Post-quantum IND-CCA-secure KEM without Additional Hash

Haodong Jiang[1,2], Zhenfeng Zhang[2,3], Long Chen[2,3], Hong Wang[1], and Zhi Ma[1,4]

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, Henan, China
[2] Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] CAS Center for Excellence and Synergetic Innovation Center in Quantum information and Quantum Physics,USTC, Hefei, Anhui, China
hdjiang13@gmail.com, {chenlong,zfzhang}@tca.iscas.ac.cn

Abstract. With the gradual progress of NIST's post-quantum cryptography standardization, several practical post-quantum secure key encapsulation mechanism (KEM) schemes have been proposed. Generally, an IND-CCA-secure KEM is usually achieved by introducing an IND-CPA-secure (or OW-CPA-secure) public-key encryption (PKE) scheme, then applying some generic transformations to it. All these generic transformations are constructed in the random oracle model (ROM). To fully assess the post-quantum security, security analysis in the quantum random oracle model (QROM) is preferred. However, current works either lacked a QROM security proof or just followed Targhi and Unruh's proof technique (TCC-B 2016) and modified the original transformations by adding an additional hash to the ciphertext to achieve the QROM security.

In this paper, by using a novel proof technique, we present QROM security reductions for two widely used generic transformations without suffering any ciphertext overhead. Meanwhile, the security bounds are much tighter than the ones derived by utilizing Targhi and Unruh's proof technique. Thus, our QROM security proofs not only provide a solid post-quantum security guarantee for previous KEM schemes, but also simplify the constructions and reduce the ciphertext sizes. We also provide QROM security reductions for Hofheinz-Hövelmanns-Kiltz modular transformations (TCC 2017), which can help to obtain a variety of combined transformations with different requirements and properties.

Keywords: quantum random oracle model · key encapsulation mechanism · IND-CCA security · generic transformation

1 Introduction

In December 2016, 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, public-key encryption (PKE), and key encapsulation mechanism (KEM) (or key exchange) [1]. Triggered by that, there has been a rapid growth of interest in post-quantum cryptographic schemes.

As a foundational cryptography primitive, KEM is efficient and versatile. It can be used to construct, in a black-box manner, PKE (the KEM-DEM paradigm [2]), key exchange and authenticated key exchange [3, 4]. Compared with designing a full PKE scheme, the KEM construction is usually somewhat easier or more efficient. Recently, to make the KEM scheme secure against quantum computers, many researchers devoted to the KEM constructions based on the hardness of certain problems over lattices [5–11] and code theory [12, 13].

Indistinguishability against chosen-ciphertext attacks (IND-CCA) [14] is widely accepted as the standard security notion for many cryptography applications. However, the security is usually much more difficult to prove than IND-CPA (indistinguishability against chosen-plaintext attacks) security. Mostly, generic transformations [15, 16] are used to create an IND-CCA-secure KEM from some weakly secure (OW-CPA or IND-CPA) PKEs, see [5–9, 12].

In his "A Designer's Guide to KEMs" paper [15], Dent provided several generic transformations from weakly secure PKE schemes to IND-CCA-secure KEMs. In particular, [15, Table 5] can be viewed as the KEM variant of Fujisaki-Okamoto (FO) transformation [17, 18], and is widely used in constructing post-quantum IND-CCA-secure KEMs, e.g., [6–8]. Recently, considering the drawbacks of previous analysis of FO transformation, such as a non-tight security reduction and the need for a perfectly correct scheme, Hofheinz, Hövelmanns and Kiltz [16] revisited the FO transformation and provided a fine-grained and modular toolkit of transformations. By combining these modular transformations, they obtained several variants of FO transformation. Subsequently, Bos et al. [5] and Barreto et al. [12] used one of these variants to construct IND-CCA-secure KEMs, Kyber (module-lattice-based) and CAKE (code-based), respectively. Specially, Kyber is part of the CRYSTALS (Cryptographic Suite for Algebraic Lattices) package that will be submitted to the NIST call for post-quantum standards.

Note that all above mentioned transformations are constructed in the random oracle model (ROM) [19]. When the KEM scheme is instantiated, the random oracle is usually replaced by a hash function, which a quantum adversary may evaluate on a quantum superposition of inputs. As a result, to fully assess post-quantum security, we should analyze security in the quantum random oracle model (QROM), as introduced in [20]. However, proving security in the QROM is quite challenging, as many classical ROM proof techniques will be invalid.

Among current works about post-quantum KEMs, they either lacked a QROM security proof [6, 8, 10] or just followed Targhi and Unruh's proof idea [21, 22] and modified the original transformations by adding an additional hash to the ciphertext to achieve QROM security [5, 7, 11, 12, 16]. Intuitively, for 128-bit post-quantum security, such a modification merely increases the ciphertext size by 256 bits [23]. However, we note that the QROM security proof in [21, 22] requires the additional hash function to be length-preserving (that has the same domain

and range size). Thus, for some schemes where the message space is strictly larger than the output space of the hash function, the increasement of the ciphertext size is significant. Hülsing et al. [7] tried several ways to circumvent this issue, unfortunately all straight forward approaches failed. For their specific NTRU-based KEM, additional 1128 bits are needed in the decapsulation, which accounts for 11% of the final encapsulation size.

In the ROM, this additional hash is clearly redundant for the constructions of IND-CCA-secure KEM [15, 16]. To use Targhi and Unruh's proof technique to accomplish the QROM security proof, [5, 7, 11, 12, 16] deliberately introduced an additional length-preserving hash to the ciphertext, which increased the ciphertext size and complicated the implementation. Thus, a natural question is that: can we improve the QROM security proof without suffering any ciphertext overhead for these constructions? In this paper, we present a positive answer.

1.1 Our Contributions

1. We prove the QROM security of two generic transformations (variants of FO transformation) by reducing the IND-CCA security of KEM to the OW-CPA security of the underlying PKE. One is the transformation $\mathrm{FO}^{\not\perp}$ in [16], we denote such a construction by FO-I in our paper. In [16], Hofheinz et al. proved the security of FO-I in the ROM. When considering the QROM security, they followed Targhi and Unruh's proof idea, and modified FO-I by adding an additional hash to the ciphertext. Kyber [5] and CAKE [12] were exactly constructed by using this modified transformation. Thus, with our security proof, Kyber and CAKE can be simplified by cutting off that additional hash, leading to performance improvement in terms of speed and sizes.
   The other is $\mathrm{FO}_m^{\not\perp}$ in [16], the transformation [15, Table 5] with implicit rejection (meaning that a pseudorandom key is returned when an invalid ciphertext is submitted to the decapsulation algorithm). We denote this transformation by FO-II in our paper. This transformation was widely used in [6–8]. But, these works either lacked a QROM security proof [6, 8] or just followed Targhi and Unruh's work [22] and modified the original transformation by adding an additional hash to the ciphertext to achieve the QROM security [7]. Thus, our QROM security proof provides a solid post-quantum security guarantee for these KEM schemes without additional ciphertext overhead.
2. For our security reductions, the advantage of the adversary $\mathcal{B}$ against the IND-CCA security of KEM $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{B})$ is approximately bounded by $q \cdot \sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})}$, which is much tighter than $q^{\frac{3}{2}} \cdot [\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})]^{\frac{1}{4}}$ achieved by [5, 12, 16], where $\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})$ is the advantage of the adversary $\mathcal{A}$ against the OW-CPA security of PKE and q is the total number of $\mathcal{B}$'s queries to various oracles.
3. We provide QROM security reductions for some fine-grained and modular transformations in [16]. Hofheinz et al. [16] provided seven fine-grained modular transformations T, $U^{\not\perp}$, $U^{\perp}$, $U_m^{\not\perp}$, $U_m^{\perp}$, $QU_m^{\not\perp}$ and $QU_m^{\perp}$, which can be

used to obtain some combined transformations with different requirements and properties. But, they just presented QROM security proofs for the transformations T, $QU_m^{\not\perp}$ and $QU_m^{\perp}$. Different from $U^{\not\perp}$, $U^{\perp}$, $U_m^{\not\perp}$ and $U_m^{\perp}$, the transformations $QU_m^{\not\perp}$ and $QU_m^{\perp}$ have an additional length-preserving hash in the ciphertext, thus the proof idea in [21, 22] can be used to prove the QROM security. As they pointed [22], such a proof technique quite relies on the additional hash. Therefore, QROM security reductions for $U^{\not\perp}$, $U^{\perp}$, $U_m^{\not\perp}$ and $U_m^{\perp}$ are missing in [16].

