Key Encapsulation Mechanism with Explicit Rejection in the Quantum Random Oracle Model

Haodong Jiang[1,2,4], Zhenfeng Zhang[2,3], and Zhi Ma[1,4]

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, Henan, China
[2] TCA Laboratory, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, Henan, China
hdjiang13@gmail.com, {zfzhang}@tca.iscas.ac.cn, {ma_zhi}@163.com

Abstract. The recent post-quantum cryptography standardization project launched by NIST increased the interest in generic key encapsulation mechanism (KEM) constructions in the quantum random oracle (QROM). Based on a OW-CPA-secure public-key encryption (PKE), Hofheinz, Hövelmanns and Kiltz (TCC 2017) first presented two generic constructions of an IND-CCA-secure KEM with quartic security loss in the QROM, one with implicit rejection (a pseudorandom key is return for an invalid ciphertext) and the other with explicit rejection (an abort symbol is returned for an invalid ciphertext). Both are widely used in the NIST Round-1 KEM submissions and the ones with explicit rejection account for 40%. Recently, the security reductions have been improved to quadratic loss under a standard assumption, and be tight under a non-standard assumption by Jiang et al. (Crypto 2018) and Saito, Xagawa and Yamakawa (Eurocrypt 2018). However, these improvements only apply to the KEM submissions with implicit rejection and the techniques do not seem to carry over to KEMs with explicit rejection.

In this paper, we provide three generic constructions of an IND-CCA-secure KEM with explicit rejection, under the same assumptions and with the same tightness in the security reductions as the aforementioned KEM constructions with implicit rejection (Crypto 2018, Eurocrypt 2018). Specifically, we develop a novel approach to verify the validity of a ciphertext in the QROM and use it to extend the proof techniques for KEM constructions with implicit rejection (Crypto 2018, Eurocrypt 2018) to our KEM constructions with explicit rejection. Moreover, using an improved version of one-way to hiding lemma by Ambainis, Hamburg and Unruh (ePrint 2018/904), for two of our constructions, we present tighter reductions to the standard IND-CPA assumption. Our results directly apply to 9 KEM submissions with explicit rejection, and provide tighter reductions than previously known (TCC 2017).

Keywords: quantum random oracle model · key encapsulation mechanism · explicit rejection · generic construction

1 Introduction

Indistinguishability against chosen-ciphertext attacks (IND-CCA) [1] is considered to be a standard security notion of a key encapsulation mechanism (KEM). Efficient IND-CCA-secure KEMs are usually constructed in the random oracle model (ROM) [2], where a hash function is idealized to be a publicly accessible random oracle (RO). Generic constructions of an efficient IND-CCA-secure KEM in the ROM are well studied by Dent [3] and Hofheinz, Hövelmanns and Kiltz [4].

The constructions of IND-CCA-secure KEMs in [4] are essentially various KEM variants of the Fujisaki-Okamoto (FO) transformation [5, 6] and the REACT/GEM transformation [7, 8], which turn a weakly secure public-key encryption (PKE) into an IND-CCA-secure KEM. These constructions can be classified into two categories according to the value for an invalid ciphertext during the decapsulation. One category contains the constructions with explicit rejection which return a rejection symbol $\perp$ when decapsulating an invalid ciphertext, including $\mathrm{FO}^{\perp}$, $\mathrm{FO}^{\perp}_{m}$, $\mathrm{QFO}^{\perp}_{m}$, $\mathrm{U}^{\perp}$, $\mathrm{U}^{\perp}_{m}$, $\mathrm{QU}^{\perp}_{m}$, where FO denotes the class of transformations that turn a PKE with standard security (one-wayness against chosen-plaintext attacks (OW-CPA) or indistinguishability against chosen-plaintext attacks (IND-CPA)) into an IND-CCA KEM, U denotes the class of transformations that turn a PKE with non-standard security (e.g., OW-PCA, one-way against plaintext checking attack [7, 8]) or a deterministic PKE (DPKE, where the encryption algorithm is deterministic) into an IND-CCA-secure KEM, m[5] (without m) means $K = H(m)$ ($K = H(m, c)$), $\not\perp$ ($\perp$) means implicit (explicit) rejection and Q means an additional Targhi-Unruh hash [9] (a length-preserving hash function that has the same domain and range size) is added into the ciphertext. The second category contains the KEM constructions with implicit rejection where a pseudorandom key is returned for an invalid ciphertext, including $\mathrm{FO}^{\not\perp}$, $\mathrm{FO}^{\not\perp}_{m}$, $\mathrm{QFO}^{\not\perp}_{m}$, $\mathrm{U}^{\not\perp}$, $\mathrm{U}^{\not\perp}_{m}$, $\mathrm{QU}^{\not\perp}_{m}$.

Recently, the National Institute of Standards and Technology (NIST) launched a Post-Quantum Cryptography Project and published a call for submissions of quantum-resistant public-key cryptographic algorithms including digital-signature, PKE, and KEM (or key exchange) [10]. Among the 69 Round-1 submissions [10], there are 39 KEM proposals. Specially, 25 NIST submissions followed above constructions in [4] to achieve IND-CCA security.

Generic constructions in the ROM have gathered renewed interest in the post-quantum setting, where adversaries are equipped with a quantum computer. In the real world, quantum adversary can execute hash functions (the instantiation of the RO) on an arbitrary superposition of inputs. Therefore, for evaluating the post-quantum security, one needs to perform the analysis in the quantum random oracle model (QROM), introduced by [11]. Unfortunately, the QROM is quite difficult to work with, since many proof techniques in the ROM including adaptive programmability or extractability, have no analog in the QROM [11].

[5] The message m here is picked at random from the message space of underlying PKE.

Hofheinz et al. [4] first presented two generic KEM constructions in the QROM, $\text{QFO}_m^{\not\perp}$ and $\text{QFO}_m^{\perp}$, where a Targhi-Unruh hash [9] is used to follow the technique in [9, 12] to prove the QROM security. However, the security reductions are highly non-tight with quartic loss.

Subsequently, Saito, Xagawa and Yamakawa [13] and Jiang et al. [14] extended the technique in [11] to remove the Targhi-Unruh hash and tighten above security reductions. Jiang et al. [14] presented security reductions for $\text{FO}_m^{\not\perp}$ and $\text{FO}^{\not\perp}$ with quadratic loss from standard OW-CPA security of underlying PKE. Saito et al. [13] proposed a new security notion for DPKE called the disjoint simulatability (DS) security, and showed that the $\text{U}_m^{\not\perp}$ transformation can convert a DS-secure DPKE into an IND-CCA-secure KEM with a tight security reduction. However, above improvements were only achieved for KEM constructions with implicit rejection due to the obstacle that the simulator needs to verify the validity of a ciphertext [13, 14].

Among the 25 NIST submissions where the generic constructions in [4] are used, 10 submissions (40%) use generic KEM constructions with explicit rejection [10] including EMBLEM and R.EMBLEM, Lepton, NTRU-HRSS-KEM, BIG QUAKE, DAGS, HQC, LOCKER, QC-MDPC, RQC and ThreeBears. Except ThreeBears [15] which provides a sketch of a QROM security reduction with quadratic loss based on their specific scheme, the other 9 submissions that use the transformation $\text{QFO}_m^{\perp}$ or $\text{QFO}^{\perp}$[6] only have a highly non-tight QROM security reduction with quartic loss.

In this paper, we focus on generic constructions of an IND-CCA-secure KEM with explicit rejection, under the same assumptions and with the same tightness in security reduction as KEMs with implicit rejection [13, 14].

1.1 Our Contributions

We present three generic constructions of an IND-CCA-secure KEM with explicit rejection, $\text{HFO}_m^{\perp}$, $\text{HFO}^{\perp}$ and $\text{HU}_m^{\perp}$, from a weakly secure PKE, by revisiting the *plaintext confirmation* method in the QROM (refer to Subsection 1.2 for details). $\text{HFO}_m^{\perp}$, $\text{HFO}^{\perp}$ and $\text{HU}_m^{\perp}$ are identical with the existing generic constructions with explicit rejection $\text{QFO}_m^{\perp}$, $\text{QFO}^{\perp}$ and $\text{QU}_m^{\perp}$ in [4] except for the hash used in *plaintext confirmation*. In $\text{HFO}_m^{\perp}$, $\text{HFO}^{\perp}$ and $\text{HU}_m^{\perp}$, a conventional hash function works. In contrast, in $\text{QFO}_m^{\perp}$, $\text{QFO}^{\perp}$ and $\text{QU}_m^{\perp}$, the hash function is required to be length-preserving, a Targhi-Unruh hash function. A length-preserving hash function will lead to a significant increase of encapsulation size in the case that the message space elements are strictly larger than a single hash value, e.g., NTRU-HRSS-KEM [16]. Thus, our constructions can directly help to reduce the encapsulation size for these KEM schemes.

[6] Actually, $\text{QFO}^{\perp}$ was not definitely presented by [4]. But, its construction is the same as $\text{QFO}_m^{\perp}$ except that $K = H(m, c)$ and its security can be easily derived from the security proof of $\text{QFO}_m^{\perp}$ in [4].

Table 1: Generic KEM constructions with explicit rejection in the QROM.

Constructions	Underlying security	Security bound
$\mathrm{QFO}_m^{\perp}$ and $\mathrm{QFO}^{\perp}$ [4]	OW-CPA	$q\sqrt{q^2\delta + q\sqrt{\epsilon}}$
Our $\mathrm{HFO}_m^{\perp}$ and $\mathrm{HFO}^{\not\perp}$	IND-CPA	$q\sqrt{\delta} + \sqrt{q\epsilon}$
Our $\mathrm{HFO}_m^{\perp}$ and $\mathrm{HFO}^{\not\perp}$	OW-CPA	$q\sqrt{\delta} + q\sqrt{\epsilon}$
Our $\mathrm{HU}_m^{\perp}$	DS	ϵ

In terms of QROM security reductions, ours are much tighter than the ones of $\mathrm{QFO}_m^{\perp}$ and $\mathrm{QFO}^{\perp}$ in [4], see Table 1. For any correctness error[7] δ ($0 \leq \delta < 1$), our obtained security bounds for $\mathrm{HFO}_m^{\perp}$ and $\mathrm{HFO}^{\perp}$ are both $\epsilon' \approx q\sqrt{\delta} + q\sqrt{\epsilon}$ which are much tighter than $\epsilon' \approx q\sqrt{q^2\delta + q\sqrt{\epsilon}}$ in [4], where ϵ' is the success probability of an adversary against the IND-CCA security of the resulting KEM, ϵ is the success probability of another adversary against the OW-CPA security of the underlying PKE, and q is the total number of $\mathcal{B}$'s queries to various oracles. For $\mathrm{HU}_m^{\perp}$, the IND-CCA security of the resulting KEM is tightly reduced to the DS security of the underlying DPKE with perfect correctness[8]. That is, our generic constructions with explicit rejection achieve the same tightness in security reductions under identical assumptions as the corresponding KEM constructions with implicit rejection $\mathrm{FO}_m^{\not\perp}$, $\mathrm{FO}^{\not\perp}$ and $\mathrm{U}_m^{\not\perp}$ in [14, 13]. Moreover, we also present tighter QROM security reductions, $\epsilon' \approx q\sqrt{\delta} + \sqrt{q\epsilon}$, for $\mathrm{HFO}_m^{\perp}$ and $\mathrm{HFO}^{\perp}$ based on the IND-CPA security of the underlying PKE.

Accordingly, our tighter QROM security reductions can directly provide more reliable security guarantee for the IND-CCA-secure KEM submissions with explicit rejection where $\mathrm{QFO}_m^{\perp}$ and $\mathrm{QFO}^{\perp}$ are used, e.g., NTRU-HRSS-KEM [16], see Table 2.

Table 2: IND-CCA-secure KEM submissions for which our tighter security reductions of $\mathrm{HFO}_m^{\perp}$ and $\mathrm{HFO}^{\perp}$ can directly provide more reliable security guarantee in the QROM.

