcal{B}_6}(\lambda) = uncert_{\mathcal{A}}(\lambda)$, such that the advantage

$$\begin{aligned}\mathsf{Adv}^{\text{mpr-ccca}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda) \leq\ & 2\mathsf{Adv}^{\mathrm{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + (4\lambda + 3k)\mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_{2k,k},\mathsf{GGen},\mathcal{B}_2}(\lambda) \\ & + 7\mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen},\mathcal{B}_3}(\lambda) + \mathsf{Adv}^{\mathrm{csnd}}_{\mathcal{L}^{\mathrm{snd}},\mathsf{PS},\mathcal{B}_4}(\lambda) + \mathsf{Adv}^{\text{PS-ind}}_{\mathcal{L}^{\mathrm{snd}},\mathsf{PS},\widetilde{\mathsf{PS}},\mathcal{B}_5}(\lambda) \\ & + \lambda\mathsf{Adv}^{\mathrm{csnd}}_{\widetilde{\mathcal{L}^{\mathrm{snd}}},\widetilde{\mathsf{PS}},\mathcal{B}_6}(\lambda) + 2\mathsf{Adv}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{B}_7}(\lambda) \\ & + ((\lambda+2)\cdot Q_{\mathrm{enc}} + 3)\cdot Q_{\mathrm{dec}} \cdot uncert_{\mathcal{A}}(\lambda) + 2^{-\Omega(\lambda)}.\end{aligned}$$

where $Q_{\mathrm{enc}}(Q_{\mathrm{dec}})$ is the total number of $\mathcal{O}_{\mathrm{enc}}(\mathcal{O}_{\mathrm{dec}})$ queries made by $\mathcal{A}$ and $\mathsf{poly}(\lambda)$ is a polynomial independent of $\mathbf{T}(\mathcal{A})$.

Proof of Theorem 3. For a fixed PPT adversary $\mathcal{A}$ with negligible uncertainty $uncert_{\mathcal{A}}(\lambda)$, consider an experiment $\mathsf{Exp}^{\text{mpr-ccca}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda)$ which first randomly selects $b \leftarrow_{\$} \{0,1\}$, then calls $\mathsf{Exp}^{\text{mpr-ccca-}b}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda)$ and gets its output b'. It is straightforward that $\mathsf{Adv}^{\text{mpr-ccca}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda) = 2\left|\Pr[b' = b \text{ in } \mathsf{Exp}^{\text{mpr-ccca}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda)] - \frac{1}{2}\right|$. Then we rewrite experiment $\mathsf{Exp}^{\text{mpr-ccca}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda)$ in Figure 13 and make changes to it gradually through game G_0 to G_9 which are defined as follows.

Game G_0. This game is identical to $\mathsf{Exp}^{\text{mpr-ccca}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda)$. Then

$$\mathsf{Adv}^{\text{mpr-ccca}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda) = 2\left|\Pr_0[b' = b] - \frac{1}{2}\right|. \tag{23}$$

Game $G_0 - G_1$. G_1 is the same as G_0 except that an additional rejection rule is added in $\mathcal{O}_{\mathrm{dec}}$. More precisely, in G_1, we use a set $\mathcal{T}$ to log all the tags $\tau_b = \mathsf{H}([\overline{\mathbf{c}_b}])$ used in oracle $\mathcal{O}_{\mathrm{enc}}$, and any $\mathcal{O}_{\mathrm{dec}}(\mathsf{pred}, \psi = ([\mathbf{c}], \Pi))$ query will be rejected if $\tau = \mathsf{H}([\overline{\mathbf{c}}]) \in \mathcal{T}$.

Lemma 1.

$$\begin{aligned}|\Pr_0[b' = b] - \Pr_1[b' = b]| \leq\ & \mathsf{Adv}^{\mathrm{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + \frac{k}{2}\cdot\mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_{2k,k},\mathsf{GGen},\mathcal{B}_2}(\lambda) \\ & + \frac{1}{2}\mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen},\mathcal{B}_3}(\lambda) + \frac{3}{2}Q_{\mathrm{dec}}\cdot uncert_{\mathcal{A}}(\lambda) + 2^{-\Omega(\lambda)}.\end{aligned}$$

We refer to the full version [21] for the proof of this lemma.