In this paper, we first define two new security notions, one-way against quantum plaintext checking attacks (OW-qPCA) and one-way against quantum plaintext and (classical) validity checking attacks (OW-qPVCA) (quantum plaintext checking attacks mean that the adversary can make quantum queries to the plaintext checking oracle). Then, we provide QROM security reductions for T from OW-qPCA to OW-CPA, $U^{\not\perp}$ from IND-CCA to OW-qPCA, $U^{\perp}$ from IND-CCA to OW-qPVCA, $U_m^{\not\perp}$ from IND-CCA to OW-CPA and $U_m^{\perp}$ from IND-CCA to OW-VA (one-way against validity checking attacks).

1.2 Techniques

As explained by Targhi and Unruh [22], their proof technique strongly relies on the additional hash. In their paper, they discussed the QROM security of a variant of FO transformation from OW-CPA-secure PKE to IND-CCA-secure PKE. To implement the security reduction, one needs to simulate the decryption oracle without possessing the secret key. In classical proof, a RO-query list is used to simulate such an oracle. In the QROM, the simulator has no way to learn the actual content of adversarial RO queries, therefore such a RO-query list does not exist. Targhi and Unruh circumvented this issue by adding an additional length-preserving hash (modeled as a RO) to the ciphertext. In the security reduction, this additional RO is simulated by a k-wise independent function. For every output of this RO, the simulator can recover the corresponding input by inverting this function. Thereby, the simulator can answer the decryption queries without a secret key.

When considering the generic transformations from weakly secure PKE scheme to IND-CCA-secure KEM, one needs to simulate the decapsulation oracle DECAPS without the secret key. Indeed, obviously, we can modify the scheme by adding an additional length-preserving hash to the ciphertext so that the simulator can carry out the decryption. Thus, using the key-derivation-function (KDF, modeled as a random oracle H), he can easily simulate the DECAPS oracle.

In [20, Theorem 6], Boneh et al. proved the QROM security of a generic hybrid encryption scheme [19], built from an injective trapdoor function and symmetric key encryption scheme. Inspired by their proof idea, we present a novel approach to simulate the DECAPS oracle.

The high level idea is that we associate the random oracle H (KDF in the KEM) with a secret random function H' by setting $H = H' \circ g$ such that $H'(\cdot) =$

DECAPS$(sk, \cdot)$. We demand that the function g should be indistinguishable from an injective function for any efficient quantum adversary. Thus, in the view of the adversary against the IND-CCA security of KEM, H is indeed a random oracle. Meanwhile, we can simulate the DECAPS oracle just by using H'. Note that in our simulation of the DECAPS oracle, we circumvent the decryption computation. Thereby, there is no need to read the content of adversarial RO queries, which makes it unnecessary to add an additional length-preserving hash to the ciphertext.

1.3 Discussion

Tightness. Having a tight security reduction is a desirable property for practice cryptography, especially in large-scale scenarios. A tight security reduction can ensure that breaking the scheme (within the respective adversarial model) is at least as hard as breaking the underlying hard computational problem. While, a non-tight security reduction requires to adapt the system parameters accordingly, which results in less efficient schemes.

In the ROM, if we assume that the underlying PKE scheme in transformations FO-I and FO-II is IND-CPA-secure, we can obtain a tight reduction from IND-CCA security of KEM to IND-CPA security of PKE [16]. Specially, if the PKE scheme in FO-II is instantiated with a Ring-LWE-based PKE scheme [24], the IND-CCA security of KEM can be reduced to the security of the underlying Ring-LWE problem [6]. Albrecht et al. [6] pointed out that it is an important open problem whether one can achieve QROM security for a Dent-like KEM construction with a tight reduction and without suffering any ciphertext overhead. In our work, although we present a series of QROM security reductions for the Dent-like KEM constructions without suffering any ciphertext overhead, these reductions are non-tight like previous QROM security reductions [5, 7, 11, 12, 16, 20, 22]. For the tight ROM security reductions in [6, 16], the simulators need to make an elaborate analysis of the RO-query inputs and determine which one of the query inputs can be used to break the one-way security of the underlying PKE scheme [16] or solve a decision Ring-LWE problem [6]. However, in the QROM, such a proof technique will be invalid for the reason that there is no way for the simulators to learn the RO-query inputs [25, 26]. Thus, in the QROM, it is still an important open problem that whether one can develop a novel proof technique to obtain a tight reduction for the KEM constructions discussed in this paper.

Implicit rejection. For most of the previous generic transformations from OW-CPA-secure (or IND-CPA-secure) PKE to IND-CCA-secure KEM, explicit rejection is adopted, i.e., an abnormal symbol $\perp$ is returned when an invalid ciphertext is submitted to the decapsulation algorithm. In [16], Hofheinz et al. presented several transformations with implicit rejection (the decapsulation algorithm returns a pseudorandom key for the invalid ciphertext). These two different versions (explicit rejection and implicit rejection) have their own merits. The

transformation with implicit rejection [16] does not require the underlying PKE scheme to be γ-spread [17, 18] (meaning that the ciphertexts generated by the probabilistic encryption algorithm have sufficiently large entropy), which may allow choosing better system parameters for the same security level. Whereas, the ones with explicit rejection have a relatively simple decapsulation algorithm.

In our paper, we just give QROM security reductions for the transformations with implicit rejection. It is not obvious how to extend our QROM security proofs for the transformation with explicit rejection, since the simulator has no way to tell if the submitted ciphertext is valid. In classical ROM, we usually assume the underlying PKE scheme is γ-spread. Then, we can recognize invalid ciphertexts just by testing if they are in the RO-query list, as the probability that the adversary makes queries to the decapsulation oracle with a valid ciphertext which is not in the RO-query list is negligible [6, 16–18]. Unfortunately, in the QROM, the adversary makes quantum queries to the RO, above RO-query list dose not exist. Thus, the ROM proof technique for the recognition of invalid ciphertexts is invalid in the QROM. Here, we leave it as an open problem to prove the QROM security of the transformations FO-I and FO-II with explicit rejection.

1.4 Related Works

In concurrent and independent work, [27] gives a QROM security reduction from IND-CCA security of KEM to IND-CPA security of PKE with quadratic loss, i.e., $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{B}) \leq q \cdot \sqrt{\mathtt{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A})}$. First, [27] presents a tight QROM reduction for $U^{\not\perp}_m$ from the IND-CCA security of KEM to the DS (Disjoint Simulatability) security of the underlying deterministic PKE (DPKE), where DS is a newly introduced security notion. Then, [27] gives a transformation $TPunc$ (a variant of T) that converts any IND-CPA-secure PKE into a DS-secure DPKE, where the underlying IND-CPA-secure PKE is required to be perfectly correct and have sufficiently large plaintext space. And, the QROM security reduction for $TPunc$ suffers from loose reduction with quadratic loss. Thus, taking the transformations $TPunc$ and $U^{\not\perp}_m$ together, [27] also obtains a QROM security reduction with quadratic loss from IND-CCA security of KEM to IND-CPA security of PKE.

2 Preliminaries

Symbol description. Denote $\mathcal{K}$, $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{R}$ as key space, message space, ciphertext space and randomness space, respectively. For a finite set X, we denote the sampling of a uniform random element x by $x \xleftarrow{\$} X$, and we denote the sampling according to some distribution D by $x \leftarrow D$. By $x =? y$ we denote the 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 $y := A(x)$ ($y \leftarrow A(x)$). A^H means that the algorithm A gets access to the oracle H.

2.1 Quantum Random Oracle Model

In the ROM [19], we assume the existence of a random function H, and give all parties oracle access to this function. The algorithms comprising any cryptographic protocol can use H, as can the adversary. Thus we modify the security games for all cryptographic systems to allow the adversary to make random oracle queries.

When a random oracle scheme is implemented, some suitable hash function H is included in the specification. Any algorithm (including the adversary) replaces oracle queries with evaluations of this hash function. In quantum setting, because a quantum algorithm can evaluate H on an arbitrary superposition of inputs, we must allow the quantum adversary to make quantum queries to the random oracle. We call this the quantum random oracle model [20]. Unless otherwise specified, the queries to random oracles are quantum in our paper.

Tools. Next we state four lemmas that we will use throughout the paper. The first two lemmas have been proved in other works, and we prove the last two in Appendixes B and C. Most of the background in quantum computation needed to understand this paper is just for above two proofs. Therefore, we present the necessary background in Appendix A. Here, we just recall two basic facts about quantum computation.

- Fact 1. Any classical computation can be implemented on a quantum computer.