Constructions	Submission
$\mathrm{HFO}_m^{\perp}$	NTRU-HRSS-KEM,DAGS,QC-MDPC
$\mathrm{HFO}^{\perp}$	EMBLEM and R.EMBLEM, Lepton, BIG QUAKE,HQC,LOCKER,RQC

1.2 Techniques

The difference between KEM constructions with explicit rejection and implicit rejection is the behavior of the decapsulation algorithm on an invalid ciphertext.

[7] The probability of decryption failure in a legitimate execution of the scheme.

[8] Perfect correctness, i.e., $\delta = 0$ is required by [13]. Here, we just follow this assumption.

In a KEM construction with implicit rejection, a pseudorandom key is returned instead of a rejection symbol $\perp$, which prevents the adversary from judging the validity of a ciphertext by querying the decapsulation oracle. Thus, the simulation of a decapsulation oracle does not need to verify if a given ciphertext is valid or not, and can use an identical hidden random orale H_q to answer the decapsulation queries for both valid ciphertexts and invalid ciphertexts [13, 14].

However, in the case of explicit rejection, the simulation of the decapsulation oracle has to first verify the validity of a given ciphertext, which is the key obstacle for the techniques in [13, 14] to carry over. Here, before showing how to overcome this obstacle, we first review two general methods [3, 4] used in the ROM to achieve an IND-CCA-secure KEM construction with explicit rejection, the γ-*spreadness* assumption and *plaintext confirmation*.

γ-spread. By assuming the underlying PKE to be γ-*spread*, we can obtain a KEM construction with explicit rejection. A γ-*spread* PKE, introduced by Fujisaki and Okamoto [5, 6], roughly speaking, requires that ciphertexts (generated by the probabilistic encryption algorithm) have sufficiently large entropy. It plays an important role in the ROM security proofs of the original FO transformation [5, 6], and $\mathrm{FO}_m^{\perp}$ and $\mathrm{FO}^{\perp}$ (the KEM variants of FO transformation) in [4]. If the underlying PKE is γ-*spread*, we can easily verify the validity of a ciphertext by checking if the ciphertext is derived by using the randomness produced by the RO [4–6]. In the ROM, adversarial queries to the RO can be recorded by a list, which makes the above checking feasible. Unfortunately, as discussed in [14], in the QROM, it is difficult to learn the actual content of an adversarial RO query.

Plaintext confirmation. Adding an extra hash value of the plaintext to the ciphertext, called *plaintext confirmation*[9], is another method to achieve a construction with explicit rejection. This method was first introduced by [3, Table 4] in the ROM, in the context of a generic construction of an IND-CCA-secure KEM with explicit rejection based on a OW-CPA-secure DPKE, which can be viewed as a simpler version of the REACT construction. Our $\mathrm{HU}_m^{\perp}$ transformation is essentially the same as [3, Table 4].

In particular, a valid ciphertext $c = (c_1, c_2)$ is produced by $c_1 = Enc(pk, m)$, $c_2 = H'(m)$ for some m, where Enc is the encryption algorithm of the underlying DPKE. In the ROM, the validity of a ciphertext ($c = (c_1, c_2)$) can be verified by testing if (c_1, c_2) is contained in a list (m,c_1,c_2), where m is an adversarial query input to H', ($c_1 = Enc(pk, m), c_2 = H'(m)$) is the corresponding ciphertext. However, this verification method will not work in the QROM due to the same reason as in the case of the γ-*spreadness* assumption method that it's hard to learn adversarial query inputs.

In [4], Hofheinz et al. follow Targhi and Unruh's technique [9, 12] and simulate H' using a random polynomial of degree $2q_{H'}$ over a finite field $\mathbb{F}_{2^n}$, where $q_{H'}$ is the number of adversarial queries to H' and n is the range size of H'. For a given

[9] This name comes from Bernstein and Persichetti's paper [17].

ciphertext $c = (c_1, c_2)$, the simulator verifies the validity by checking if c_1 lies within the encryptions of the roots of $H'(X) - c_2$. To make H' invertible, H' is required to be length-preserving. Additionally, the technique in [4, 9] requires two instances of the one-way to hiding (OW2H) lemma [18, Lemma 6.2], which is a practical tool to prove the indistinguishability between games where the random oracles are reprogrammed. Nevertheless, the OW2H lemma will inherently incur a quadratic security loss. Thus, the security reductions of $\mathrm{QFO}_m^{\perp}$ and $\mathrm{QFO}^{\not\perp}$ in [4] suffer a quartic security loss.

In this paper, we develop a novel verification method for the KEM construction with explicit rejection based on *plaintext confirmation*, and circumvent the learning of adversarial queries. Specifically, the simulator replaces H' by $H'_q \circ Enc(pk, \cdot)$[10], where H'_q is a secret random function that is not given to the adversary. We require $Enc(pk, \cdot)$ to be indistinguishable from an injective function for any efficient quantum adversary. Thus, in the adversary's view, $H'_q \circ Enc(pk, \cdot)$ is a perfect random oracle. Then, we note that if c is a valid ciphertext, $H'_q(c_1) = c_2$, and if c is invalid, then $H'_q(c_1) = c_2$ with negligible probability. Thereby, using H'_q, we can verify the validity of a ciphertext $c = (c_1, c_2)$ just by testing if $H'_q(c_1) = c_2$ or not.

With this novel verification method for the validity of a ciphertext, we can extend the techniques in [13, 14] to the constructions with explicit rejection in this paper. Thus, the OW2H lemma is instantiated only once in the security reductions for $\mathrm{HFO}^{\perp}$ and $\mathrm{HFO}_m^{\perp}$, and never used during the security reduction for $\mathrm{HU}_m^{\perp}$, which lead to the same security loss as the corresponding KEM constructions with implicit rejection in [13, 14].

Tighter reduction from IND-CPA. Different from the adversary against the OW-CPA security of PKE, the adversary against the IND-CPA security of PKE knows the plaintexts m_0 and m_1 of which one is encrypted to obtain the challenge ciphertext. Thus, the simulator can make an elaborate analysis of the RO-query inputs, e.g., testing whether m_0 (or m_1) has been queried to the RO [4], and determine which one of the query inputs can be used to break the IND-CPA security instead of just uniformly choosing at random. Particularly, in the ROM, [4] presents tight reductions for FO transformations from IND-CPA security of underlying PKE to IND-CCA security of resulting KEM. However, the techniques in [4] require the simulator to maintain a RO-query list which is difficult to implement in the QROM. In our case, we instead use a semi-classical oracle technique (refer to Lemma 3 for details), recently introduced by Ambainis, Hamburg and Unruh [19], to test whether m_0 (or m_1) has been queried. Then, the security bound $q\sqrt{\delta} + q\sqrt{\epsilon}$ is improved to be $q\sqrt{\delta} + \sqrt{q\epsilon}$.

1.3 Discussion

As in prior works [4, 13, 14], we do not provide a general definition of explicit/implicit rejection on the KEM level. Although on first sight it seems these

[10] Such a non-adaptive RO programming technique is also used in [11, 13, 14].

notions could be clearly defined, it turns out that grabbing these concepts formally is quite challenging. This seems to be mostly due to the fact that the notion of an "invalid ciphertext", on which the definition of explicit/implicit rejection would likely be based, remains elusive as well.

KEMs either have implicit rejection or explicit rejection, or do not satisfy either of these two. The advantages and disadvantages of explicit rejection and implicit rejection for specific KEM constructions have been discussed by [15] and [17]. The goal of this paper is not to take part in this discussion, but rather to expand the proof techniques from KEMs with implicit rejection to KEMs with explicit rejection. In particular, we show security reductions in the QROM for our generic KEM constructions with explicit rejection that preserve the same assumptions and tightness as previously known for KEMs with implicit rejection [13, 14].

1.4 Related Work

In a concurrent and independent work, Zhandry [20] presented a proof in the QROM for original FO transformation in [5, 6] from OW-CPA security of underlying PKE and one-time security of underlying symmetric key encryption to quantum CCA security of resulting PKE (quantum CCA security of PKE [21] is identical to CCA security except that adversaries can make decryption queries in quantum superpositions). However, the security proofs for KEM variants of FO transformation in this paper were not presented and the tightness was not discussed either. Moreover, their proof techniques are quite different from ours, and require γ-spread assumption of underlying PKE.

1.5 Future work

We note that the Targhi-Unruh hash is removed in generic KEM constructions with implicit rejection [13, 14]. However, a conventional extra hash (although not a Targhi-Unruh hash) is still required in our generic KEM constructions with explicit rejection, just like in the generic construction in the ROM [3, Table 4]. The ThreeBears [15] claims that removing this extra hash will not significantly impact the IND-CCA security of their KEM scheme even though the explicit rejection is used. Indeed, it seems possible that if the underlying PKE has some specific algebraic structure which can be used for the validity verification of the ciphertext, this extra hash can be removed even in a construction with explicit rejection.

In our future work, we will research the specific algebraic structure of the underlying PKE, which can help to achieve an IND-CCA-secure KEM construction with explicit rejection and without this extra hash.

2 Preliminaries

Symbol description. A security parameter is denoted by λ. The abbreviation PPT stands for probabilistic polynomial time. $\mathcal{K}$, $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{R}$ are denoted as

key space, message space, ciphertext space and randomness space, respectively. Given a finite set X, we denote the sampling of a uniformly random element x by $x \xleftarrow{\$} X$. Denote the sampling from some distribution D by $x \leftarrow D$. $x =? y$ is denoted as an integer that is 1 if $x = y$, and otherwise 0. $\Pr[P : G]$ is the probability that the predicate P holds true where free variables in P are assigned according to the program in G. Denote deterministic (probabilistic) computation of an algorithm A on input x by $y := A(x)$ ($y \leftarrow A(x)$). Let $|X|$ be the cardinality of set X. A^H means that the algorithm A gets access to the oracle H. $f \circ g(\cdot)$ means $f(g(\cdot))$.

2.1 Quantum Random Oracle Model

We refer the reader to [22] for basic of quantum computation.

Random oracle model (ROM) [2] is an idealized model, where a hash function is modeled as a publicly accessible random oracle. In quantum setting, an adversary with quantum computer can off-line evaluate the hash function on an arbitrary superposition of inputs. As a result, the quantum adversary should be allowed to query the random orale with quantum state. We call this the quantum random oracle model (QROM), introduced by Boneh et al. [11]. Particularly, [11] argued that to prove post-quantum security one needs to prove security in the QROM.

Tools. Next, we will present several existing lemmas that we will use throughout the paper.

Lemma 1 (Simulating the random oracle [23, Theorem 6.1]). *Let H be an oracle drawn from the set of $2q$-wise independent functions uniformly at random. Then the advantage any quantum algorithm making at most q queries to H has in distinguishing H from a truly random function is identically 0.*

Lemma 2 (Generic search problem [24, 25, 14]). *Let $\gamma \in [0, 1]$. Let Z be a finite set. $F : Z \to \{0, 1\}$ is the following random function: For each z, $F(z) = 1$ with probability p_z ($p_z \leq \gamma$), and $F(z) = 0$ else. Let N be the function with $\forall z : N(z) = 0$. If an oracle algorithm A makes at most q quantum queries to F (or N), then*

$$\left|\Pr[b = 1 : b \leftarrow A^F] - \Pr[b = 1 : b \leftarrow A^N]\right| \leq 2q\sqrt{\gamma}.$$

Particularly, the probability of A finding a z such that $F(z) = 1$ is at most $2q\sqrt{\gamma}$, i.e., $\Pr[F(z) = 1 : z \leftarrow A^F] \leq 2q\sqrt{\gamma}$.

One way to hiding (OW2H) lemma [18, Lemma 6.2] is a practical tool to argue the indistinguishability between games where the random oracles are reprogrammed. Following are improved versions of OW2H lemma, recently introduced by [19].