Game $G_1 - G_2$. G_2 is almost the same as G_1 except for two changes in $\mathcal{O}_{\mathrm{enc}}$. The first change is that PPrv is replaced with PSim. The second change is that $\mathsf{sk}_{\mathsf{KEM}}$ is used to calculate γ_1. More precisely, for $[\mathbf{c}_1] = [\mathbf{A}]\mathbf{r}_1$ in oracle $\mathcal{O}_{\mathrm{enc}}$,

- in G_1, $(\Pi_1, [\kappa_1]) \leftarrow \mathsf{PPrv}(\mathsf{ppk}, [\mathbf{c}_1], \mathbf{r}_1)$, $\gamma_1 \leftarrow ([\mathbf{p}_0^\top] + \tau_1[\mathbf{p}_1^\top])\cdot\mathbf{r}_1 + [\kappa_1]$;
- in G_2, $(\Pi_1, [\kappa_1]) \leftarrow \mathsf{PSim}(\mathsf{psk}, [\mathbf{c}_1])$, $\gamma_1 \leftarrow (\mathbf{k}_0^\top + \tau_1\mathbf{k}_1^\top)\cdot[\mathbf{c}_1] + [\kappa_1]$.

Due to the perfect zero-knowledge property of PS, we have $\mathsf{PPrv}(\mathsf{ppk}, [\mathbf{c}_1], \mathbf{r}_1) = \mathsf{PSim}(\mathsf{psk}, [\mathbf{c}_1])$. Meanwhile, $[\mathbf{p}_0^\top] = \mathbf{k}_0^\top[\mathbf{A}]$ and $[\mathbf{p}_1^\top] = \mathbf{k}_1^\top[\mathbf{A}]$, so we have $([\mathbf{p}_0^\top] + \tau_1[\mathbf{p}_1^\top])\cdot\mathbf{r}_1 + [\kappa_1] = (\mathbf{k}_0^\top + \tau_1\mathbf{k}_1^\top)\cdot[\mathbf{c}_1] + [\kappa_1]$.

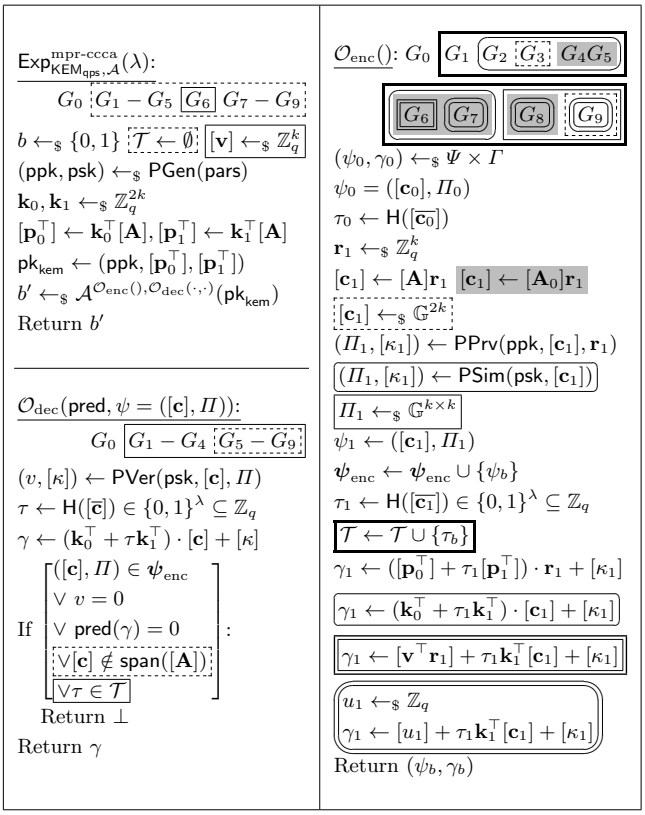

Fig. 13. Game $G_0 - G_9$ in the proof of Theorem 3.

These changes are only conceptual, so G_1 is identical to G_2 and

$$\Pr_1[b' = b] = \Pr_2[b' = b]. \tag{24}$$

Game $G_2 - G_3$. G_3 is the same as G_2 except for one difference in $\mathcal{O}_{\text{enc}}$.