- Fact 2. Any function that has an efficient classical algorithm computing it can be implemented efficiently as a quantum-accessible oracle.

Lemma 1 (Simulating the random oracle [28, 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 [29, 30]). *Let $\gamma \in [0,1]$. Let Z be a finite set. $F : Z \to \{0,1\}$ is the following 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., $\Pr[F(z) = 1 : z \leftarrow A^F] \leq 2q\sqrt{\gamma}$.

Note. [29, Lemma 37] and [30, Theorem 1] just consider the specific case where all p_zs are equal to γ. But in our security proof, we need to consider the case where $p_z \leq \gamma$ and p_zs are in general different from each other. Fortunately, it is not difficult to verify that the proof of [29, Lemma 37] can be extended to this generic case.

The one-way to hiding (OW2H) lemma [31, Lemma 6.2] is a useful tool for reducing a hiding (i.e., indistinguishability) property to a guessing (i.e., one-wayness) property in the security proof. Roughly speaking, the lemma states that if there exits an oracle algorithm A who issuing at most q_1 queries to random oracle $\mathcal{O}_1$ can distinguish $(x, \mathcal{O}_1(x))$ from (x, y), where y is chosen uniformly at random, we can construct another oracle algorithm B who can find x by running A and measuring one of A's query. However, in our security proof, the oracle $\mathcal{O}_1$ is not a perfectly random function and A can have access to other oracle $\mathcal{O}_2$ associated to $\mathcal{O}_1$. Therefore, we generalize the OW2H lemma.

Lemma 3 (One-way to hiding, with redundant oracle). *Let oracles $\mathcal{O}_1$, $\mathcal{O}_2$, input parameter inp and x be sampled from some joint distribution D, where $x \in \{0,1\}^n$ (the domain of $\mathcal{O}_1$). Consider an oracle algorithm $A^{\mathcal{O}_1,\mathcal{O}_2}$ that makes at most q_1 queries to $\mathcal{O}_1$ and q_2 queries to $\mathcal{O}_2$. Denote E_1 as the event that $A^{\mathcal{O}_1,\mathcal{O}_2}$ on input $(inp, x, \mathcal{O}_1(x))$ outputs 1. Reprogram $\mathcal{O}_1$ at x and replace $\mathcal{O}_1(x)$ by a uniformly random y from $\{0,1\}^m$, the codomain of $\mathcal{O}_1$. Denote E_2 as the event that $A^{\mathcal{O}_1,\mathcal{O}_2}$ on input (inp, x, y) outputs 1 after $\mathcal{O}_1$ is reprogrammed. Let $B^{\mathcal{O}_1,\mathcal{O}_2}$ be an oracle algorithm that on input (inp, x) does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_1\}$ and $y \xleftarrow{\$} \{0,1\}^m$, run $A^{\mathcal{O}_1,\mathcal{O}_2}(inp, x, y)$ until the i-th query to $\mathcal{O}_1$, measure the argument of the query in the computational basis, and output the measurement outcome. (When A makes less than i queries, B outputs $\perp \notin \{0,1\}^n$.) Let*

$$\Pr[E_1] = \Pr[b' = 1 : (\mathcal{O}_1, \mathcal{O}_2, inp, x) \leftarrow D, b' \leftarrow A^{\mathcal{O}_1,\mathcal{O}_2}(inp, x, \mathcal{O}_1(x))]$$
$$\Pr[E_2] = \Pr[b' = 1 : (\mathcal{O}_1, \mathcal{O}_2, inp, x) \leftarrow D, y \xleftarrow{\$} \{0,1\}^m, b' \leftarrow A^{\mathcal{O}_1,\mathcal{O}_2}(inp, x, y]$$
$$P_B := \Pr[x' = x : (\mathcal{O}_1, \mathcal{O}_2, inp, x) \leftarrow D, x' \leftarrow B^{\mathcal{O}_1,\mathcal{O}_2}(inp, x)].$$

Then

$$|\Pr[E_1] - \Pr[E_2]| \leq 2q_1\sqrt{P_B}.$$

Note that $\mathcal{O}_2$ is unchanged during the reprogramming of $\mathcal{O}_1$ at x. Thus, intuitively, $\mathcal{O}_2$ is redundant and unhelpful for A distinguishing $(x, \mathcal{O}_1(x))$ from (x, y). The complete proof of Lemma 3 is similar to the proof of the OW2H lemma [31, Lemma 6.2] and we present it in Appendix B.

Lemma 4. *Let Ω_H ($\Omega_{H'}$) be the set of all functions $H : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ ($H' : \{0,1\}^{n_2} \to \{0,1\}^m$). Let $H \xleftarrow{\$} \Omega_H$, $H' \xleftarrow{\$} \Omega_{H'}$, $x \xleftarrow{\$} \{0,1\}^{n_1}$. Let $F_0 = H(x, \cdot)$, $F_1 = H'(\cdot)$ Consider an oracle algorithm A^{H,F_i} that makes at most q queries to H and F_i ($i \in \{0,1\}$). If x is independent from the A^{H,F_i}'s view,*

$$\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right| \leq 2q\frac{1}{\sqrt{2^{n_1}}}.$$

We now sketch the proof of Lemma 4. The complete proof is in Appendix C.

Proof sketch. In classical setting, it is obvious that $\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right|$ can be bounded by the probability that A performs an H-query with input $(x, *)$. As x is independent from A^{H,F_i}'s view, $\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right| \leq q\frac{1}{2^{n_1}}$. In quantum setting, it is not well-defined that $\mathcal{A}$ queries $(x, *)$ from H, since H can be queried in superposition. To circumvent this problem, we follow Unruh's proof technique in [31, Lemma 6.2] and define a new adversary B who runs A, but at some random query stops and measures the query input. Let P_B be the probability that B measures x. Similarly to [31, Lemma 6.2], we can bound $\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right|$ by $2q\sqrt{P_B}$. Since x is independent from the A^{H,F_i}'s view, $P_B = \frac{1}{2^{n_1}}$. Thus, $\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right| \leq 2q\frac{1}{\sqrt{2^{n_1}}}$.

2.2 Cryptographic Primitives

Definition 1 (Public-key encryption). *A public-key encryption scheme* $\mathrm{PKE} = (Gen, Enc, Dec)$ *consists of a triple of polynomial time (in the security parameter* λ*) algorithms and a finite message space* $\mathcal{M}$*. Gen, the key generation algorithm, is a probabilistic algorithm which on input* 1^λ *outputs a public/secret key-pair* (pk, sk)*. 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, the decryption algorithm, is a deterministic algorithm which on input sk and a ciphertext c outputs a message* $m := Dec(sk, c)$ *or a special symbol* $\perp \notin \mathcal{M}$ *to indicate that c is not a valid ciphertext. We follow the definition of correctness in [16]. The 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 now define four security notions for public-key encryption: one-way against chosen plaintext attacks (OW-CPA), one-way against validity checking attacks (OW-VA), one-way against quantum plaintext checking attacks (OW-qPCA) and one-way against quantum plaintext and (classical) validity checking attacks (OW-qPVCA).

Definition 2 (OW-ATK-secure PKE). *Let* $\mathrm{PKE} = (Gen, Enc, Dec)$ *be a public-key encryption scheme with message space* $\mathcal{M}$*. For* $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{VA}, \mathrm{qPCA}, \mathrm{qPVCA}\}$*, we define OW-ATK games as in Fig. 1, where*

$$O_{ATK} := \begin{cases} \perp & \mathrm{ATK} = \mathrm{CPA} \\ \textsc{Val}(\cdot) & \mathrm{ATK} = \mathrm{VA} \\ \textsc{Pco}(\cdot, \cdot) & \mathrm{ATK} = \mathrm{qPCA} \\ \textsc{Pco}(\cdot, \cdot), \textsc{Val}(\cdot) & \mathrm{ATK} = \mathrm{qPVCA}. \end{cases}$$

Define the OW-ATK advantage function of an adversary $\mathcal{A}$ against PKE as $\mathtt{Adv}^{\text{OW-ATK}}_{\mathrm{PKE}}(\mathcal{A}) := \Pr[\text{OW-ATK}^{\mathcal{A}}_{\mathrm{PKE}} = 1]$.

Remark. We note that the security game OW-qPCA (OW-qPVCA) is the same as OW-PCA (OW-PVCA) except the adversary $\mathcal{A}$'s queries to the Pco oracle. In OW-qPCA (OW-qPVCA) game, $\mathcal{A}$ can make quantum queries to the Pco oracle, while in OW-PCA (OW-PVCA) game only the classical queries are allowed. These two new security notations will be used in the modular analysis of FO transformation in Sec. 4.