Lemma 3 (Semi-classical OW2H [19, Theorem 1]). *Let $S \subseteq X$ be random. Let $\mathcal{O}_1$, $\mathcal{O}_2$ be oracles with domain X and codomain Y such that $\mathcal{O}_1(x) = \mathcal{O}_2(x)$ for any $x \notin S$. Let z be a random bitstring. ($\mathcal{O}_1$, $\mathcal{O}_2$, S and z may have arbitrary joint distribution D.) Let f_S be the indicator function, where $f_S(x) = 1$ if $x \in S$ and 0 otherwise. Let $\mathcal{O}_S^{SC}$ be an oracle that performs the semi-classical measurements corresponding to the projectors M_y when queried with $|x\rangle$, where $M_y := \sum_{x \in X : f_S(x) = y} |x\rangle\langle x|$ ($y \in 0, 1$). Let $\mathcal{O}_2 \backslash S$ ("$\mathcal{O}_2$ punctured on S") be an oracle that first queries $\mathcal{O}_S^{SC}$ and then $\mathcal{O}_2$.*

Let $A^{\mathcal{O}_1}(z)$ be an oracle algorithm with query number d. Denote Find as the event that in the execution of $A^{\mathcal{O}_2 \backslash S}(z)$, $\mathcal{O}_S^{SC}$ ever outputs 1 during semi-classical measurements.

Let

$$\begin{aligned} P_{left} &= \Pr[b = 1 : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, b \leftarrow A^{\mathcal{O}_1}(z)] \\ P_{right} &= \Pr[b = 1 \wedge \neg \mathsf{Find} : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, b \leftarrow A^{\mathcal{O}_2 \backslash S}(z)] \\ P_{find} &:= \Pr[\mathsf{Find} : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, A^{\mathcal{O}_2 \backslash S}(z)]. \end{aligned}$$

Then
$|P_{left} - P_{right}| \leq \sqrt{(d+1)P_{find}}$ *and* $\left|\sqrt{P_{left}} - \sqrt{P_{right}}\right| \leq \sqrt{(d+1)P_{find}}$.

Remark: There are several other definitions of P_{right} in [19, Theorem 1]. In this paper, we just need above definition in our security proofs.

Semi-classical oracle. Roughly speaking, semi-classical oracle O_S^{SC} only measures the output $|f_S(x)\rangle$ but not the input $|x\rangle$. Formally, for a query to O_S^{SC} with $\sum_{x,z} a_{x,z}|x\rangle|z\rangle$, $\mathcal{O}_S^{SC}$ does the following

1. initialize a single qubit L with $|0\rangle$,
2. transform $\sum_{x,z} a_{x,z}|x\rangle|z\rangle|0\rangle$ into $\sum_{x,z} a_{x,z}|x\rangle|z\rangle|f_S(x)\rangle$,
3. measure L.

Then, after performing a semi-classical measurement, the query state will become $\sum_{x,z:f_S(x)=y} a_{x,z}|x\rangle|z\rangle$ (non-normalized) if the measurement outputs y ($y \in 0, 1$).

Lemma 4 (Search in semi-classical oracle [19, Corollary 1]). *Suppose that S and z are independent, and that A is a q-query algorithm. Let $P_{max} := \max_{x \in X} \Pr[x \in S]$. Then*

$$\Pr[\mathsf{Find} : A^{O_S^{SC}}(z)] \leq 4q \cdot P_{max}.$$

Lemma 5 (OW2H, Probabilities [19, Theorem 3]). *Let $S \subseteq X$ be random. Let $\mathcal{O}_1$, $\mathcal{O}_2$ be oracles with domain X and codomain Y such that $\mathcal{O}_1(x) = \mathcal{O}_2(x)$ for any $x \notin S$. Let z be a random bitstring. ($\mathcal{O}_1$, $\mathcal{O}_2$, S and z may have arbitrary joint distribution D.)*

Let $A^{\mathcal{O}_1}(z)$ be an oracle algorithm with query number d. Let $B^{\mathcal{O}_1}$ be an oracle algorithm that on input z does the following: pick $i \xleftarrow{\$} \{1, \ldots, d\}$, run $A^{\mathcal{O}_1}(z)$ until (just before) the i-th query, measure all query input registers in the computational basis, and output the set T of measurement outcomes. (When A makes less than i queries, B outputs $\perp \notin X$.)

Let

$$P_{left} = \Pr[b = 1 : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, b \leftarrow A^{\mathcal{O}_1}(z)]$$
$$P_{right} = \Pr[b = 1 : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, b \leftarrow A^{\mathcal{O}_2}(z)]$$
$$P_{guess} := \Pr[S \cap T \neq \emptyset : (\mathcal{O}_1, \mathcal{O}_2, S, z) \leftarrow D, T \leftarrow B^{\mathcal{O}_1}(z)].$$

Then
$|P_{left} - P_{right}| \leq 2d\sqrt{P_{guess}}$ *and* $\left|\sqrt{P_{left}} - \sqrt{P_{right}}\right| \leq 2d\sqrt{P_{guess}}$.

2.2 Cryptographic Primitives

Definition 1 (Public-key encryption). *A public-key encryption scheme* PKE $= (Gen, Enc, Dec)$ *consists of a triple of polynomial time (in the security parameter* λ*) algorithms and a finite message space* $\mathcal{M}$*.*

- $Gen(1^\lambda) \to (pk, sk)$*: the key generation algorithm, is a probabilistic algorithm which on input* 1^λ *outputs a public/secret key-pair* (pk, sk)*. Usually, for brevity, we will omit the input of Gen.*
- $Enc(pk, m) \to c$*: the encryption algorithm Enc, on input pk and a message* $m \in \mathcal{M}$*, outputs a ciphertext* $c \leftarrow Enc(pk, m)$*. If necessary, we make the used randomness of encryption explicit by writing* $c := Enc(pk, m; r)$*, where* $r \xleftarrow{\$} \mathcal{R}$ *(*$\mathcal{R}$ *is the randomness space).*
- $Dec(sk, c) \to m$*: the decryption algorithm Dec, is a deterministic algorithm which on input sk and a ciphertext c outputs a message* $m := Dec(sk, c)$ *or a rejection symbol* $\perp \notin \mathcal{M}$*.*

A PKE is determined if *Enc* is deterministic. We denote DPKE to stand for a determined PKE.

Definition 2 (Correctness [4]). *A public-key encryption scheme* PKE *is* δ-correct *if*

$$E[\max_{m \in \mathcal{M}} \Pr[Dec(sk, c) \neq m : c \leftarrow Enc(pk, m)]] \leq \delta,$$

where the expectation is taken over $(pk, sk) \leftarrow Gen$*. We say a PKE is perfectly correct if* $\delta = 0$*.*

Next, we define three security notions, one-wayness against chosen-plaintext attacks (OW-CPA) of PKE, indistinguishability against chosen-plaintext attacks (IND-CPA) of PKE, and disjoint simulatability (DS) of DPKE.

Definition 3 (OW-CPA-secure PKE). *Let* PKE $= (Gen, Enc, Dec)$ *be a public-key encryption scheme with message space* $\mathcal{M}$*. Define* $\mathrm{OW-CPA}$ *game of PKE as in Fig. 1. Define the* $\mathrm{OW-CPA}$ *advantage function of an adversary* $\mathcal{A}$ *against PKE as*

$$\mathtt{Adv}^{\mathrm{OW\text{-}CPA}}_{\mathrm{PKE}}(\mathcal{A}) := \Pr[\mathrm{OW\text{-}CPA}^{\mathcal{A}}_{\mathrm{PKE}} = 1].$$

Game OW-CPA		Game IND-CPA	
1 :	$(pk, sk) \leftarrow Gen$	1 :	$(pk, sk) \leftarrow Gen$
2 :	$m^* \xleftarrow{\$} \mathcal{M}$	2 :	$b \leftarrow \{0,1\}$
3 :	$c^* \leftarrow Enc(pk, m^*)$	3 :	$(m_0, m_1) \leftarrow \mathcal{A}(pk)$
4 :	$m' \leftarrow \mathcal{A}(pk, c^*)$	4 :	$c^* \leftarrow Enc(pk, m_b)$
5 :	**return** $m' =? m^*$	5 :	$b' \leftarrow \mathcal{A}(pk, c^*)$
		6 :	**return** $b' =? b$

Fig. 1: Game OW-CPA and game IND-CPA for PKE.

Definition 4 (IND-CPA-secure PKE). *Let* $\mathrm{PKE} = (Gen, Enc, Dec)$ *be a public-key encryption scheme with message space* $\mathcal{M}$*. Define* $\mathrm{IND-CPA}$ *game of PKE as in Fig. 1, where* m_0 *and* m_1 *have the same length. Define the* $\mathrm{IND-CPA}$ *advantage function of an adversary* $\mathcal{A}$ *against PKE as*

$$\mathtt{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A}) := \left|\Pr[\text{IND-CPA}^{\mathcal{A}}_{\text{PKE}} = 1] - 1/2\right|.$$

Definition 5 (DS-secure DPKE [13]). *Let* $D_{\mathcal{M}}$ *denote an efficiently sampleable distribution on a set* $\mathcal{M}$*. A DPKE scheme (Gen,Enc,Dec) with plaintext and ciphertext spaces* $\mathcal{M}$ *and* $\mathcal{C}$ *is* $D_{\mathcal{M}}$*-disjoint simulatable if there exists a PPT algorithm* S *that satisfies the following.*

- *Statistical disjointness:*

$$\textsc{Disj}_{\text{PKE},S} := \max_{(pk,sk)\in Gen(1^\lambda;\mathcal{R}_{gen})} \Pr[c \in Enc(pk, \mathcal{M}) : c \leftarrow S(pk)]$$

  is negligible, where $\mathcal{R}_{gen}$ *denotes a randomness space for Gen.*
- *Ciphertext indistinguishability: For any PPT adversary* $\mathcal{A}$*,*

$$\mathtt{Adv}^{\text{DS-IND}}_{\text{PKE},D_{\mathcal{M}},S}(\mathcal{A}) := \left| \begin{array}{l} \Pr\left[\mathcal{A}(pk, c^*) \to 1 : \begin{array}{c}(pk, sk) \leftarrow Gen; m^* \leftarrow D_{\mathcal{M}};\\ c^* = Enc(pk, m^*)\end{array}\right] \\ -\Pr[\mathcal{A}(pk, c^*) \to 1 : (pk, sk) \leftarrow Gen; c^* \leftarrow S(pk)] \end{array} \right|$$

  is negligible.

Definition 6 (Key encapsulation). *A key encapsulation mechanism KEM consists of three algorithms Gen, Encaps and Decaps.*

- $Gen(1^\lambda) \to (pk, sk)$*: the key generation algorithm Gen outputs a key pair* (pk, sk)*. Usually, for brevity, we will omit the input of Gen.*
- $Encaps(pk) \to (K, c)$*: the encapsulation algorithm Encaps, on input pk, outputs a tuple* (K, c)*, where* $K \in \mathcal{K}$ *and ciphertext c is said to be an encapsulation of the key* K*. If necessary, we make the used randomness of encapsulation explicit by writing* $(K, c) := Encaps(pk; r)$*, where* $r \in \mathcal{R}$ *(*$\mathcal{R}$ *is the randomness space).*

– *$Decaps(sk, c) \to K$: the deterministic decapsulation algorithm Decaps, on input sk and an encapsulation c, outputs either a key $K := Decaps(sk, c) \in \mathcal{K}$ or a rejection symbol $\perp \notin \mathcal{K}$.*

Next, we now define a security notion for KEM: indistinguishability against chosen-ciphertext attacks (IND-CCA).