- In game G_2, $[\mathbf{c}_1]$ is uniform over $\mathsf{span}([\mathbf{A}])$ for each $\mathcal{O}_{\text{enc}}$ query.
- In game G_3, $[\mathbf{c}_1]$ is uniform over $\mathbb{G}^{2k}$ for each $\mathcal{O}_{\text{enc}}$ query.

We can build an adversary $\mathcal{B}_2$ and show that

$$|\Pr_2[b' = b] - \Pr_3[b' = b]| \leq k \cdot \mathsf{Adv}^{\text{mddh}}_{\mathcal{D}_{2k,k},\mathsf{GGen},\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)}. \tag{25}$$

The reduction is straightforward, since $\mathcal{B}_2$ can simulate $G_2(G_3)$ by generating the secret key itself and embed its own challenge in $[\mathbf{c}_1]$. We omit the details.

We refer to the full version [21] for the proof of this lemma.

Game $G_3 - G_4$. G_4 is the same as G_3 except for one difference in $\mathcal{O}_{\text{enc}}$.

- In game G_3, $[\mathbf{c}_1]$ is uniform over $\mathbb{G}^{2k}$ for each $\mathcal{O}_{\text{enc}}$ query.
- In game G_4, $[\mathbf{c}_1]$ is uniform over $\mathsf{span}([\mathbf{A}_0])$ for each $\mathcal{O}_{\text{enc}}$ query.

We can build an adversary $\mathcal{B}_3$ and show that

$$|\Pr_3[b'=b]-\Pr_4[b'=b]| \le \mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen},\mathcal{B}_3}(\lambda)+2^{-\Omega(\lambda)}. \tag{26}$$

The reduction is straightforward and the proof of (26) is almost the same as (25).

Game G_4-G_5. G_5 is almost the same as G_4 except that a rejection rule is added in $\mathcal{O}_{\mathrm{dec}}$. More precisely, in G_5, an $\mathcal{O}_{\mathrm{dec}}(\mathsf{pred},\psi=([\mathbf{c}],\Pi))$ query is directly rejected if $[\mathbf{c}]\notin \mathrm{span}([\mathbf{A}])$. We have that

$$\begin{aligned}|\Pr_4[b'=b]-\Pr_5[b'=b]| &\le \frac{1}{2}\mathsf{Adv}^{\mathrm{csnd}}_{\mathcal{L}^{\mathrm{snd}},\mathsf{PS},\mathcal{B}_4}(\lambda)+\frac{1}{2}\mathsf{Adv}^{\text{PS-ind}}_{\mathcal{L}^{\mathrm{snd}},\mathsf{PS},\widetilde{\mathsf{PS}},\mathcal{B}_5}(\lambda)+Q_{\mathrm{enc}}\cdot 2^{-\Omega(\lambda)}\\ &+2\lambda\cdot\mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_{2k,k},\mathsf{GGen},\mathcal{B}_2}(\lambda)+\frac{\lambda}{2}\mathsf{Adv}^{\mathrm{csnd}}_{\widetilde{\mathcal{L}^{\mathrm{snd}}},\widetilde{\mathsf{PS}},\mathcal{B}_6}(\lambda)+\frac{\lambda+2}{2}\cdot Q_{\mathrm{enc}}\cdot Q_{\mathrm{dec}}\cdot \mathit{uncert}_{\mathcal{A}}(\lambda)\end{aligned} \tag{27}$$

The proof of (27) is the same as Lemma 9 in [8]. We refer [8] for details.

Game G_5-G_6. G_6 is almost the same as G_5 except for one difference in $\mathcal{O}_{\mathrm{enc}}$.

- In game G_5, $\gamma_1=(\mathbf{k}_0^\top+\tau_1\mathbf{k}_1^\top)\cdot[\mathbf{c}_1]+[\kappa_1]$ for each $\mathcal{O}_{\mathrm{enc}}$ query.
- In game G_6, $\gamma_1=[\mathbf{v}^\top\mathbf{r}_1]+\tau_1\mathbf{k}_1^\top[\mathbf{c}_1]+[\kappa_1]$ where $\mathbf{v}$ is uniformly chosen from $\mathbb{Z}_q^k$ beforehand but will be fixed for each $\mathcal{O}_{\mathrm{enc}}$ query.