Game OW-ATK	Pco(m,c)	Val(c)
1 : $(pk, sk) \leftarrow Gen$	1 : **if** $m \notin \mathcal{M}$	1 : $m := Dec(sk, c)$
2 : $m^* \xleftarrow{\$} \mathcal{M}$	2 : **return** $\perp$	2 : **if** $m \in \mathcal{M}$
3 : $c^* \leftarrow Enc(pk, m^*)$	3 : **else return**	3 : **return** 1
4 : $m' \leftarrow \mathcal{A}^{O_{\mathrm{ATK}}}(pk, c^*)$	4 : $Dec(sk, c) =?m$	4 : **else return** 0
5 : **return** $m' =?m^*$		

Fig. 1: Games OW-ATK (ATK $\in$ {CPA, VA, qPCA, qPVCA}) for PKE, where O_{ATK} is defined in Definition 2. In games qPCA and qPVCA, the adversary $\mathcal{A}$ can query the Pco oracle with quantum state.

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 : $b' \leftarrow \mathcal{A}^{\mathrm{Decaps}}(pk, c^*, K_b^*)$	
6 : **return** $b' =?b$	

Fig. 2: IND-CCA game for KEM.

Definition 3 (Key encapsulation). *A key encapsulation mechanism KEM consists of three algorithms Gen, Encaps and Decaps. The key generation algorithm Gen outputs a key pair (pk, sk). The encapsulation algorithm Encaps, on input pk, outputs a tuple (K, c) where c is said to be an encapsulation of the key K which is contained in key space $\mathcal{K}$. 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 special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid encapsulation.*

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

Definition 4 (IND-CCA-secure KEM). *We define the IND-CCA game as in Fig. 2 and the IND-CCA advantage function of an adversary $\mathcal{A}$ against* KEM *as* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{A}) := \left|\Pr[\text{IND-CCA}^{\mathcal{A}}_{\text{KEM}} = 1] - \frac{1}{2}\right|$.

We also define OW-ATK security of PKE and IND-CCA security of KEM in the QROM, where adversary A can make quantum queries to a random oracle H. Following the work [16], 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 Security Proofs for Two Generic KEM Constructions in the QROM

In this section, we revisit two generic transformations from OW-CPA-secure PKE to IND-CCA-secure KEM. One is the transformation $\text{FO}^{\not\perp}$ in [16], which we call FO-I in our paper (see Fig. 3). The other is $\text{FO}^{\not\perp}_m$ in [16], the transformation [15, Table 5] with implicit rejection, which is denoted by FO-II (see Fig. 4). These two transformations are widely used in the post-quantum IND-CCA-secure KEM constructions [5–8, 12]. But, there are no QROM security proofs for them. To achieve QROM security, they followed Targhi and Unruh's proof idea [21, 22] and modified FO-I [5, 12, 16] and FO-II [6, 7, 16] by adding an additional length-preserving hash function to the ciphertext. Here, we present two QROM security proofs for FO-I and FO-II respectively without suffering any ciphertext overhead.

Gen'	$Encaps(pk)$	$Decaps(sk', c)$
1 : $(pk, sk) \leftarrow Gen$	1 : $m \xleftarrow{\$} \mathcal{M}$	1 : Parse $sk' = (sk, s)$
2 : $s \xleftarrow{\$} \mathcal{M}$	2 : $c = Enc(pk, m; G(m))$	2 : $m' := Dec(sk, c)$
3 : $sk' := (sk, s)$	3 : $K := H(m, c)$	3 : **if** $Enc(pk, m'; G(m')) = c$
4 : **return** (pk, sk')	4 : **return** (K, c)	4 : **return** $K := H(m', c)$
		5 : **else return**
		6 : $K := H(s, c)$

Fig. 3: IND-CCA-secure KEM-I=FO-I[PKE,G,H]

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 a pseudorandom function f with key space $\mathcal{K}^{prf}$, we associate KEM-

I=FO-I[PKE,G,H] and KEM-II=FO-II[PKE,G,H,f][5] shown in Fig. 3 and Fig. 4, respectively. The following two theorems establish that IND-CCA securities of KEM-I and KEM-II can both reduce to the OW-CPA security of PKE, in the QROM.

Gen'	$Encaps(pk)$	$Decaps(sk', c)$
1 : $(pk, sk) \leftarrow Gen$	1 : $m \xleftarrow{\$} \mathcal{M}$	1 : Parse $sk' = (sk, k)$
2 : $k \xleftarrow{\$} \mathcal{K}^{prf}$	2 : $c = Enc(pk, m; G(m))$	2 : $m' := Dec(sk, c)$
3 : $sk' := (sk, k)$	3 : $K := H(m)$	3 : **if** $Enc(pk, m'; G(m')) = c$
4 : **return** (pk, sk')	4 : **return** (K, c)	4 : **return** $K := H(m')$
		5 : **else return**
		6 : $K := f(k, c)$

Fig. 4: IND-CCA-secure KEM-II=FO-II[PKE,G,H,f]

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

Proof. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-I, issuing at most q_D queries to DECAPS, at most q_G queries to G and at most q_H queries to H. Denote Ω_G, Ω_H and $\Omega_{H'}$ as the sets of all functions $G : \mathcal{M} \to \mathcal{R}$, $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ and $H' : \mathcal{C} \to \mathcal{K}$, respectively. Consider the games in Fig. 5 and Fig. 8.

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

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

GAME G_1. In game G_1, we change the DECAPS oracle that $H_2(c)$ is returned instead of $H(s, c)$ for an invalid encapsulation c. Define an oracle algorithm A^{H,F_i} ($i \in \{0, 1\}$) as Fig. 6. Let $H = H_3$, $F_0(\cdot) = H_3(s, \cdot)$ ($s \xleftarrow{\$} \mathcal{M}$) and $F_1 = H_2$, where H_2 and H_3 are chosen in the same way as G_0 and G_1. Then,

[5] FO-II is the generic version of $\text{FO}^{\not\perp}_m$ in [16]. In their work, such a pseudorandom function f is instantiated with $H(s, \cdot)$ (s is a random seed and contained in the secret key sk').

$\Pr[G_i^{\mathcal{B}} \Rightarrow 1] = \Pr[1 \leftarrow A^{H,F_i}]$. Since the uniform secret s is chosen independently of A^{H,F_i}'s view, we can use Lemma 4 to obtain

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_H \cdot \frac{1}{\sqrt{\mathcal{M}}}.$$

GAMES $G_0 - G_4$

$(pk, sk') \leftarrow Gen'; G \xleftarrow{\$} \Omega_G$
$H_1, H_2 \xleftarrow{\$} \Omega_{H'}; H_3 \xleftarrow{\$} \Omega_H$
$m^* \xleftarrow{\$} \mathcal{M}$
$r^* := G(m^*)$ $\quad //G0 - G_3$
$r^* \xleftarrow{\$} \mathcal{R}$ $\quad //G_4$
$c^* := Enc(pk, m^*; r^*)$
$k_0^* := H(m^*, c^*)$
$k_0^* \xleftarrow{\$} \mathcal{K}$ $\quad //G_4$
$k_1^* \xleftarrow{\$} \mathcal{K}$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow B^{G,H,\textsc{Decaps}}(pk, c^*, k_b^*)$
return $b' =? b$

$H(m, c)$

if $Enc(pk, m; G(m)) = c$ $\quad //G_2 - G_4$
  return $H_1(c)$ $\quad //G_2 - G_4$
return $H_3(m, c)$

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

Parse $sk' = (sk, s)$
$m' := Dec(sk, c)$
if $Enc(pk, m'; G(m')) = c$
  return $K := H(m', c)$
else return
  $K := H(s, c)$ $\quad //G_0$
  $K := H_2(c)$ $\quad //G_1 - G_2$

DECAPS $(c \neq c^*)$ $\quad //G_3 - G_4$

return $K := H_1(c)$

Fig. 5: Games G_0-G_4 for the proof of Theorem 1

A^{H,F_i}

$(pk, sk) \leftarrow Gen; G \xleftarrow{\$} \Omega_G$
$m^* \xleftarrow{\$} \mathcal{M}$
$r^* := G(m^*)$
$c^* := Enc(pk, m^*; r^*)$
$k_0^* := H(m^*, c^*); k_1^* \xleftarrow{\$} \mathcal{K}$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow B^{G,H,\textsc{Decaps}}(pk, c^*, k_b^*)$
return $b' =? b$

DECAPS $(c \neq c^*)$

$m' := Dec(sk, c)$
if $Enc(pk, m'; G(m')) = c$
  return $K := H(m', c)$
else return
  $K := F_i(c)$

Fig. 6: A^{H,F_i} for the application of Lemma 4 in the proof of Theorem 1.