Definition 7 (IND-CCA-secure KEM). *We define the* $\mathrm{IND-CCA}$ *game as in Fig. 2 and the* $\mathrm{IND-CCA}$ *advantage function of an adversary $\mathcal{A}$ against* KEM *as*

$$\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{A}) := \left|\Pr[\text{IND-CCA}^{\mathcal{A}}_{\text{KEM}} = 1] - 1/2\right|.$$

Game IND-CCA		DECAPS(sk, c)	
1 :	$(pk, sk) \leftarrow Gen$	1 :	**if** $c = c^*$
2 :	$b \xleftarrow{\$} \{0,1\}$	2 :	**return** $\perp$
3 :	$(K_0^*, c^*) \leftarrow Encaps(pk)$	3 :	**else return**
4 :	$K_1^* \xleftarrow{\$} \mathcal{K}$	4 :	$K := Decaps(sk, c)$
5 :	$b' \leftarrow \mathcal{A}^{\text{DECAPS}}(pk, c^*, K_b^*)$		
6 :	**return** $b' =? b$		

Fig. 2: IND-CCA game for KEM.

Following the work [4], we also make the convention that the number q_H of the adversarial queries to H counts the total number of times H is executed in the experiment. That is, the number of $\mathcal{A}$'s explicit queries to H plus the number of implicit queries to H made by the experiment.

3 Generic KEM constructions with explicit rejection

Using Targhi-Unruh technique [9], Hofheinz et al. [4] first presented two generic constructions of an IND-CCA-secure KEM with explicit rejection $\mathrm{QFO}^{\perp}_{m}$ and $\mathrm{QFO}^{\not\perp}_{m}$ in the QROM, by reducing the OW-CPA security of underlying PKE scheme to the IND-CCA security of resulting KEM with quartic security loss. These two constructions are widely used to achieve IND-CCA security in the NIST Round-1 KEM submissions [10]. Subsequently, Jiang et al. [14] improved above security loss to be quadratic for $\mathrm{FO}^{\not\perp}_{m}$ and $\mathrm{FO}^{\not\perp}$. For the transformation $\mathrm{U}^{\not\perp}_{m}$ in [4], Saito et al. [13] gave a tight reduction from the DS security of underlying perfectly correct DPKE to the IND-CCA security of resulting KEM. However, the proof techniques in [14, 13] are restricted to the KEM constructions with implicit rejection.

$Encaps(pk)$	$Decaps(sk, c)$
$m \xleftarrow{\$} \mathcal{M}$	Parse $c = (c_1, c_2)$
$c_1 = Enc(pk, m; G(m))$	$m' := Dec(sk, c_1)$
$c_2 = H'(m)$	**if** $Enc(pk, m'; G(m')) = c_1 \wedge H'(m') = c_2$
$c = (c_1, c_2)$	**return** $K := H(m')$
$K := H(m)$	**else return** $\perp$
return (K, c)	

Fig. 3: IND-CCA-secure KEM-I=$\mathrm{HFO}^{\perp}_{m}$[PKE,G,H,H']

$Encaps(pk)$	$Decaps(sk, c)$
$m \xleftarrow{\$} \mathcal{M}$	Parse $c = (c_1, c_2)$
$c_1 = Enc(pk, m; G(m))$	$m' := Dec(sk, c_1)$
$c_2 = H'(m)$	**if** $Enc(pk, m'; G(m')) = c_1 \wedge H'(m') = c_2$
$c = (c_1, c_2)$	**return** $K := H(m', c)$
$K := H(m, c)$	**else return** $\perp$
return (K, c)	

Fig. 4: IND-CCA-secure KEM-II=$\mathrm{HFO}^{\perp}$[PKE,G,H,H']

$Encaps(pk)$	$Decaps(sk, c)$
$m \xleftarrow{\$} \mathcal{M}$	Parse $c = (c_1, c_2)$
$c_1 = Enc(pk, m)$	$m' := Dec(sk, c_1)$
$c_2 = H'(m)$	**if** $Enc(pk, m') = c_1 \wedge H'(m') = c_2$
$c = (c_1, c_2)$	**return** $K := H(m')$
$K := H(m)$	**else return** $\perp$
return (K, c)	

Fig. 5: IND-CCA-secure KEM-III=$\mathrm{HU}^{\perp}_{m}$[DPKE,H,H']

In this section, first, we will present three generic constructions of an IND-CCA-secure KEM with explicit rejection, $\mathrm{HFO}^{\perp}_{m}$, $\mathrm{HFO}^{\perp}$ and $\mathrm{HU}^{\perp}_{m}$, corresponding the implicit ones, i.e., $\mathrm{FO}^{\not\perp}_{m}$, $\mathrm{FO}^{\not\perp}$ and $\mathrm{U}^{\not\perp}_{m}$ in [4, 14, 13]. Then, assuming OW-CPA security of underlying PKE, we will provide security reductions for $\mathrm{HFO}^{\perp}_{m}$ and $\mathrm{HFO}^{\perp}$ with quadratic security loss. Particularly, we also present tighter security reductions for $\mathrm{HFO}^{\perp}_{m}$ and $\mathrm{HFO}^{\perp}$ with the IND-CPA security assumption of underlying PKE. Finally, we will give a tight security reduction

for $\mathrm{HU}_m^{\perp}$, from the DS security of underlying perfectly correct DPKE to the IND-CCA security of resulting KEM.

To a public-key encryption scheme PKE = (*Gen*, *Enc*, *Dec*) with message space $\mathcal{M}$ and randomness space $\mathcal{R}$, hash functions $G : \mathcal{M} \to \mathcal{R}$, $H : \{0,1\}^* \to \{0,1\}^n$ and $H' : \mathcal{M} \to \{0,1\}^{n'}$[11], we associate KEM-I=$\mathrm{HFO}_m^{\perp}$[PKE,G,H,H'], KEM-II=$\mathrm{HFO}^{\perp}$[PKE,G,H,H'], and KEM-III=$\mathrm{HU}^{\perp}$[PKE,H,H'], shown[12] in Fig. 3, Fig. 4 and Fig. 5, respectively. To make the presentation concise, we make the convention that $\mathcal{K} = \{0,1\}^n$.

Remark: Explicit (implicit resp.) rejection[13] means a rejection symbol $\perp$ (pseudorandom key, resp.) is returned for an invalid ciphertext, of which the definition can be specified by a concrete KEM construction. For KEM-I and KEM-II (KEM-III, resp.), we say a ciphertext $c = (c_1, c_2)$ is *invalid* if $(c_1, c_2) \neq (Enc(pk, m'; G(m')), H'(m'))$ ($(c_1, c_2) \neq (Enc(pk, m'), H'(m'))$, resp.), where $m' = Dec(sk, c_1)$.

Theorem 1 (PKE IND-CPA $\overset{QROM}{\Rightarrow}$ KEM-I IND-CCA). *If* PKE *is* δ-correct*, for any IND-CCA adversary* $\mathcal{B}$ *against KEM-I, issuing at most* q_D *queries to the decapsulation oracle* Decaps*, at most* q_G (q_H, $q_{H'}$) *queries to the random oracle* G (H, H')*, there exists an IND-CPA adversary* $\mathcal{A}$ *against* PKE *such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}) \leq 2\sqrt{2(q_G + q_H + 1)\mathtt{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A}) + 4\frac{(q_G+q_H+1)^2}{|\mathcal{M}|}} + 4q_G\sqrt{\delta} + \frac{q_D}{2^{n'}}$ *and the running time of* $\mathcal{A}$ *is about that of* $\mathcal{B}$.

Proof. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-I, issuing at most q_D queries to the decapsulation oracle Decaps, at most q_G (q_H, $q_{H'}$) queries to the random oracle G (H, H'). Denote Ω_G, Ω_H, $\Omega_{H'}$, Ω_{H_q} and $\Omega_{H'_q}$ as the sets of all functions $G : \mathcal{M} \to \mathcal{R}$, $H : \mathcal{M} \to \{0,1\}^n$, $H' : \mathcal{M} \to \{0,1\}^{n'}$, $H_q : \mathcal{C}_1 \to \{0,1\}^n$ and $H'_q : \mathcal{C}_1 \to \{0,1\}^{n'}$, respectively, where $\mathcal{C}_1$ is the ciphertext space of underlying PKE scheme. Consider the games in Fig. 6.

Game G_0. Since game G_0 is exactly the IND-CCA game,

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - 1/2\right| = \mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}).$$

Given (pk, sk) and $m \in \mathcal{M}$, define "bad" randomness $\mathcal{R}_{\text{bad}}(pk, sk, m)$ and "good" randomness $\mathcal{R}_{\text{good}}(pk, sk, m) = \mathcal{R} \backslash \mathcal{R}_{\text{bad}}(pk, sk, m)$, where $\mathcal{R}_{\text{bad}}(pk, sk, m) = \{r \in \mathcal{R} : Dec(sk, Enc(pk, m; r)) \neq m\}$. Let

$$\delta(pk, sk, m) = \frac{|\mathcal{R}_{\text{bad}}(pk, sk, m)|}{|\mathcal{R}|}$$

[11] We assume that G, H, H' are not used in the algorithms of PKE, including *Gen*, *Enc* and *Dec*.

[12] The key generation algorithms *Gen* in KEM-I, KEM-II and KEM-III are the same as the ones in corresponding underlying PKEs.

[13] There may exist some KEMs with neither explicit nor implicit rejection.

as the fraction of bad randomness and $\delta(pk, sk) = \max_{m \in \mathcal{M}} \delta(pk, sk, m)$. Thus, $\delta = \mathbf{E}[\delta(pk, sk)]$, where the expectation is taken over $(pk, sk) \leftarrow Gen$.

Let G' be a random function such that $G'(m)$ is sampled according to the uniform distribution in $\mathcal{R}_{\text{good}}(pk, sk, m)$. Let $\Omega_{G'}$ be the set of all functions G'.

GAME G_1. In game G_1, we replace G by G' that uniformly samples from "good" randomness at random, i.e., $G' \xleftarrow{\$} \Omega_{G'}$. First, let's show that any adversary distinguishing G_0 from G_1 can be converted into an adversary distinguishing G from G' in the following way.

GAMES $G_0 - G_8$

$(pk, sk) \leftarrow Gen; G \xleftarrow{\$} \Omega_G$

$G' \xleftarrow{\$} \Omega_{G'}; G := G' \quad //G_1 - G_3$

$g(\cdot) = Enc(pk, \cdot; G(\cdot))$

$H \xleftarrow{\$} \Omega_H; H' \xleftarrow{\$} \Omega_{H'} \quad //G_0 - G_1$

$H_q \xleftarrow{\$} \Omega_{H_q}; H'_q \xleftarrow{\$} \Omega_{H'_q}$

$m^* \xleftarrow{\$} \mathcal{M}$

$r^* := G(m^*)$

$r^* \xleftarrow{\$} \mathcal{R} \qquad //G_6 - G_8$

$c_1^* := Enc(pk, m^*; r^*) \qquad //G_0 - G_7$

$m'^* \xleftarrow{\$} \mathcal{M} \qquad //G_8$

$c_1^* := Enc(pk, m'^*; r^*) \qquad //G_8$

$c_2^* := H'(m^*) \quad //G_0 - G_1$

$c_2^* := H'_q(c_1^*) \quad //G_2 - G_8$

$c^* = (c_1^*, c_2^*)$

$k_0^* := H(m^*)$

$k_0^* \xleftarrow{\$} \mathcal{K} \qquad //G_6 - G_8$

$k_1^* \xleftarrow{\$} \mathcal{K}; b \xleftarrow{\$} \{0, 1\}$

$b' \leftarrow \mathcal{B}^{G, H, H', \text{DECAPS}}(pk, c^*, k_b^*) \quad //G_0 - G_4$

$\ddot{G} := G; \ddot{G}(m^*) \xleftarrow{\$} \mathcal{R} \qquad //G_5 - G_6$

$\ddot{H} := H; \ddot{H}(m^*) \xleftarrow{\$} \mathcal{K} \qquad //G_5 - G_6$

$g(\cdot) = Enc(pk, \cdot; \ddot{G} \backslash m^*(\cdot)) \qquad //G_5 - G_6$

$b' \leftarrow \mathcal{B}^{\ddot{G} \backslash m^*, \ddot{H} \backslash m^*, H', \text{DECAPS}}(pk, c^*, k_b^*) //G_5 - G_6$