We have that

$$|\Pr_5[b'=b]-\Pr_6[b'=b]| \le 2^{-\Omega(\lambda)}. \tag{28}$$

The proof of (28) is almost the same as (15), and is put in our full version [21].

Game G_6-G_7. G_7 is almost the same as G_6 except for one difference in $\mathcal{O}_{\mathrm{enc}}$.

- In game G_6, $\gamma_1=[\mathbf{v}^\top\mathbf{r}_1]+\tau_1\mathbf{k}_1^\top[\mathbf{c}_1]+[\kappa_1]$ for each $\mathcal{O}_{\mathrm{enc}}$ query.
- In game G_7, $\gamma_1\leftarrow[u_1]+\tau_1\mathbf{k}_1^\top[\mathbf{c}_1]+[\kappa_1]$ where $u_1\leftarrow_{\$}\mathbb{Z}_q$ for each $\mathcal{O}_{\mathrm{enc}}$ query. In other words, γ_1 is uniform for each $\mathcal{O}_{\mathrm{enc}}$ query in G_7. We have that

$$|\Pr_6[b'=b]-\Pr_7[b'=b]| \le \mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen},\mathcal{B}_3}(\lambda)+2^{-\Omega(\lambda)}. \tag{29}$$

The proof of (29) is almost the same as that of (16). We can set $\mathbf{r}_1=\mathbf{W}\mathbf{s}$ and $[\mathbf{B}]=\left[\begin{smallmatrix}\mathbf{W}\\ \mathbf{v}^\top\mathbf{W}\end{smallmatrix}\right]\in\mathbb{G}^{(k+1)\times k}$ which has the distribution $\mathcal{U}_{k+1,k}$ overwhelmingly. Then we can reduce the indistinguishability between G_6 and G_7 to the Q_{enc}-fold $\mathcal{U}_{k+1,k}$-MDDH assumption. We omit the detailed proof here.

Note that, in game G_7, $[\kappa_1]$ is not needed any longer since we can just select a uniform γ_1 for each $\mathcal{O}_{\mathrm{enc}}$ query.

Game G_7-G_8. G_8 is almost the same as G_7 except for one difference in $\mathcal{O}_{\mathrm{enc}}$.

- In game G_7, Π_1 is the output of $\mathsf{PSim}(\mathsf{psk},[\mathbf{c}_1])$ for each $\mathcal{O}_{\mathrm{enc}}$ query.
- In game G_8, Π_1 is uniform selected for each $\mathcal{O}_{\mathrm{enc}}$ query.

We can build an adversary $\mathcal{B}_7$ and show that

$$|\Pr_7[b'=b]-\Pr_8[b'=b]| \le \mathsf{Adv}^{\text{pr-proof}}_{\mathsf{PS},\mathcal{B}_7}(\lambda). \tag{30}$$

On input ppk, $\mathcal{B}_7$ uniformly selects $b\leftarrow_{\$}\{0,1\}$ and sets $\mathcal{T}\leftarrow\emptyset$. Then $\mathcal{B}_7$ uniformly selects $\mathbf{k}_0,\mathbf{k}_1\leftarrow_{\$}\mathbb{Z}_q^{2k}$ and sets $[\mathbf{p}_0^\top]\leftarrow\mathbf{k}_0^\top[\mathbf{A}]$, $[\mathbf{p}_1^\top]\leftarrow\mathbf{k}_1^\top[\mathbf{A}]$, $\mathsf{pk}_{\mathsf{KEM}}\leftarrow(\mathsf{ppk},[\mathbf{p}_0^\top],[\mathbf{p}_1^\top])$. Then $\mathcal{B}_7$ calls $\mathcal{A}^{\mathcal{O}_{\mathrm{enc}}(),\mathcal{O}_{\mathrm{dec}}(\cdot,\cdot)}(\mathsf{pk}_{\mathsf{KEM}})$ by simulating the two oracles for $\mathcal{A}$ in the following way.