GAME G_2. Note that in game G_1, $H(m,c) = H_3(m,c)$. In game G_2, if H-query input (m, c) satisfies $g(m) = c$ ($g(\cdot) = Enc(pk, \cdot; G(\cdot))$), the response is replaced by $H_1^g(m) = H_1 \circ g(m) = H_1(g(m)) = H_1(c)$. If the function g is injective, the output distribution of H is the same as the one in G_1. In quantum setting, distinguishing the function g from an injective function is equivalent to detecting a collision in g [32–34]. Note that a collision implies an incorrect decryption. Define the event E that $\mathcal{B}$ finds a plaintext m such that $Dec(sk, g(m)) \neq m$. Then, we can bound $\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1]\right|$ by the probability of E.

Define $g' : \mathcal{M} \to \{0,1\}$ such that $g'(m) = 0$ if $Dec(sk, g(m)) = m$, and otherwise $g'(m) = 1$. If $\mathcal{B}$ can find a plaintext m such that $Dec(sk, g(m)) \neq m$ with at most q_G quantum queries to g, we can easily construct another adversary $\mathcal{B}'$ who can find a plaintext m such that $g'(m) = 1$ with at most q_G quantum queries to g'. Considering that the PKE scheme is δ-*correct*, we can derive the upper bound of $\Pr[E]$ by utilizing Lemma 2, $\Pr[E] \leq \Pr[g'(m) = 1 : m \leftarrow \mathcal{B}'^{g'}] \leq 2q_G\sqrt{\delta}$. Then,

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G\sqrt{\delta}.$$

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 ($c \neq c^*$), $K := H_1(c)$ is returned as the response. Let $m' := Dec(sk, c)$ and consider the following two cases.

Case 1: $Enc(pk, m'; G(m')) = c$. In this case, $H(m', c) = H_1(c)$. Thus, both DECAPS oracles in G_2 and G_3 return the same value.

Case 2: $Enc(pk, m'; G(m')) \neq c$. Random values $H_2(c)$ and $H_1(c)$ are returned in G_2 and G_3 respectively. In G_2, H_2 is a random function independent of the oracles G and H, thus $H_2(c)$ is uniform at random in $\mathcal{B}$'s view. In G_3, $\mathcal{B}$'s queries to H can only help him get access to H_1 at $\hat{c}$ such that $Enc(pk, \hat{m}; G(\hat{m})) = \hat{c}$ for some $\hat{m}$. Consequently, if $\mathcal{B}$ can not find a m'' such that $Enc(pk, m''; G(m'')) = c$, $H_1(c)$ is also a fresh random key just like $H_2(c)$ in his view. Since $m'' \neq m'$, finding such a m'' is exactly the event E. That is, in this case, if E does not happen, the output distributions of the DECAPS oracles in G_2 and G_3 are same in $\mathcal{B}$'s view.

As a result, G_2 and G_3 only differ when E happens. Therefore,

$$\left|\Pr[G_2^{\mathcal{B}} \Rightarrow 1] - \Pr[G_3^{\mathcal{B}} \Rightarrow 1]\right| \leq \Pr[E] \leq 2q_G\sqrt{\delta}.$$

GAME G_4. In game G_4, r^* and k_0^* are chosen uniformly at random from $\mathcal{R}$ and $\mathcal{K}$, respectively. In this game, bit b is independent from $\mathcal{B}$'s view. Hence,

$$\Pr[G_4^{\mathcal{B}} \Rightarrow 1] = \frac{1}{2}.$$

Note that in this game we reprogram the oracles G and H on inputs m^* and (m^*, c^*) respectively. Similarly, in classical setting, this will be unnoticed unless the event QUERY that $\mathcal{B}$ queries G on m^* or H on (m^*, c^*) happens. Then we can

argue that G_3 and G_4 are indistinguishable until QUERY happens. In quantum setting, due to the quantum queries to G and H, the case is complicated and we will use Lemma 3 to bound $\left|\Pr[G_2^{\mathcal{B}} \Rightarrow 1] - \Pr[G_3^{\mathcal{B}} \Rightarrow 1]\right|$. Note that (m^*, c^*) is a valid plaintext-ciphertext pair, i.e., $g(m^*) = c^*$. Therefore, $H(m^*, c^*) = H_1(c^*) = H_1(g(m^*))$. Actually, we just reprogram G and H_1^g ($H_1^g(\cdot) = H_1(g(\cdot))$) at m^*. Let $(G \times H_1^g)(x) := (G(x), H_1^g(x))$[6]. H_1^g and H_3 are internal random oracles that $\mathcal{B}$ can have access to only by querying the oracle H. Then, the number of total queries to $G \times H_1^g$ is at most $q_G + q_H$. Let H_1' be the function such that $H_1'(g(m^*)) = \perp$ and $H_1' = H_1$ everywhere else. H_1' is exactly the DECAPS oracle in G_3 and G_4 and unchanged during the reprogramming of $G \times H_1^g$.

Let $A^{G \times H_1^g, H_1'}$ be an oracle algorithm that has quantum access to $G \times H_1^g$ and H_1', see Fig. 7. Sample G, H_1, H_1^g and pk in the same way as G_3 and G_4, i.e., $(pk, sk') \leftarrow Gen', G \xleftarrow{\$} \Omega_G$, $H_1 \xleftarrow{\$} \Omega_{H'}$ and $H_1^g := H_1 \circ g$. Let $m^* \xleftarrow{\$} \mathcal{M}$. Then, if $r^* := G(m^*)$ and $k_0^* := H_1^g(m^*)$, $A^{G \times H_1^g, H_1'}$ on input $(pk, m^*, (r^*, k_0^*))$ perfectly simulates G_3. And, if $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, $A^{G \times H_1^g, H_1'}$ on input $(pk, m^*, (r^*, k_0^*))$ perfectly simulates G_4. Let $B^{G \times H_1^g, H_1'}$ be an oracle algorithm that on input (pk, m^*) does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$, $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, run $A^{G \times H_1^g, H_1'}(pk, m^*, (r^*, k_0^*))$ until the i-th query to $G \times H_1^g$, measure the argument of the query in the computational basis, output the measurement outcome (when $A^{G \times H_1^g, H_1'}$ makes less than i queries, output $\perp$). Define game G_5 as Fig. 8. Then, $\Pr[B^{G \times H_1^g, H_1'} \Rightarrow m^*] = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$.

Applying Lemma 3 with $\mathcal{O}_1 = G \times H_1^g$, $\mathcal{O}_2 = H_1'$, $inp = pk$, $x = m^*$ and $y = (r^*, k_0^*)$, we have

$$\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - \Pr[G_4^{\mathcal{B}} \Rightarrow 1]\right| \leq 2(q_G + q_H)\sqrt{\Pr[G_5^{\mathcal{B}} \Rightarrow 1]}.$$

$A^{G \times H_1^g, H_1'}(pk, m^*, (r^*, k_0^*))$

1 : $H_3 \xleftarrow{\$} \Omega_H$
2 : $c^* := Enc(pk, m^*; r^*)$
3 : $k_1^* \xleftarrow{\$} \mathcal{K}$
4 : $b \xleftarrow{\$} \{0, 1\}$
5 : $b' \leftarrow \mathcal{B}^{G, H, \text{DECAPS}}(pk, c^*, k_b^*)$
6 : **return** $b' =? b$

$H(m, c)$

1 : **if** $g(m) = c$
2 : **return** $H_1^g(m)$
3 : **else return** $H_3(m, c)$

DECAPS $(c \neq c^*)$

1 : **return** $K := H_1'(c)$

Fig. 7: $A^{G \times H_1^g, H_1'}$ for the proof of Theorem 1.

[6] Note that if one wants to make queries to G (or H_1^g) by accessing to $G \times H_1^g$, he just needs to prepare a uniform superposition of all states in the output register responding to H_1^g (or G). This trick [35, 33, 22]has been used to ignore part of the output of an oracle.