$b' \leftarrow \mathcal{B}^{G \backslash m^*, H \backslash m^*, H', \text{DECAPS}}(pk, c^*, k_b^*) //G_7 - G_8$

return $b' =? b$

$H(m)$ $//G_2 - G_8$

return $H_q \circ g(m)$

$H'(m)$ $//G_2 - G_8$

return $H'_q \circ g(m)$

DECAPS $(c \neq c^*)$ $//G_0 - G_2$

Parse $c = (c_1, c_2)$

$m' := Dec(sk, c_1)$

if $g(m') = c_1 \wedge H'(m') = c_2$

  return $K := H(m')$

else return $\perp$

DECAPS $(c \neq c^*)$ $//G_3 - G_8$

Parse $c = (c_1, c_2)$

if $H'_q(c_1) = c_2$

  return $K := H_q(c_1)$

else return $\perp$

Fig. 6: Games G_0-G_8 for the proof of Theorem 1

Construct an adversary $B^{\widetilde{G}}(pk, sk)$ against the distinguishing problem between G and G' by taking the accessible oracle $\widetilde{G}$ as G, simulating $\mathcal{B}$'s view and outputting in the same way as G_0 and G_1. We note that for any (pk, sk) generated by Gen, if $\widetilde{G} = G$, $B^{\widetilde{G}}(pk, sk)$ perfectly simulates G_0 and $\Pr[1 \leftarrow B^G : (pk, sk)] = \Pr[G_0^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]$. If $\widetilde{G} = G'$, $B^{\widetilde{G}}(pk, sk)$ perfectly simulates G_1 and $\Pr[1 \leftarrow B^{G'} : (pk, sk)] = \Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]$.

Thus,

$$\begin{aligned} & \left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]\right| \\ = & \left|\Pr[1 \leftarrow B^G : (pk, sk)] - \Pr[1 \leftarrow B^{G'} : (pk, sk)]\right|. \end{aligned}$$

Next, we will show that any adversary distinguishing G from G' can be converted into an adversary distinguishing F_1 from F_2, where F_1 is a function such that $F_1(m)$ is sampled according to Bernoulli distribution $B_{\delta(pk,sk,m)}$, i.e., $\Pr[F_1(m) = 1] = \delta(pk, sk, m)$ and $\Pr[F_1(m) = 0] = 1 - \delta(pk, sk, m)$, and F_2 is a constant function that always outputs 0 for any input.

$A^F(pk, sk)$	$\widetilde{G}(m)$
1 : Pick a $2q_G$-wise function f	1 : **if** $F(m) = 0$
2 : $b'' \leftarrow B^{\widetilde{G}}(pk, sk)$	2 : $\quad \widetilde{G}(m) = \mathit{Sample}(\mathcal{R}_{\mathrm{good}}(pk, sk, m); f(m))$
3 : **return** b''	3 : **else**
	4 : $\quad \widetilde{G}(m) = \mathit{Sample}(\mathcal{R}_{\mathrm{bad}}(pk, sk, m); f(m))$
	5 : **return** $\widetilde{G}(m)$

Fig. 7: A^F for the proof of Theorem 1

Given any adversary $B^{\widetilde{G}}(pk, sk)$, we construct an adversary $A^F(pk, sk)$ as in Fig. 7. $\mathit{Sample}(\mathcal{Y})$ is a probabilistic algorithm that returns a uniformly distributed $y \xleftarrow{\$} \mathcal{Y}$. $\mathit{Sample}(\mathcal{Y}; f(m))$ denotes the deterministic execution of $\mathit{Sample}(\mathcal{Y})$ using explicitly given randomness $f(m)$. Note that $\widetilde{G} = G$ if $F = F_1$ and $\widetilde{G} = G'$ if $F = F_2$. Thus, for any fixed (pk, sk) generated by Gen, $\Pr[1 \leftarrow A^{F_1} : (pk, sk)] = \Pr[1 \leftarrow B^G : (pk, sk)]$ and $\Pr[1 \leftarrow A^{F_2} : (pk, sk)] = \Pr[1 \leftarrow B^{G'} : (pk, sk)]$. Conditioned on a fixed (pk, sk) we obtain by Lemma 2

$$\begin{aligned} & \left|\Pr[1 \leftarrow B^G : (pk, sk)] - \Pr[1 \leftarrow B^{G'} : (pk, sk)]\right| \\ = & \left|\Pr[1 \leftarrow A^{F_1} : (pk, sk)] - \Pr[1 \leftarrow A^{F_2} : (pk, sk)]\right| \leq 2q_G\sqrt{\delta(pk, sk)}. \end{aligned}$$

As $\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]\right|$ can be bounded by the maximum distinguishing probability between G and G' for $B^{\widetilde{G}}(pk, sk)$,

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]\right| \leq 2q_G\sqrt{\delta(pk, sk)}.$$

By averaging over $(pk, sk) \leftarrow Gen$ we finally obtain

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G \mathbf{E}[\sqrt{\delta(pk, sk)}] \leq 2q_G\sqrt{\delta}.$$

GAME G_2. In this game, replace H and H' by $H_q \circ g$ and $H'_q \circ g$ respectively, where

$$g(\cdot) = Enc(pk, \cdot; G(\cdot)).$$

Note that g in this game is an injective function since it only samples from "good" randomness. Thus, the distributions of H in G_1 and G_2 are identical. Therefore,

$$\Pr[G_1^{\mathcal{B}} \Rightarrow 1] = \Pr[G_2^{\mathcal{B}} \Rightarrow 1].$$

GAME G_3. In game G_3, the DECAPS oracle is changed that it makes no use of the secret key sk any more. When $\mathcal{B}$ queries the DECAPS oracle on $c = (c_1, c_2)$ ($c \neq c^*$), $K := H_q(c_1)$ is returned if $H'_q(c_1) = c_2$, otherwise $\perp$. Let $m' := Dec(sk, c_1)$ and consider the following three cases.

Case 1: $Enc(pk, m'; G(m')) = c_1$ and $H'(m') = c_2$. Since $H = H_q \circ g$ and $H' = H'_q \circ g$, both DECAPS oracles in G_2 and G_3 return the same value $H_q(c_1)$.

Case 2: $Enc(pk, m'; G(m')) = c_1$ and $H'(m') \neq c_2$. In this case, $H'(m') = H'_q(c_1) \neq c_2$. Therefore, both DECAPS oracles in G_2 and G_3 return $\perp$.

Case 3: $Enc(pk, m'; G(m')) \neq c_1$. In G_2, the DECAPS oracle returns $\perp$. In G_3, note that if there exists an m'' such that $Enc(pk, m''; G(m'')) = c_1$, $m'' = m'$ since G in this game only samples from "good" randomness. That is, $Enc(pk, m'; G(m')) = c_1$ which contradicts the condition $Enc(pk, m'; G(m')) \neq c_1$. Therefore, above m'' does not exist. Meantime, we also note that $\mathcal{B}$'s queries to H' can only help him get access to H'_q at $\hat{c}_1$ such that $Enc(pk, \hat{m}; G(\hat{m})) = \hat{c}_1$ for some $\hat{m}$, thus $H'_q(c_1)$ is uniformly random in $\mathcal{B}$'s view. As a result, in this case, $\Pr[H'_q(c_1) = c_2] = \frac{1}{2^{n'}}$ and the DECAPS oracle in G_3 also returns $\perp$ with probability $1 - \frac{1}{2^{n'}}$.

By the union bound, we know that G_2 and G_3 can be distinguished with probability at most $\frac{q_D}{2^n}$. That is,

$$\left|\Pr[G_2^{\mathcal{B}} \Rightarrow 1] - \Pr[G_3^{\mathcal{B}} \Rightarrow 1]\right| \leq \frac{q_D}{2^{n'}}.$$

GAME G_4. In game G_4, we switch the G that only samples from "good" randomness back to an ideal random oracle G. Then, similar to the case of G_0 and G_1, the distinguishing problem between G_3 and G_4 can also be converted to the distinguishing problem between G and G'. Using the same analysis method in bounding the difference between G_0 and G_1, we can have

$$\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - \Pr[G_4^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G\sqrt{\delta}.$$

Let $\ddot{G}$ ($\ddot{H}$) be the function that $\ddot{G}(m^*) = \dot{r}^*$ ($\ddot{H}(m^*) = \dot{k}_0^*$), and $\ddot{G} = G$ ($\ddot{H} = H$) everywhere else, where $\dot{r}^*$ and $\dot{k}_0^*$ are picked uniformly at random from $\mathcal{R}$ and $\mathcal{K}$.

GAME G_5. In game G_5, replace G and H by $\ddot{G}\backslash m^*$ and $\ddot{H}\backslash m^*$ respectively. Note that in this game for $\mathcal{B}$'s query to G (H), $\ddot{G}\backslash m^*$ ($\ddot{H}\backslash m^*$) will first query $\mathcal{O}^{SC}_{m^*}$, i.e., perform a semi-classical measurement, and then query $\ddot{G}$ ($\ddot{H}$). Let Find be the event that $\mathcal{O}^{SC}_{m^*}$ ever outputs 1 during semi-classical measurements of $\mathcal{B}'s$ queries to G and H. Note that the state after semi-classical measurements is exactly the state just before querying $\ddot{G}$ and $\ddot{H}$. Thus, if the event $\neg\mathsf{Find}$ that $\mathcal{O}^{SC}_{m^*}$ always outputs 0 happens, there will be no m^* term for the state just before querying $\ddot{G}$ and $\ddot{H}$ (that is, the amplitude corresponding to $|m^*\rangle$ will be 0) and $\mathcal{B}$ never learns the values of $G(m^*)$ and $H(m^*)$. Therefore, if $\neg\mathsf{Find}$ happens, bit b is independent of $\mathcal{B}$'s view. Hence,

$$\Pr[G_5^{\mathcal{B}} \Rightarrow 1 \wedge \neg\mathsf{Find}] = 1/2 \Pr[\neg\mathsf{Find} : G_5] = 1/2(1 - \Pr[\mathsf{Find} : G_5]).$$

Let $(G\times H)(\cdot) = (G(\cdot), H(\cdot))$, $(\ddot{G}\times\ddot{H})(\cdot) = (\ddot{G}(\cdot), \ddot{H}(\cdot))$, and $(\ddot{G}\times\ddot{H})\backslash m^*(\cdot) = (\ddot{G}\backslash m^*(\cdot), \ddot{H}\backslash m^*(\cdot))$. If one wants to make queries to G (or H) by accessing to $G \times H$, he just needs to prepare a uniform superposition of all states in the output register responding to H (or G). The number of total queries to $G \times H$ is at most $q_G + q_H$. Let $\bar{H}_q$ be the function that $\bar{H}_q(c_1^*) = \perp$ and $\bar{H}_q = H_q$ everywhere else.

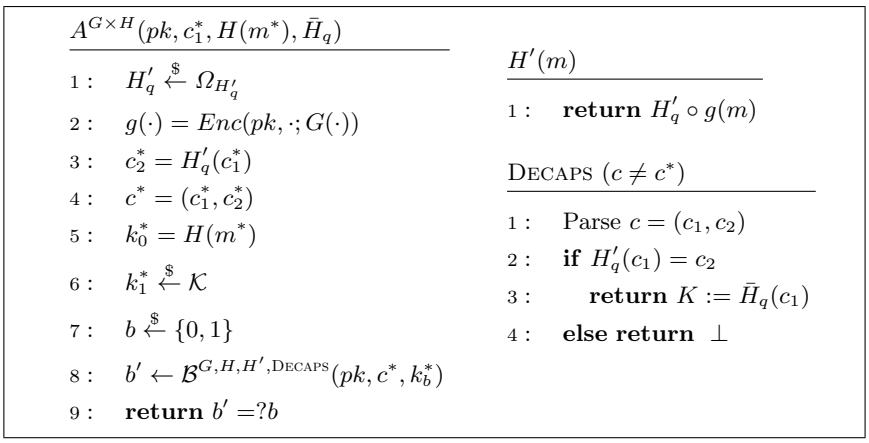

Fig. 8: $A^{G\times H}$ for the proof of Theorem 1.