- For $\mathcal{A}$'s $\mathcal{O}_{\text{enc}}()$ query, $\mathcal{B}_7$ uniformly chooses (ψ_0, γ_0) and calculates τ_0 just like game $G_7(G_8)$. Then $\mathcal{B}_7$ submits an $\mathcal{O}_{\text{sim}}$ query to its own oracle and gets $([\mathbf{c}], \Pi)$ where $[\mathbf{c}]$ is uniform over $\mathcal{L}^{\text{snd}} \backslash \mathcal{L} = \mathsf{span}([\mathbf{A}_0])$ and Π is either an output of $\mathsf{PSim}(\mathsf{psk}, [\mathbf{c}])$ or uniformly chosen from $\boldsymbol{\Pi}$. After that $\mathcal{B}_7$ sets $[\mathbf{c}_1] \leftarrow [\mathbf{c}]$ and $\Pi_1 \leftarrow \Pi$. Then $\mathcal{B}_7$ sets $\boldsymbol{\psi}_{\text{enc}}$, calculates τ_1 from $[\overline{\mathbf{c}_1}]$ and uniformly selects γ_1 just like game $G_7(G_8)$. Finally $\mathcal{B}_7$ returns (ψ_b, γ_b) to $\mathcal{A}$.
- For $\mathcal{A}$'s $\mathcal{O}_{\text{dec}}(\mathsf{pred}, \psi = ([\mathbf{c}], \Pi))$ query, $\mathcal{B}_7$ submits $\mathcal{O}_{\text{ver}}([\mathbf{c}], \Pi)$ query to its own oracle and gets the response K. If $K = \perp$, $\mathcal{B}_7$ returns $\perp$ to $\mathcal{A}$. Since $K = \perp$ means $[\mathbf{c}] \notin \mathsf{span}([\mathbf{A}])$ or the verification $\mathsf{PVer}(\mathsf{psk}, [\mathbf{c}], \Pi)$ does not pass, $\mathcal{B}_7$ acts exactly the same as game $G_7(G_8)$ in such cases. If $[\kappa] = K \neq \perp$, $\mathcal{B}_7$ calculates τ and γ just like game $G_7(G_8)$. Then $\mathcal{B}_7$ tests if $([\mathbf{c}], \Pi) \in \boldsymbol{\psi}_{\text{enc}}$ or $\mathsf{pred}(\gamma) = 0$ or $\vee\tau \in \mathcal{T}$ happens. If so, $\mathcal{B}_7$ returns $\perp$ to $\mathcal{A}$. Otherwise $\mathcal{B}_7$ returns γ to $\mathcal{A}$.

Finally, according to $\mathcal{A}$'s output b', $\mathcal{B}_7$ outputs 1 if and only if $b' = b$. It is clear that if Π is an output of $\mathsf{PSim}(\mathsf{psk}, [\mathbf{c}])$ for each $\mathcal{O}_{\text{sim}}$ query, $\mathcal{B}_7$ perfectly simulates game G_7 for $\mathcal{A}$. And if Π is uniformly chosen from $\boldsymbol{\Pi}$ for each $\mathcal{O}_{\text{sim}}$ query, $\mathcal{B}_7$ perfectly simulates game G_8 for $\mathcal{A}$. Thus (30) follows.

Game $G_8 - G_9$. G_9 is the same as G_8 except for one difference in $\mathcal{O}_{\text{enc}}$.

- In game G_8, $[\mathbf{c}_1]$ is uniform selected from $\mathsf{span}([\mathbf{A}_0])$ for each $\mathcal{O}_{\text{enc}}$ query.
- In game G_9, $[\mathbf{c}_1]$ is uniform selected from $\mathbb{G}^{2k}$ for each $\mathcal{O}_{\text{enc}}$ query.

We can build an adversary $\mathcal{B}_3$ and show that

$$|\Pr_8[b' = b] - \Pr_9[b' = b]| \leq \mathsf{Adv}^{\text{mddh}}_{\mathcal{U}_k, \mathsf{GGen}, \mathcal{B}_3}(\lambda) + 2^{-\Omega(\lambda)}. \tag{31}$$

The reduction is straightforward and the proof of (31) is the same as the proof for (25). We omit the details here.