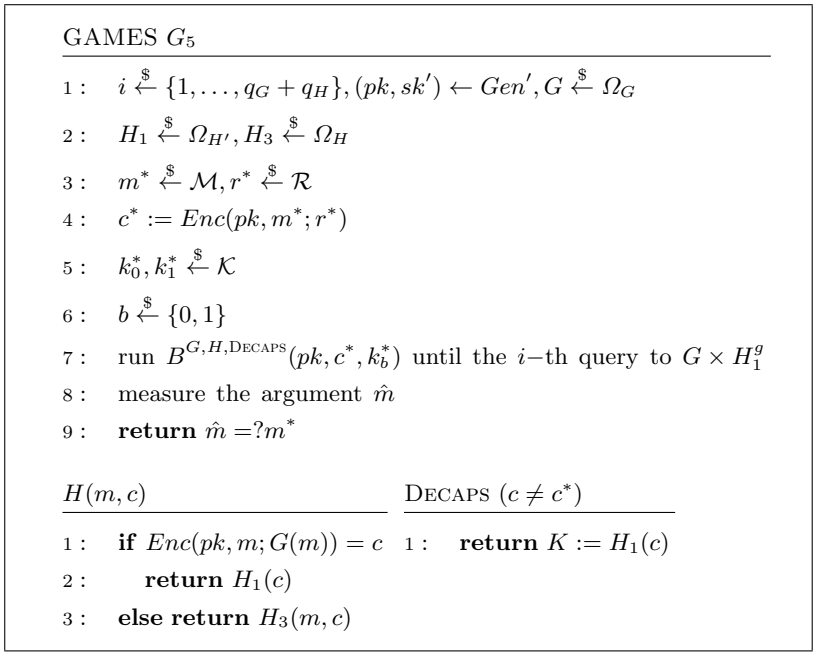

Fig. 8: Game G_5 for Theorem 1

Next, we construct an adversary $\mathcal{A}$ against the OW-CPA security of the PKE scheme such that $\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. The adversary $\mathcal{A}$ on input (1^λ, pk, c) does the following:

1. Run the adversary $\mathcal{B}$ in Game G_5.
2. Use a $2q_G$-wise independent function and two different $2q_H$-wise independent functions to simulate the random oracles G, H_1 and H_3 respectively. The random oracle H is simulated in the same way as the one in game G_5.
3. Answer the decapsulation queries by using the DECAPS oracle in Fig. 8.
4. Select $k^* \xleftarrow{\$} \mathcal{K}$ and respond to $\mathcal{B}$'s challenge query with (c, k^*).
5. Select $i \xleftarrow{\$} \{1,\ldots,q_G+q_H\}$, measure the argument $\hat{m}$ of i-th query to $G \times H_1^q$ and output $\hat{m}$.

According to Lemma 1, $\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. Finally, combing this with the bounds derived above, we can conclude that

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

□

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

Proof. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-II, issuing at most q_D classical queries to DECAPS, at most q_G queries to G and at most q_H queries to H. Consider the sequence of games given in Fig. 9. Let $\Omega_{H''}$ be the set of all functions $H'' : \mathcal{M} \to \mathcal{K}$ and we follow the same notations Ω_G, Ω_H and $\Omega_{H'}$ in Theorem 1.

GAMES $G_0 - G_4$

1 : $(pk, sk') \leftarrow Gen'$
2 : $G \xleftarrow{\$} \Omega_G, H_1 \xleftarrow{\$} \Omega_{H''}$
3 : $H_2, H_3 \xleftarrow{\$} \Omega_{H'}$
4 : $m^* \xleftarrow{\$} \mathcal{M}$
5 : $r^* := G(m^*)$ $//G0 - G_3$
6 : $r^* \xleftarrow{\$} \mathcal{R}$ $//G_4$
7 : $c^* := Enc(pk, m^*; r^*)$
8 : $k_0^* := H(m^*)$ $//G_0 - G_3$
9 : $k_0^* \xleftarrow{\$} \mathcal{K}$ $//G_4$
10 : $k_1^* \xleftarrow{\$} \mathcal{K}$
11 : $b \xleftarrow{\$} \{0,1\}$
12 : $b' \leftarrow B^{G,H,\text{DECAPS}}(pk, c^*, k_b^*)$
13 : **return** $b' =? b$

$H(m)$

1 : **return** $H_1(m)$ $//G_0 - G_1$
2 : $g(\cdot) := Enc(pk, \cdot; G(\cdot))//G_2 - G_4$
3 : **return** $H_2(g(m))$ $//G_2 - G_4$

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

1 : Parse $sk' = (sk, k)$
2 : $m' := Dec(sk, c)$
3 : **if** $Enc(pk, m'; G(m')) = c$
4 : $K := H(m')$
5 : **else return**
6 : **return** $K := f(k, c)$ $//G_0$
7 : **return** $K := H_3(c)$ $//G_1 - G_2$

DECAPS $(c \neq c^*)$ $//G_3 - G_4$

1 : **return** $K := H_2(c)$

Fig. 9: Games $G_0 - G_4$ for the proof of Theorem 2

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

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

GAME G_1. In game G_1, the DECAPS oracle is changed that the pseudorandom function f is replaced by a random function H_3. Thus, the private key k, contained in the secret key sk', is never used in G_1. Because $\mathcal{B}$'s queries to DECAPS are just classical, $\mathcal{B}$ can make classical queries to f at most q_D times. $\mathcal{B}$'s views in G_0 and G_1 are same unless there exists some adversary $\mathcal{A}'$ who can distinguish f from the random function H_3 with at most q_D classical queries to f. Then,

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq \mathtt{Adv}_{\text{PRF}}(\mathcal{A}').$$

GAME G_2. In game G_2, H_1 is substituted with $H_2 \circ g$ ($g(\cdot) := Enc(pk, \cdot; G(\cdot))$). If the function g is injective, $H_2 \circ g$ is a perfect random function. Note that $\mathcal{B}$ can not distinguish g from an injective function unless he can find a collision that $g(m_1) = g(m_2)$ ($m_1 \neq m_2$). A collision implies that the event E that $\mathcal{B}$ finds a plaintext m such that $Dec(sk, g(m)) \neq m$ happens. Using the same method in Theorem 1, we obtain $\Pr[E] \leq 2q_G\sqrt{\delta}$. Thus,

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G\sqrt{\delta}.$$

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 ($c \neq c^*$), $K := H_2(c)$ is returned as the response. Using the same analysis in Theorem 1, we know that G_2 and G_3 only differ when E happens. Hence,

$$\left|\Pr[G_2^{\mathcal{B}} \Rightarrow 1] - \Pr[G_3^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G\sqrt{\delta}.$$

GAME G_4. In game G_4, r^* and k_0^* are chosen uniformly at random from $\mathcal{R}$ and $\mathcal{K}$, respectively. In this game, bit b is independent from $\mathcal{B}$'s view. Hence,

$$\Pr[G_4^{\mathcal{B}} \Rightarrow 1] = \frac{1}{2}.$$

Let $(G \times H_2^g)(m) = (G(m), H_2^g(m))$. The number of total queries to $G \times H_2^g$ is at most $q_G + q_H$. Let H_2' be the function that $H_2'(g(m^*)) = \perp$ and $H_2' = H_2$ everywhere else.

Let $A^{G \times H_2^g, H_2'}$ be an oracle algorithm on input $(pk, m^*, (r^*, k_0^*))$ in Fig. 10. Sample G, H_2, H_2^g and pk in the same way as G_3 and G_4, i.e., $(pk, sk') \leftarrow Gen', G \xleftarrow{\$} \Omega_G$, $H_2 \xleftarrow{\$} \Omega_{H'}$ and $H_2^g := H_2 \circ g$. Let $m^* \xleftarrow{\$} \mathcal{M}$. Then, if $r^* := G(m^*)$ and $k_0^* := H_2^g(m^*)$, $A^{G \times H_2^g, H_2'}$ on input $(pk, m^*, (r^*, k_0^*))$ perfectly simulates G_3. And, if $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, $A^{G \times H_2^g, H_2'}$ on input $(pk, m^*, (r^*, k_0^*))$ perfectly simulates G_4. Let $B^{G \times H_2^g, H_2'}$ be an oracle algorithm that on input (pk, m^*) does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$, $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, run $A^{G \times H_2^g, H_2'}(pk, m^*, (r^*, k_0^*))$ until the i-th query to $G \times H_2^g$, measure the argument of the query in the computational basis, output the measurement outcome (when $A^{G \times H_2^g, H_2'}$ makes less than i queries, output $\perp$). Define game G_5 as Fig. 11. Then, $\Pr[B^{G \times H_2^g, H_2'} \Rightarrow m^*] = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$.