Let $A^{G\times H}$ be an oracle algorithm on input $(pk, c_1^*, H(m^*), \bar{H}_q)$[14] in Fig. 8. Sample pk, m^*, G, H_q, H and c_1^* in the same way as G_4 and G_5, i.e., $(pk, sk) \leftarrow Gen$, $m^* \xleftarrow{\$} \mathcal{M}$, $G \xleftarrow{\$} \Omega_G$, $H_q \xleftarrow{\$} \Omega_{H_q}$, $H := H_q \circ g$ and $c_1^* = Enc(pk, m^*; G(m^*))$.

[14] $\bar{H}_q$ here in the input of $A^{G\times H}$ is the whole truth table of $\bar{H}_q$. One may wonder that the size of $A^{G\times H}$'s memory needs to be exponentially large. Don't worry about this. $\bar{H}_q$ is just taken as an oracle to make queries (with at most q_H times) in actual games. That is, we can also take $\bar{H}_q$ as an accessible oracle instead of a whole truth table.

Then, $A^{G\times H}$ on input $(pk, c_1^*, H(m^*), \bar{H}_q)$ perfectly simulates G_4. If we replace $G \times H$ by $(\ddot{G} \times \ddot{H})\backslash m^*$, $A^{(\ddot{G}\times\ddot{H})\backslash m^*}$ on input $(pk, c_1^*, H(m^*), \bar{H}_q)$ perfectly simulates G_5.

Applying Lemma 3 with $X = \mathcal{M}$, $Y = (\mathcal{R}, \mathcal{K})$, $S = \{m^*\}$, $\mathcal{O}_1 = G \times H$, $\mathcal{O}_2 = \ddot{G} \times \ddot{H}$ and $z = (pk, c_1^*, H(m^*), \bar{H}_q)$, we can have

$$\left|\Pr[G_4^{\mathcal{B}} \Rightarrow 1] - \Pr[G_5^{\mathcal{B}} \Rightarrow 1 \wedge \neg\mathsf{Find}]\right| \leq \sqrt{(q_G + q_H + 1)\Pr[\mathsf{Find} : G_5]}.$$

GAME G_6. In game G_6, replace $r^* := G(m^*)$ and $k_0^* := H(m^*)$ by $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$. We do not care about $\mathcal{B}$'s output, but only whether the event Find happens. Note that in G_5 and G_6, there is no information of $(G(m^*), H(m^*))$ in the oracle $(\ddot{G} \times \ddot{H})\backslash m^*$. Thus, apparently,

$$\Pr[\mathsf{Find} : G_5] = \Pr[\mathsf{Find} : G_6].$$

$\mathcal{A}(1^\lambda, pk)$

1 : $m_0 \xleftarrow{\$} \mathcal{M}$
2 : $m_1 \xleftarrow{\$} \mathcal{M}$
3 : $b'' \xleftarrow{\$} \{0,1\}$
4 : $r^* \xleftarrow{\$} \mathcal{R}$
5 : $c_1^* = Enc(pk, m_{b''}; r^*)$
6 : $c_2^* = H'_q(c_1^*)$
7 : $c^* = (c_1^*, c_2^*)$
8 : $k^* \xleftarrow{\$} \mathcal{K}$
9 : Pick a $2q_G$-wise function G
10 : Pick a $2q_H$-wise function H_q
11 : Pick a $2q_{H'}$-wise function H'_q
12 : $g(\cdot) := Enc(pk, \cdot; G(\cdot))$
13 : $b' \leftarrow \mathcal{B}^{G\backslash m_0, H\backslash m_0, H', \textsc{Decaps}}(pk, c^*, k^*)$
14 : **return** Find

$H(m)$

1 : **return** $H_q \circ g(m)$

$H'(m)$

1 : **return** $H'_q \circ g(m)$

DECAPS $(c \neq c^*)$

1 : Parse $c = (c_1, c_2)$
2 : **if** $H'_q(c_1) = c_2$
3 : **return** $K := H_q(c_1)$
4 : **else return** $\perp$

Fig. 9: Adversary $\mathcal{A}$ for the proof of Theorem 1

GAME G_7. In game G_7, replace $\ddot{G}$ and $\ddot{H}$ by G and H. Since $G(m^*)$ and $H(m^*)$ have never been used for simulating $\mathcal{B}$'s view,

$$\Pr[\mathsf{Find} : G_6] = \Pr[\mathsf{Find} : G_7].$$

GAME G_8. In game G_8, use m'^* instead of m^* for generating the challenge ciphertext, but keep using the original m^* for $G\backslash m^*$ and $H\backslash m^*$, where m'^* is chosen uniformly and independently of m^*. Note that the information of m^* in this game only exists in the oracles $G\backslash m^*$ and $H\backslash m^*$. By Lemma 4,

$$\Pr[\mathsf{Find} : G_8] \leq 4\frac{q_G + q_H + 1}{|\mathcal{M}|}.$$

Next, we show that any adversary distinguishing G_7 from G_8 can be converted into an adversary against the IND-CPA security of underlying PKE scheme. Construct an adversary $\mathcal{A}$ on input $(1^\lambda, pk)$ as in Fig. 9. Then, according to Lemma 1, if $b'' = 0$, $\mathcal{A}$ perfectly simulates G_7 and $\Pr[\mathsf{Find} : G_7] = \Pr[1 \leftarrow \mathcal{A} : b'' = 0]$. If $b'' = 1$, $\mathcal{A}$ perfectly simulates G_8 and $\Pr[\mathsf{Find} : G_8] = \Pr[1 \leftarrow \mathcal{A} : b'' = 1]$. Since $\mathtt{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A}) = 1/2\,|\Pr[1 \leftarrow \mathcal{A} : b'' = 0] - \Pr[1 \leftarrow \mathcal{A} : b'' = 1]|$,

$$|\Pr[\mathsf{Find} : G_7] - \Pr[\mathsf{Find} : G_8]| = 2\mathtt{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A}).$$

Finally, combing this with the bounds derived above, we can conclude that

$$\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}) \leq 4q_G\sqrt{\delta} + \frac{q_D}{2^{n'}} + 2\sqrt{2(q_G + q_H + 1)\mathtt{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A}) + 4\frac{(q_G + q_H + 1)^2}{|\mathcal{M}|}}.$$

□

Theorem 2 (PKE OW-CPA $\overset{QROM}{\Rightarrow}$ KEM-I IND-CCA). *If* PKE *is* δ-correct*, for any IND-CCA adversary* $\mathcal{B}$ *against KEM-I, issuing at most* q_D *queries to the decapsulation oracle* DECAPS*, at most* q_G (q_H, $q_{H'}$) *queries to the random oracle* G (H, H')*, there exists a OW-CPA adversary* $\mathcal{A}$ *against* PKE *such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}) \leq 4q_G\sqrt{\delta} + \frac{q_D}{2^{n'}} + 2(q_G + q_H) \cdot \sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})}$ *and the running time of* $\mathcal{A}$ *is about that of* $\mathcal{B}$*.*

Different from $\text{FO}^\perp_m$, $\text{HFO}^\perp_m$ adds the plaintext confirmation and adopts explicit rejection for decapsulation. In [14, Theorem 2], a security proof of $\text{FO}^\perp_m$ is given. The sole and key obstacle of applying the proof techniques in [14, Theorem 2] to $\text{HFO}^\perp_m$ is the validity verification of ciphertext when simulating the decapsulation oracle. Fortunately, this can be overcome with the same verification method used in the proof of Theorem 1. Thus, combing the proofs of Theorem 1 and [14, Theorem 2], we can obtain a proof of Theorem 2, see Appendix A.

Different from the one in KEM-I, the hash function H in KEM-II takes both the plaintext m and the ciphertext c as input. Using the same proof method in [14, Theorem 1], we can divide the H-inputs (m, c) into two categories, matched inputs and unmatched inputs, by judging whether $c = (Enc(pk, m; G(m)), H'(m))$, and replace H by $H_q \circ g$ only for the matched inputs. Then, following the proofs of Theorem 1 and Theorem 2, we can derive Theorem 3 and Theorem 4.

Theorem 3 (PKE IND-CPA $\overset{QROM}{\Rightarrow}$ KEM-II IND-CCA). *If* PKE *is* δ-correct*, for any IND-CCA adversary* $\mathcal{B}$ *against KEM-II, issuing at most* q_D

queries to the decapsulation oracle DECAPS*, at most q_G (q_H, $q_{H'}$) queries to the random oracle G (H, H'), there exists an IND-CPA adversary $\mathcal{A}$ against* PKE *such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-II}}(\mathcal{B}) \leq 2\sqrt{2(q_G+q_H+1)\mathtt{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A}) + 4\frac{(q_G+q_H+1)^2}{|\mathcal{M}|}} + 4q_G\sqrt{\delta} + \frac{q_D}{2^{n'}}$ *and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.*

Theorem 4 (PKE OW-CPA $\overset{QROM}{\Rightarrow}$ KEM-II IND-CCA). *If* PKE *is* δ-correct*, for any IND-CCA adversary $\mathcal{B}$ against KEM-II, issuing at most q_D queries to the decapsulation oracle* DECAPS*, at most q_G (q_H, $q_{H'}$) queries to the random oracle G (H, H'), there exists a OW-CPA adversary $\mathcal{A}$ against* PKE *such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-II}}(\mathcal{B}) \leq 4q_G\sqrt{\delta} + \frac{q_D}{2^{n'}} + 2(q_G+q_H)\cdot\sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})}$ *and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.*

Theorem 5 (PKE DS $\overset{QROM}{\Rightarrow}$ KEM-III IND-CCA). *If* PKE *is deterministic and perfectly correct, for any IND-CCA adversary $\mathcal{B}$ against KEM-III, issuing at most q_D queries to the decapsulation oracle* DECAPS*, at most q_H ($q_{H'}$) queries to the random oracle H (H'), there exists an adversary $\mathcal{A}$ against the DS security with an algorithm S such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-III}}(\mathcal{B}) \leq \frac{q_D}{2^{n'}} + \mathtt{Adv}^{\text{DS-IND}}_{\text{PKE},U_{\mathcal{M}},S}(\mathcal{A}) + \text{DISJ}_{\text{PKE},S}$*, where $U_{\mathcal{M}}$ is the uniform distribution in $\mathcal{M}$, and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.*

Proof. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-III, issuing at most q_D queries to DECAPS, at most q_H ($q_{H'}$) queries to H (H'). We follow the notations Ω_H, $\Omega_{H'}$, Ω_{H_q} and $\Omega_{H'_q}$ in Theorem 1. Consider the games in Fig. 10.

GAME G_0. Since game G_0 is exactly the IND-CCA game,

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - 1/2\right| = \mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-III}}(\mathcal{B}).$$

GAME G_1. Replace H and H' by $H_q \circ g$ and $H'_q \circ g$ respectively, where

$$g(\cdot) = Enc(pk, \cdot).$$

As PKE is perfectly correct, g is an injective functions. Thus, $H_q \circ g$ ($H'_q \circ g$) is also a uniformly random function as H (H') in G_0. Therefore, we can have

$$\Pr[G_0^{\mathcal{B}} \Rightarrow 1] = \Pr[G_1^{\mathcal{B}} \Rightarrow 1].$$

GAME G_2. In game G_2, the DECAPS oracle is changed that it makes no use of the secret key sk any more. When $\mathcal{B}$ queries the DECAPS oracle on $c = (c_1, c_2)$ ($c \neq c^*$), $K := H_q(c_1)$ is returned if $H'_q(c_1) = c_2$, otherwise $\perp$.