Game G_9. In game G_9, (ψ_1, Π_1) is uniform over $\Psi \times \Gamma$ for each $\mathcal{O}_{\text{enc}}$ query, which distributes exactly the same as (ψ_0, Π_0). Thus we have

$$\Pr_9[b' = b] = \frac{1}{2}. \tag{32}$$

Finally, Theorem 3 follows from (23), Lemma 1, (24)−(32). ■

Theorem 4. *The KEM* $\mathsf{KEM}_{\mathsf{qps}}$ *in Figure 12 is RER secure. Specifically, for each PPT adversary* $\mathcal{A}$ *with negligible uncertainty* $\text{uncert}_{\mathcal{A}}(\lambda)$*, the advantage* $\mathsf{Adv}^{\text{rer}}_{\mathsf{KEM}_{\mathsf{qps}}, \mathcal{A}}(\lambda) \leq 2^{-\Omega(\lambda)}$.

Proof of Theorem 4. In $\mathsf{Exp}^{\text{rer-}b}_{\mathsf{KEM}_{\mathsf{qps}}, \mathcal{A}}(\lambda)$, among all the $\mathcal{O}_{\text{cha}}(\psi, \mathsf{pred})$ queries submitted by $\mathcal{A}$, if $\psi \notin \boldsymbol{\psi}_{\text{ran}}$, the oracle $\mathcal{O}_{\text{cha}}$ will answer $\mathcal{A}$ with $\mathsf{pred}(\mathsf{KDec}(\mathsf{sk}_{\mathsf{KEM}}, \psi))$. Thus no information about b is leaked to $\mathcal{A}$.

Therefore, we only consider those $\mathcal{O}_{\text{cha}}(\psi, \mathsf{pred})$ queries such that $\psi = ([\mathbf{c}], \Pi) \in \boldsymbol{\psi}_{\text{ran}}$. In this case, both $[\mathbf{c}]$ and Π are uniform.

If $b = 0$, $\mathcal{O}_{\text{cha}}(\psi, \mathsf{pred})$ will always return 0 in $\mathsf{Exp}^{\text{rer-0}}_{\mathsf{KEM}_{\mathsf{qps}}, \mathcal{A}}(\lambda)$.

If $b = 1$, $\mathcal{O}_{\mathrm{cha}}(\psi, \mathsf{pred})$ will use $\mathsf{KDec}(\mathsf{sk}_{\mathsf{KEM}}, \psi)$ to decapsulate ψ. More precisely, it will invoke $\mathsf{PVer}(\mathsf{psk}, [\mathbf{c}], \Pi)$ to obtain $(v, [\kappa])$ and output $\perp$ if $v = 0$. By the proof uniqueness of PS and the uniformness of Π, the probability that $v = 1$ in this query is at most $\frac{1}{|\mathbf{\Pi}|}$. Taking into account all the Q_{cha} queries, a union bound suggests that $\mathcal{O}_{\mathrm{cha}}(\psi, \mathsf{pred})$ always outputs 0 in $\mathsf{Exp}^{\text{rer-1}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda)$ except with probability at most $\frac{Q_{\mathrm{cha}}}{|\mathbf{\Pi}|} = 2^{-\Omega(\lambda)}$. Thus

$$\mathsf{Adv}^{\mathrm{rer}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda) = \left|\Pr\left[\mathsf{Exp}^{\text{rer-0}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{rer-1}}_{\mathsf{KEM}_{\mathsf{qps}},\mathcal{A}}(\lambda) = 1\right]\right| \leq 2^{-\Omega(\lambda)}.$$

Acknowledgments. Lin Lyu, Shengli Liu and Shuai Han are supported by the National Natural Science Foundation of China (Grant Nos. 61672346, 61373153). Dawu Gu is supported by the National Natural Science Foundation of China (Grant No. U1636217) together with Program of Shanghai Academic Research Leader (16XD1401300). The authors greatly thank the anonymous reviewers of PKC 2018 for their comments and suggestions.