Applying Lemma 3 with $\mathcal{O}_1 = G \times H_2^g$, $\mathcal{O}_2 = H_2'$, $inp = pk$, $x = m^*$ and $y = (r^*, k_0^*)$, we have

$$\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - \Pr[G_4^{\mathcal{B}} \Rightarrow 1]\right| \leq 2(q_G + q_H)\sqrt{\Pr[G_5^{\mathcal{B}} \Rightarrow 1]}.$$

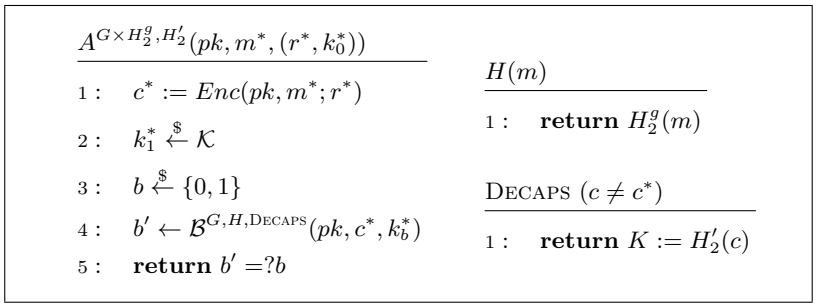

Fig. 10: $A^{G\times H_2^g, H_2'}$ for the proof of Theorem 2.

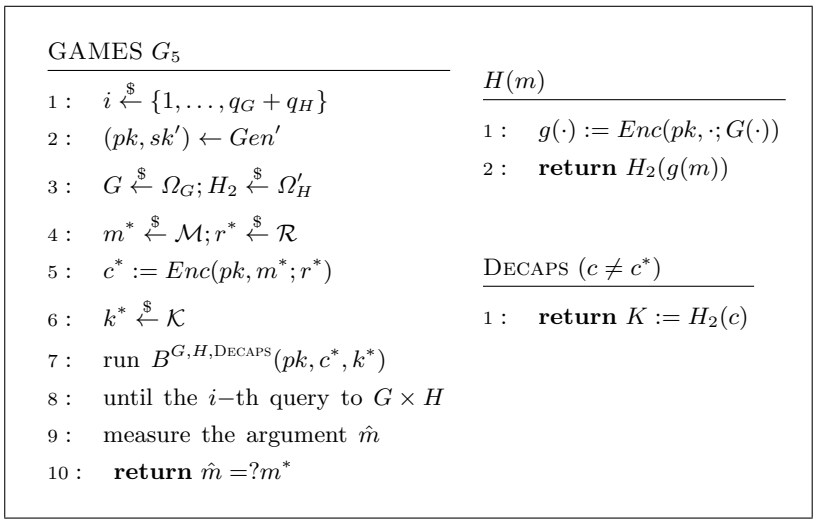

Fig. 11: Game G_5 for Theorem 2

Then, we construct an adversary $\mathcal{A}$ against the OW-CPA security of PKE such that $\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. The adversary $\mathcal{A}$ on input $(1^\lambda, pk, c)$ does the following:

1. Run the adversary $\mathcal{B}$ in game G_5.
2. Use a $2q_G$-wise independent function and a $2q_H$-wise independent function to simulate random oracles G and H_2 respectively. The random oracle H is simulated by $H_2 \circ g$. Use $G \times H$ to answer $\mathcal{B}$'s queries to both G and H.
3. Answer the decapsulation queries by using the DECAPS oracle in Fig. 11.
4. Select $k^* \xleftarrow{\$} \mathcal{K}$ and respond to $\mathcal{B}$'s challenge query with (c, k^*).
5. 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 $\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. Combing this with the bounds derived above, we can conclude that

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

□

Remark. For the reduction from the IND-CCA security of KEM to the OW-CPA security of PKE, we inevitably reprogram the quantum random oracles G and H. Lemma 3 (one-way to hiding, O2H) is a practical tool to argue the indistinguishability between games where the random oracles are reprogrammed. [16] analyzed the QROM security of $QFO^{\not\perp}_m$ (a Targhi-Unruh variant of FO-II) by two steps. First, they presented a QROM security reduction from the OW-PCA security of a intermediate scheme PKE$'$ to the OW-CPA security of the underlying PKE. In this step, the random oracle G was reprogrammed, thus by using the O2H lemma they obtained[7] that $\mathtt{Adv}^{\text{OW-PCA}}_{\text{PKE}'}(\mathcal{C}) \leq q\sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})}$. In the second step, they reduced the IND-CCA security of KEM to OW-PCA security of PKE$'$, where the random oracles H and H' (the additional hash) were reprogrammed. Again, by using the O2H lemma, they gained $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{B}) \leq q\sqrt{\mathtt{Adv}^{\text{OW-PCA}}_{\text{PKE}'}(\mathcal{C})}$. Finally, combing above two bounds, they obtained the security bound of KEM,

$$\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{B}) \leq q^{\frac{3}{2}} \cdot [\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})]^{\frac{1}{4}}. \tag{1}$$

Direct combination of the modular analyses leads to twice utilization of O2H lemma, which makes security bound highly non-tight. In our security reductions for FO-I and FO-II, we just reduce the IND-CCA security of KEM to OW-CPA security of underlying PKE scheme directly without introducing the intermediate scheme PKE$'$. Specifically, the quantum random oracles G and H are reprogrammed simultaneously, thus the O2H lemma is used just once in our reductions. Our derived security bound is approximately $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{B}) \leq q \cdot \sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})}$, which is much tighter than the bound (1).

4 Modular Analysis of FO transformation in the QROM

In [16], Hofheinz et al. introduced seven modular transformations T, $U^{\not\perp}$, $U^{\perp}$, $U^{\not\perp}_m$, $U^{\perp}_m$, $QU^{\not\perp}_m$ and $QU^{\perp}_m$. But, they just presented QROM security reductions for the transformations T, $QU^{\not\perp}_m$ and $QU^{\perp}_m$. Different from the transformations $U^{\not\perp}$, $U^{\perp}$, $U^{\not\perp}_m$ and $U^{\perp}_m$, the transformations $QU^{\not\perp}_m$ and $QU^{\perp}_m$ have an additional length-preserving hash in the ciphertext, thus they can follow the proof technique in [21, 22] to give QROM security reductions for them. As they pointed [22],

[7] The bounds here are informal. Concretely, the negligible terms and constant coefficients are not considered and the numbers of adversarial queries to different oracles are replaced by the total number q of adversarial queries to various oracles.

their QROM security reductions quite rely on this additional hash. And, QROM security reductions for $U^{\not\perp}$, $U^{\perp}$, $U_m^{\not\perp}$ and $U_m^{\perp}$ are missing in [16].

In this section, we revisit the transformations $U^{\not\perp}$, $U^{\perp}$, $U_m^{\not\perp}$ and $U_m^{\perp}$, and argue their QROM security without any modification to the constructions. [16] has shown that the transformation T can turn OW-CPA-secure PKE into OW-PCA-secure PKE in the QROM. In Section 4.1, we first show that the resulting PKE scheme by applying T to OW-CPA-secure PKE is also OW-qPCA-secure. The QROM security reduction for $U^{\not\perp}$ ($U^{\perp}$) from IND-CCA security of KEM to OW-qPCA (OW-qPVCA) security of PKE is given in Section 4.2 (4.3). In Section 4.4, we show that $U_m^{\not\perp}$ ($U_m^{\perp}$) transforms any OW-CPA-secure (OW-VA-secure) deterministic PKE into an IND-CCA-secure KEM in the QROM.

4.1 T: from OW-CPA to OW-qPCA in the QROM

To a public-key encryption PKE=(Gen,Enc,Dec) with message space $\mathcal{M}$ and randomness space R, and a hash function $G : \mathcal{M} \to \mathcal{R}$, we associate $\mathrm{PKE}' = T[\mathrm{PKE}, G]$. The algorithms of PKE'=(Gen,Enc',Dec') are defined in Fig. 12.

Theorem 3 (PKE OW-CPA $\overset{QROM}{\Rightarrow}$ PKE′ OW-qPCA). *If* PKE *is* δ-correct*, for any OW-qPCA* $\mathcal{B}$ *against* PKE′*, issuing at most* q_G *quantum queries to the random oracle* G *and at most* q_P *quantum queries to the plaintext checking oracle* Pco*, there exists an OW-CPA adversary* $\mathcal{A}$ *against PKE such that* $\mathtt{Adv}^{\mathrm{OW}-q\mathrm{PCA}}_{\mathrm{PKE}'}(\mathcal{B}) \leq 2q_G \cdot \sqrt{\delta} + (1 + 2q_G) \cdot \sqrt{\mathtt{Adv}^{\mathrm{OW\text{-}CPA}}_{\mathrm{PKE}}(\mathcal{A})}$.