Let $m' := Dec(sk, c_1)$. Consider the following three cases.

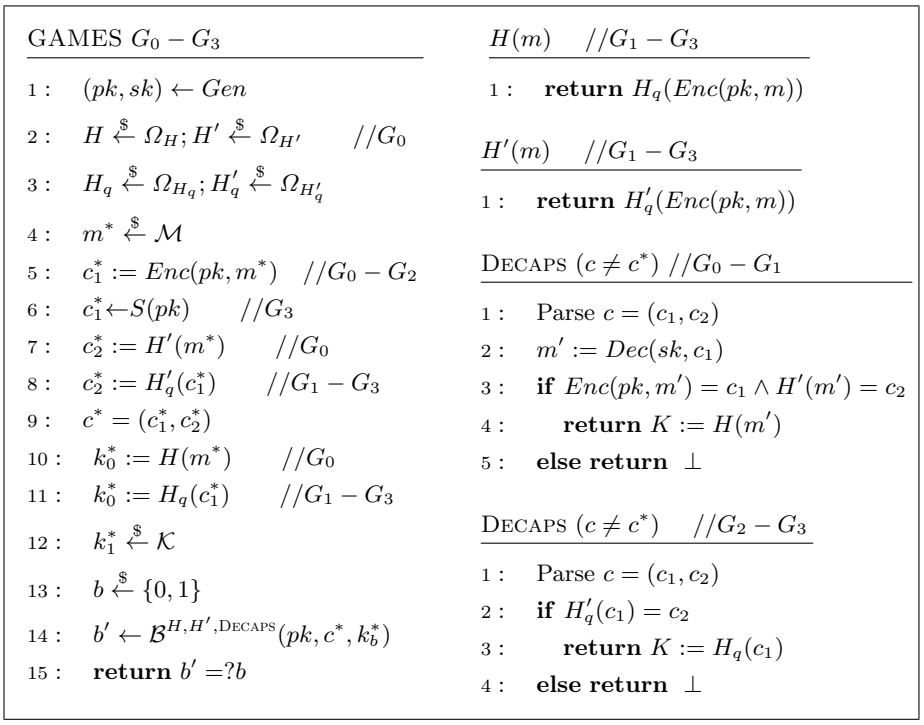

Fig. 10: Games G_0-G_3 for the proof of Theorem 3

Case 1: If $Enc(pk, m') \neq c_1$. In this case, the DECAPS oracle in G_1 returns $\perp$. We note that $\mathcal{B}$'s queries to H' can only help him get access to H'_q at $\hat{c}_1$ such that $Enc(pk, \hat{m}) = \hat{c}_1$ for some $\hat{m}$. Such a $\hat{m}$ that $Enc(pk, \hat{m}) = c_1$ does not exist due to the perfect correctness of underlying DPKE. Thus, $H'_q(c_1)$ is uniformly random in $\mathcal{B}$'s view and $H'_q(c_1) \neq c_2$ with probability $1 - \frac{1}{2^{n'}}$. Therefore, the DECAPS oracle in G_2 returns $\perp$ with probability $1 - \frac{1}{2^{n'}}$.

Case 2: $Enc(pk, m') = c_1 \wedge H'(m') = c_2$. In this case, $H'(m') = H'_q(Enc(pk, m')) = H'_q(c_1) = c_2$. Thus, both DECAPS oracles in G_1 and G_2 return the same value $H(m') = H_q \circ g(m') = H_q(c_1)$.

Case 3: $Enc(pk, m') = c_1 \wedge H'(m') \neq c_2$. In this case, $H'(m') = H'_q(Enc(pk, m')) = H'_q(c_1) \neq c_2$, both DECAPS oracles in G_1 and G_2 return $\perp$.

Therefore, the DECAPS oracles in G_1 and G_2 output different values with probability at most $\frac{1}{2^{n'}}$. By the union bound we obtain

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1]\right| \leq \frac{q_D}{2^{n'}}.$$

GAME G_3. In game G_3, c_1^* is given by $c_1^* \leftarrow S(pk)$. Then, we can bound $\left|\Pr[G_2^{\mathcal{B}} \Rightarrow 1] - \Pr[G_3^{\mathcal{B}} \Rightarrow 1]\right|$ and $\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - 1/2\right|$ as in the proof of $U_m^{\not\perp}$ [13, Theorem 4.2].

Construct an adversary $\mathcal{A}$ on input $(1^\lambda, pk, c_1^*)$ that does the following:

1. Pick a $2q_H$-wise ($2q_{H'}$-wise) independent function uniformly at random and use it to simulate the random oracle H_q (H'_q). The random oracle H (H') is simulated by $H_q \circ g$ ($H'_q \circ g$), where $g(\cdot) = Enc(pk, \cdot)$.
2. Let $c_2^* = H'_q(c_1^*)$, $c^* = (c_1^*, c_2^*)$, $k_0^* = H_q(c_1^*)$, $k_1^* \xleftarrow{\$} \mathcal{K}$ and $b \xleftarrow{\$} \{0,1\}$.
3. Answer the decapsulation queries by using the DECAPS oracle as in G_2 and G_3.
4. Invoke $b' \leftarrow \mathcal{B}^{H,H',\text{DECAPS}}(pk, c^*, k_b^*)$.
5. Return $b' =? b$.

Obviously, $\mathcal{A}$ perfectly simulates G_2 if $c_1^* = Enc(pk, m^*)$ ($m^* \xleftarrow{\$} \mathcal{M}$) and G_3 if $c_1^* \leftarrow S(pk)$. Therefore,

$$\left|\Pr[G_2^{\mathcal{B}} \Rightarrow 1] - \Pr[G_3^{\mathcal{B}} \Rightarrow 1]\right| \leq \mathtt{Adv}^{\text{DS-IND}}_{\text{PKE},U_{\mathcal{M}},S}(\mathcal{A}),$$

Let Bad be the event that $c_1^* \in Enc(pk, \mathcal{M})$ in G_3. Then, $\Pr[\mathsf{Bad}] \leq \text{DISJ}_{\text{PKE},S}$. We note that if $\neg\mathsf{Bad}$ happens, $H_q(c_1^*)$ is uniformly random in $\mathcal{B}$'s view since queries to H can only reveal $H_q(c)$ for $c \in Enc(pk, \mathcal{M})$. Therefore, $\Pr[G_3^{\mathcal{B}} \Rightarrow 1 : \neg\mathsf{Bad}] = 1/2$. We also note that $\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - 1/2\right| \leq \Pr[\mathsf{Bad}] + \left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1 : \neg\mathsf{Bad}] - 1/2\right|$. Thus,

$$\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - 1/2\right| \leq \text{DISJ}_{\text{PKE},S}.$$

Combing the above bounds, Theorem 5 is proven. □

Acknowledgements. We would like to thank anonymous reviews of PKC 2019 for their insightful comments. In particular, we are also grateful to Chris Brzuska for his kind suggestions which are helpful in improving our paper. This work is supported by the National Key Research and Development Program of China (No. 2017YFB0802000), the National Natural Science Foundation of China (No. U1536205, 61472446, 61701539), and the National Cryptography Development Fund (mmjj20180107, mmjj20180212).

References

1. Rackoff, C., Simon, D.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Feigenbaum, J., ed.: Advances in Cryptology – CRYPTO 1991. Volume 576 of LNCS., Springer (1992) 433–444
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V., eds.: Proceedings of the 1st ACM Conference on Computer and Communications Security – CCS 1993, ACM (1993) 62–73
3. Dent, A.W.: A designer's guide to KEMs. In Paterson, K.G., ed.: Cryptography and Coding: 9th IMA International Conference. Volume 2898 of LNCS., Springer-Verlag (2003) 133–151
4. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In Kalai, Y., Reyzin, L., eds.: Theory of Cryptography - 15th International Conference – TCC 2017. Volume 10677 of LNCS., Springer (2017) 341–371

5. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In Wiener, M.J., ed.: Advances in Cryptology – CRYPTO 1999. Volume 99 of LNCS., Springer (1999) 537–554
6. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of cryptology **26**(1) (2013) 1–22
7. Okamoto, T., Pointcheval, D.: REACT: Rapid enhanced-security asymmetric cryptosystem transform. In Naccache, D., ed.: Topics in Cryptology – CT-RSA 2001. Volume 2020 of LNCS., Springer (2001) 159–174
8. Jean-Sébastien, C., Handschuh, H., Joye, M., Paillier, P., Pointcheval, D., Tymen, C.: GEM: A generic chosen-ciphertext secure encryption method. In Preneel, B., ed.: Topics in Cryptology – CT-RSA 2002. Volume 2271 of LNCS., Springer (2002) 263–276
9. Targhi, E.E., Unruh, D.: Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Hirt, M., Smith, A.D., eds.: Theory of Cryptography Conference – TCC 2016-B. Volume 9986 of LNCS., Springer (2016) 192–216
10. NIST: National institute for standards and technology. Post quantum crypto project (2017) `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.
11. Boneh, D., Dagdelen, O., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In Lee, D.H., Wang, X., eds.: Advances in Cryptology – ASIACRYPT 2011. Volume 7073 of LNCS., Springer (2011) 41–69
12. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In Oswald, E., Fischlin, M., eds.: Advances in Cryptology – EUROCRYPT 2015. Volume 9057 of LNCS., Springer (2015) 755–784
13. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Nielsen, J.B., Rijmen, V., eds.: Advances in Cryptology – EUROCRYPT 2018. Volume 10822 of LNCS. (2018) 520–551
14. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Shacham, H., Boldyreva, A., eds.: Advances in Cryptology – CRYPTO 2018. Volume 10993 of LNCS. (2018) 96–125
15. Hamburg, M.: Module-LWE: The three bears. Technical report, https://www.shiftleft.org/papers/threebears/
16. Hülsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P.: High-speed key encapsulation from NTRU. In Fischer, W., Homma, N., eds.: Cryptographic Hardware and Embedded Systems – CHES 2017. Volume 10529 of LNCS., Springer-Verlag (2017) 232–252
17. Bernstein, D.J., Persichetti, E.: Towards KEM unification. Cryptology ePrint Archive, Report 2018/526 (2018) `https://eprint.iacr.org/2018/526`.
18. Unruh, D.: Revocable quantum timed-release encryption. Journal of the ACM **62**(6) (2015) 49:1–49:76
19. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. Cryptology ePrint Archive, Report 2018/904 (2018) `https://eprint.iacr.org/2018/904`.
20. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. Cryptology ePrint Archive, Report 2018/276 (2018) `https://eprint.iacr.org/2018/276`.
21. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In Canetti, R., Garay, J.A., eds.: Advances in Cryptology – CRYPTO 2013. Volume 8043 of LNCS., Springer (2013) 361–379

22. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Number 2. Cambridge University Press (2000)
23. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In Safavi-Naini, R., Canetti, R., eds.: Advances in Cryptology – CRYPTO 2012. Volume 7417 of LNCS., Springer (2012) 758–775
24. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: 55th IEEE Annual Symposium on Foundations of Computer Science – FOCS 2014, IEEE (2014) 474–483
25. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In Cheng, C., Chung, K., Persiano, G., Yang, B., eds.: Public-Key Cryptography – PKC 2016. Volume 9614 of LNCS., Springer (2016) 387–416

A Proof of Theorem 2

Proof. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-I, issuing at most q_D queries to the decapsulation oracle DECAPS, at most q_G (q_H, $q_{H'}$) queries to the random oracle G (H, H'). Follow the same notations Ω_G, Ω_H, $\Omega_{H'}$, Ω_{H_q}, $\Omega_{H'_q}$, $\Omega_{G'}$ and $\mathcal{C}_1$ as in the proof of Theorem 1. Consider the games in Fig. 11 and Fig. 13.