References

[1] Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. Lecture Notes in Computer Science, vol. 5479, pp. 1–35. Springer (2009), https://doi.org/10.1007/978-3-642-01001-9_1

[2] Böhl, F., Hofheinz, D., Kraschewski, D.: On definitions of selective opening security. In: Fischlin, M., Buchmann, J.A., Manulis, M. (eds.) PKC 2012. Lecture Notes in Computer Science, vol. 7293, pp. 522–539. Springer (2012), https://doi.org/10.1007/978-3-642-30057-8_31

[3] Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. Lecture Notes in Computer Science, vol. 8043, pp. 435–460. Springer (2013), https://doi.org/10.1007/978-3-642-40084-1_25

[4] Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. Lecture Notes in Computer Science, vol. 1880, pp. 432–450. Springer (2000), http://dx.doi.org/10.1007/3-540-44598-6_27

[5] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: FOCS 1999. pp. 523–534. IEEE Computer Society (1999), https://doi.org/10.1109/SFFCS.1999.814626

[6] Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 381–402. Springer (2010), https://doi.org/10.1007/978-3-642-13190-5_20

[7] Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly cca-secure encryption without pairings. In: Fischlin, M., Coron, J. (eds.) EUROCRYPT 2016. Lecture Notes in Computer Science, vol. 9665, pp. 1–27. Springer (2016), http://dx.doi.org/10.1007/978-3-662-49890-3_1

[8] Gay, R., Hofheinz, D., Kohl, L.: Kurosawa-desmedt meets tight security. In: Katz and Shacham [16], pp. 133–160, https://doi.org/10.1007/978-3-319-63697-9_5

[9] Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984), https://doi.org/10.1016/0022-0000(84)90070-9
[10] Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 70–88. Springer (2011), https://doi.org/10.1007/978-3-642-25385-0_4
[11] Hofheinz, D.: All-but-many lossy trapdoor functions. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 209–227. Springer (2012), https://doi.org/10.1007/978-3-642-29011-4_14
[12] Hofheinz, D.: Adaptive partitioning. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017. Lecture Notes in Computer Science, vol. 10212, pp. 489–518 (2017), https://doi.org/10.1007/978-3-319-56617-7_17
[13] Hofheinz, D., Jager, T., Rupp, A.: Public-key encryption with simulation-based selective-opening security and compact ciphertexts. In: Hirt, M., Smith, A.D. (eds.) TCC 2016. Lecture Notes in Computer Science, vol. 9986, pp. 146–168 (2016), https://doi.org/10.1007/978-3-662-53644-5_6
[14] Huang, Z., Liu, S., Qin, B.: Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. Lecture Notes in Computer Science, vol. 7778, pp. 369–385. Springer (2013), https://doi.org/10.1007/978-3-642-36362-7_23
[15] Huang, Z., Liu, S., Qin, B., Chen, K.: Fixing the sender-equivocable encryption scheme in eurocrypt 2010. In: 5th International Conference on Intelligent Networking and Collaborative Systems 2013. pp. 366–372. IEEE (2013), http://dx.doi.org/10.1109/INCoS.2013.69
[16] Katz, J., Shacham, H. (eds.): Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III, Lecture Notes in Computer Science, vol. 10403. Springer (2017), https://doi.org/10.1007/978-3-319-63697-9
[17] Lai, J., Deng, R.H., Liu, S., Weng, J., Zhao, Y.: Identity-based encryption secure against selective opening chosen-ciphertext attack. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. Lecture Notes in Computer Science, vol. 8441, pp. 77–92. Springer (2014), https://doi.org/10.1007/978-3-642-55220-5_5
[18] Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: Katz and Shacham [16], pp. 332–364, https://doi.org/10.1007/978-3-319-63697-9_12
[19] Liu, S., Paterson, K.G.: Simulation-based selective opening CCA security for PKE from key encapsulation mechanisms. In: Katz, J. (ed.) PKC 2015. Lecture Notes in Computer Science, vol. 9020, pp. 3–26. Springer (2015), http://dx.doi.org/10.1007/978-3-662-46447-2_1
[20] Lyu, L., Liu, S., Han, S.: Public-key encryption with tight simulation-based selective-opening security. The Computer Journal pp. 1–31 (2017), http://dx.doi.org/10.1093/comjnl/bxx080
[21] Lyu, L., Liu, S., Han, S., Gu, D.: Tightly sim-so-cca secure public key encryption from standard assumptions. Cryptology ePrint Archive, Report 2018/030 (2018), https://eprint.iacr.org/2018/030
[22] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Dwork, C. (ed.) ACM Symposium on Theory of Computing 2008. pp. 187–196. ACM (2008), http://doi.acm.org/10.1145/1374376.1374406