GAME G_0. Since game G_0 is exactly the IND-CCA game,

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - 1/2\right| = \mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}).$$

GAME G_1. In game G_1, we replace G by G' that uniformly samples from "good" randomness at random, i.e., $G' \xleftarrow{\$} \Omega_{G'}$.

GAME G_2. In this game, replace H and H' by $H_q \circ g$ and $H'_q \circ g$ respectively, where

$$g(\cdot) = Enc(pk, \cdot; G(\cdot)).$$

GAME G_3. In game G_3, the DECAPS oracle is changed that it makes no use of the secret key sk any more. When $\mathcal{B}$ queries the DECAPS oracle on $c = (c_1, c_2)$ ($c \neq c^*$), $K := H_q(c_1)$ is returned if $H'_q(c_1) = c_2$, otherwise $\perp$.

GAME G_4. In game G_4, we switch the G that only samples from "good" randomness back to an ideal random oracle G.

Using the same analysis as in the proof of Theorem 1, we can have

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_4^{\mathcal{B}} \Rightarrow 1]\right| \leq 4q_G\sqrt{\delta} + \frac{q_D}{2^{n'}}.$$

Let $\ddot{G}$ ($\ddot{H}$) be the function that $\ddot{G}(m^*) = \dot{r}^*$ ($\ddot{H}(m^*) = \dot{k}_0^*$), and $\ddot{G} = G$ ($\ddot{H} = H$) everywhere else, where $\dot{r}^*$ and $\dot{k}_0^*$ are picked uniformly at random from $\mathcal{R}$ and $\mathcal{K}$.

GAMES $G_0 - G_5$

1 : $(pk, sk) \leftarrow Gen; G \xleftarrow{\$} \Omega_G$
2 : $H_q \xleftarrow{\$} \Omega_{H_q}; H'_q \xleftarrow{\$} \Omega_{H'_q}$
3 : $G' \xleftarrow{\$} \Omega_{G'}; G := G' \quad //G_1 - G_3$
4 : $g(\cdot) = Enc(pk, \cdot; G(\cdot))$
5 : $H \xleftarrow{\$} \Omega_H; H' \xleftarrow{\$} \Omega_{H'} \quad //G_0 - G_1$
6 : $m^* \xleftarrow{\$} \mathcal{M}$
7 : $c_1^* := Enc(pk, m^*; G(m^*))$
8 : $c_2^* := H'(m^*) \quad //G_0 - G_1$
9 : $c_2^* := H'_q(c_1^*) \quad //G_2 - G_5$
10 : $c^* = (c_1^*, c_2^*)$
11 : $k_0^* := H(m^*)$
12 : $k_1^* \xleftarrow{\$} \mathcal{K}$
13 : $b \xleftarrow{\$} \{0,1\}$
14 : $b' \leftarrow \mathcal{B}^{G,H,H',\textsc{Decaps}}(pk, c^*, k_b^*) //G_0 - G_4$
15 : $\ddot{G} := G; \ddot{G}(m^*) \xleftarrow{\$} \mathcal{R} \quad //G_5$
16 : $\ddot{H} := H; \ddot{H}(m^*) \xleftarrow{\$} \mathcal{K} \quad //G_5$
17 : $g(\cdot) = Enc(pk, \cdot; \ddot{G}(\cdot) \quad //G_5$
18 : $b' \leftarrow \mathcal{B}^{\ddot{G},\ddot{H},H',\textsc{Decaps}}(pk, c^*, k_b^*) //G_5$
19 : **return** $b' =? b$

$H(m)$ $//G_2 - G_5$

1 : **return** $H_q(g(m))$

$H'(m)$ $//G_2 - G_5$

1 : **return** $H'_q(g(m))$

DECAPS $(c \neq c^*)$ $//G_0 - G_2$

1 : Parse $c = (c_1, c_2)$
2 : $m' := Dec(sk, c_1)$
3 : **if** $g(m') = c_1 \wedge H'(m') = c_2$
4 : **return** $K := H(m')$
5 : **else return** $\perp$

DECAPS $(c \neq c^*)$ $//G_3 - G_5$

1 : Parse $c = (c_1, c_2)$
2 : **if** $H'_q(c_1) = c_2$
3 : **return** $K := H_q(c_1)$
4 : **else return** $\perp$

Fig. 11: Games G_0-G_5 for the proof of Theorem 2

$A^{G \times H}(pk, c_1^*, H(m^*), \bar{H}_q)$

1 : $H'_q \xleftarrow{\$} \Omega_{H'_q}$
2 : $g(\cdot) = Enc(pk, \cdot; G(\cdot))$
3 : $c_2^* = H'_q(c_1^*)$
4 : $c^* = (c_1^*, c_2^*)$
5 : $k_0^* = H(m^*)$
6 : $k_1^* \xleftarrow{\$} \mathcal{K}$
7 : $b \xleftarrow{\$} \{0,1\}$
8 : $b' \leftarrow \mathcal{B}^{G,H,H',\textsc{Decaps}}(pk, c^*, k_b^*)$
9 : **return** $b' =? b$

$H'(m)$

1 : **return** $H'_q \circ g(m)$

DECAPS $(c \neq c^*)$

1 : Parse $c = (c_1, c_2)$
2 : **if** $H'_q(c_1) = c_2$
3 : **return** $K := \bar{H}_q(c_1)$
4 : **else return** $\perp$

Fig. 12: $A^{G \times H}$ for the proof of Theorem 2.

GAMES G_6

$i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$
$(pk, sk) \leftarrow Gen; G \xleftarrow{\$} \Omega_G$
$H_q \xleftarrow{\$} \Omega_{H_q}; H'_q \xleftarrow{\$} \Omega_{H'_q}$
$H(\cdot) = H_q(Enc(pk, \cdot; G(\cdot)))$
$m^* \xleftarrow{\$} \mathcal{M}$
$r^* \xleftarrow{\$} \mathcal{R}$
$\ddot{G} := G; \ddot{G}(m^*) = r^*$
$g(\cdot) := Enc(pk, \cdot; \ddot{G}(\cdot))$
$c_1^* = Enc(pk, m^*; G(m^*))$
$c_2^* = H'_q(c_1^*)$
$c^* = (c_1^*, c_2^*)$
$k^* \xleftarrow{\$} \mathcal{K}$
$\ddot{H} := H; \ddot{H}(m^*) = k^*$
run $\mathcal{B}^{\ddot{G}, \ddot{H}, H', \textsc{Decaps}}(pk, c^*, H(m^*))$
until the $i-$th query to $\ddot{G} \times \ddot{H}$
measure the argument $\hat{m}$
return $\hat{m} =? m^*$

DECAPS $(c \neq c^*)$

Parse $c = (c_1, c_2)$
if $H'_q(c_1) = c_2$
  return $K := H_q(c_1)$
else return $\perp$

GAMES G_7

$i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$
$(pk, sk) \leftarrow Gen; G \xleftarrow{\$} \Omega_G$
$H_q \xleftarrow{\$} \Omega_{H_q}; H'_q \xleftarrow{\$} \Omega_{H_q}$
$H(\cdot) = H_q(Enc(pk, \cdot; G(\cdot)))$
$m^* \xleftarrow{\$} \mathcal{M}$
$r^* \xleftarrow{\$} \mathcal{R}$
$g(\cdot) := Enc(pk, \cdot; G(\cdot))$
$c_1^* = Enc(pk, m^*; r^*)$
$c_2^* = H'_q(c_1^*)$
$c^* = (c_1^*, c_2^*)$
$k^* \xleftarrow{\$} \mathcal{K}$
run $\mathcal{B}^{G, H, H', \textsc{Decaps}}(pk, c^*, k^*)$
until the $i-$th query to $G \times H$
measure the argument $\hat{m}$
return $\hat{m} =? m^*$

$H'(m)$

return $H'_q \circ g(m)$

Fig. 13: Game G_6 and game G_7 for the proof of Theorem 2

GAME G_5. In game G_5, replace G and H by $\ddot{G}$ and $\ddot{H}$ respectively. In this game, bit b is independent of $\mathcal{B}$'s view. Hence,

$$\Pr[G_5^{\mathcal{B}} \Rightarrow 1] = 1/2.$$

Let $(G \times H)(\cdot) = (G(\cdot), H(\cdot))$ and $(\ddot{G} \times \ddot{H})(\cdot) = (\ddot{G}(\cdot), \ddot{H}(\cdot))$. Let $\bar{H}_q$ be the function that $\bar{H}_q(c_1^*) = \perp$ and $\bar{H}_q = H_q$ everywhere else. Define $A^{G \times H}$ as in Fig. 12. Sample pk, m^*, G, H_q, H and c_1^* in the same way as G_4 and G_5, i.e., $(pk, sk) \leftarrow Gen$, $m^* \xleftarrow{\$} \mathcal{M}$, $G \xleftarrow{\$} \Omega_G$, $H_q \xleftarrow{\$} \Omega_{H_q}$, $H := H_q \circ g$ and $c_1^* = Enc(pk, m^*; G(m^*))$, where $g(\cdot) = Enc(pk, \cdot; G(\cdot))$.

Then, $A^{G \times H}$ on input $(pk, c_1^*, H(m^*), \bar{H}_q)$ perfectly simulates G_4. If we replace $G \times H$ by $\ddot{G} \times \ddot{H}$, $A^{\ddot{G} \times \ddot{H}}$ on input $(pk, c_1^*, H(m^*), \bar{H}_q)$ perfectly simulates G_5.

Let $B^{\ddot{G}\times\ddot{H}}$ be an oracle algorithm that on input $(pk, c_1^*, H(m^*), \bar{H}_q)$ does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$, run $A^{\ddot{G}\times\ddot{H}}(pk, c_1^*, H(m^*), \bar{H}_q)$ until the i-th query, measure the argument of the query in the computational basis, output the measurement outcome. Define game G_6 as in Fig. 13.

Applying Lemma 5 with $X = \mathcal{M}$, $Y = (\mathcal{R}, \mathcal{K})$, $S = \{m^*\}$, $\mathcal{O}_1 = \ddot{G} \times \ddot{H}$, $\mathcal{O}_2 = G \times H$ and $z = (pk, c_1^*, H(m^*), \bar{H}_q)$, we can have

$$\left|\Pr[G_4^{\mathcal{B}} \Rightarrow 1] - \Pr[G_5^{\mathcal{B}} \Rightarrow 1]\right| \leq 2(q_G + q_H)\sqrt{\Pr[G_6^{\mathcal{B}} \Rightarrow 1]}.$$

Rearrange game G_6 into game G_7, see Fig. 13. Clearly, $\Pr[G_6^{\mathcal{B}} \Rightarrow 1] = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$. Then, we construct an adversary $\mathcal{A}$ against the OW-CPA security of PKE such that $\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A}) = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$. The adversary $\mathcal{A}$ on input $(1^\lambda, pk, c_1^*)$ does the following:

1. Run the adversary $\mathcal{B}$ in game G_7.
2. Pick a $2q_G$ ($2q_H$, $2q_{H'}$)-wise independent function uniformly at random and use it to simulate the random oracle G (H_q, H'_q). The random oracle H (H') is simulated by $H_q \circ g$ ($H'_q \circ g$). Use $G \times H$ to answer $\mathcal{B}$'s queries to both G and H.
3. Let $c_2^* = H'_q(c_1^*)$ and $c^* = (c_1^*, c_2^*)$.
4. Answer the decapsulation queries by using the DECAPS oracle as in Fig. 13.
5. Select $k^* \xleftarrow{\$} \mathcal{K}$ and respond to $\mathcal{B}$'s challenge query with (c^*, k^*).
6. Select $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$, measure the argument $\hat{m}$ of the i-th query to $G \times H$ and output $\hat{m}$.

It is obvious that $\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A}) = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$. Combing this with the bounds derived above, we can conclude that

$$\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}) \leq 4q_G \cdot \sqrt{\delta} + \frac{q_D}{2^{n'}} + 2(q_H + q_G) \cdot \sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})}.$$

□