The proof is essentially the same as the one of [16, Theorem 4.4] except the argument about the difference in $\mathcal{B}$'s success probability between game G_0 and game G_1. Game G_0 is exactly the original OW-qPCA game. In game G_1, the Pco oracle is replaced by a simulation that $Enc(pk, m; G(m)) =? c$ is returned for the query input (m, c). As pk is public and G is a quantum random oracle, such a Pco simulation can be queried on a quantum superposition of inputs. Note that $\mathcal{B}$'s views in game G_0 and game G_1 are totally identical unless he can find a plaintext m such that $Dec(sk, Enc(pk, m, G(m))) \neq m$. Thus, using Lemma 2, we can obtain that $\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G \cdot \sqrt{\delta}$. Then, following the security reduction for [16, Theorem 4.4], we can easily prove Theorem 3.

$Enc'(pk, m)$		$Dec'(sk, c)$	
1 :	$c = Enc(pk, m; G(m))$	1 :	$m' := Dec(sk, c)$
2 :	**return** c	2 :	**if** $Enc(pk, m'; G(m')) = c$
		3 :	**return** m'
		4 :	**else return** $\perp$

Fig. 12: OW-qPCA-secure $\mathrm{PKE}' = T[\mathrm{PKE}, G]$

4.2 $U^{\not\perp}$: from OW-qPCA to IND-CCA in the QROM

To a public-key encryption PKE′=(Gen',Enc',Dec') and a hash function H, we associate KEM-III = $U^{\not\perp}$[PKE′, H]. The algorithms of KEM-III=(Gen,$Encaps$,$Decaps$) are defined in Fig. 13.

Gen	$Encaps(pk)$	$Decaps(sk', c)$
1 : $(pk, sk) \leftarrow Gen'$	1 : $m \xleftarrow{\$} \mathcal{M}$	1 : Parse $sk' = (sk, s)$
2 : $s \xleftarrow{\$} \mathcal{M}$	2 : $c \leftarrow Enc'(pk, m)$	2 : $m' := Dec'(sk, c)$
3 : $sk' := (sk, s)$	3 : $K := H(m, c)$	3 : **if** $m' = \perp$
4 : **return** (pk, sk')	4 : **return** (K, c)	4 : **return** $K := H(s, c)$
		5 : **else return**
		6 : $K := H(m', c)$

Fig. 13: IND-CCA-secure KEM-III = $U^{\not\perp}$[PKE′, H]

Theorem 4 (PKE′ OW-qPCA $\overset{QROM}{\Rightarrow}$ KEM-III IND-CCA). *If* PKE′ *is* δ-correct*, for any IND-CCA* $\mathcal{B}$ *against KEM-III, issuing at most* q_D *(classical) queries to the decapsulation oracle* Decaps *and at most* q_H *queries to the quantum random oracle* H*, there exists a quantum OW-qPCA adversary* $\mathcal{A}$ *against PKE′ that makes at most* q_H *queries to the* Pco *oracle such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-III}}(\mathcal{B}) \leq 2q_H \frac{1}{\sqrt{\mathcal{M}}} + 2q_H \cdot \sqrt{\mathtt{Adv}^{\text{OW}-q\text{PCA}}_{\text{PKE}'}(\mathcal{A})}$.

Proof. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-III, issuing at most q_D queries to Decaps and at most q_H queries to H. We follow the notations Ω_G, Ω_H and $\Omega_{H'}$ in Theorem 1. Consider the games in Fig. 14.

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

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

Game G_1. In game G_1, the Decaps oracle is changed that $H_2(c)$ is returned instead of $H(s, c)$ for the invalid encapsulation c. Considering that $\mathcal{B}$'s view is independent from (the uniform secret) s, we can use Lemma 4 to obtain

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_H \cdot \frac{1}{\sqrt{\mathcal{M}}}.$$

Game G_2. In game G_2, H is changes that $H_1(c)$ is returned instead of $H_3(m, c)$ when $(m,\ c)$ satisfies $\text{Pco}(m, c) = 1$ (i.e., $Dec'(sk, c) = m$). Note that it is impossible that $\text{Pco}(m_1, c) = \text{Pco}(m_2, c) = 1$ for $m_1 \neq m_2$ because Dec' is

a deterministic algorithm. Further, as H_1 is a random function independent of H_3, H in game G_2 is also a uniformly random function like the one in game G_1. Thus,

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

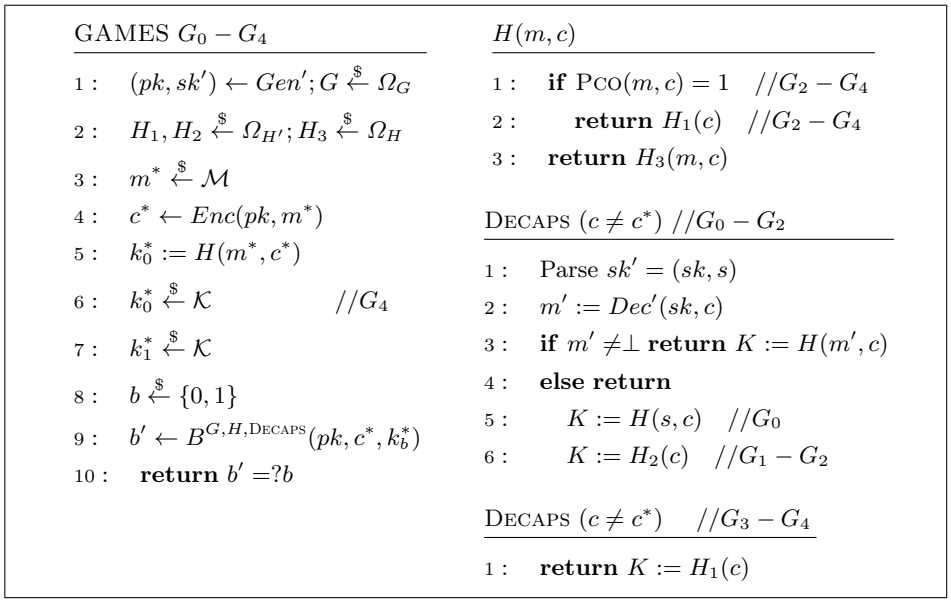

Fig. 14: Games G_0-G_4 for the proof of Theorem 4

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 ($c \neq c^*$), $K := H_1(c)$ is returned as the response. In order to show that the output distributions of DECAPS are identical in G_2 and G_3, we consider the following cases for a fixed ciphertext c and $m' := Dec'(sk, c)$.

Case 1: $m' \neq \perp$. Note that $H(m', c) = H_1(c)$ on account of PCO$(m', c) = 1$. Therefore, the two DECAPS oracles in games G_2 and G_3 return the same value.

Case 2: $m' = \perp$. Random values $H_2(c)$ and $H_1(c)$ in $\mathcal{K}$ are returned in G_2 and G_3, respectively. In G_2, H_2 is a random function independent of G and H. In G_3, $\mathcal{B}$'s queries to H can only help him get access to H_1 at c such that $Dec'(sk, c) = \hat{m}$ for some $\hat{m} \neq \perp$. Therefore, $\mathcal{B}$ never sees $H_1(c)$ by querying G and H. Hence, in $\mathcal{B}$'s view, $H_1(c)$ is totally uniform at random like $H_2(c)$. As a result, the DECAPS oracle in G_3 has the same output distribution as the one in G_2.

We have shown that $\mathcal{B}$'s views are identical in both games and

$$\Pr[G_2^{\mathcal{B}} \Rightarrow 1] = \Pr[G_3^{\mathcal{B}} \Rightarrow 1].$$

GAME G_4. In game G_4, k_0^* is chosen uniformly at random from $\mathcal{K}$. In this game, bit b is independent from $\mathcal{B}$'s view. Hence,

$$\Pr[G_4^{\mathcal{B}} \Rightarrow 1] = \frac{1}{2}.$$

$A^{H,H_1'}(pk, (m^*, c^*), k_0^*)$

1 : $k_1^* \xleftarrow{\$} \mathcal{K}$
2 : $b \xleftarrow{\$} \{0,1\}$
3 : $b' \leftarrow \mathcal{B}^{H,\text{DECAPS}}(pk, c^*, k_b^*)$
4 : **return** $b' =?b$

DECAPS $(c \neq c^*)$

1 : **return** $K := H_1'(c)$

Fig. 15: $A^{H,H_1'}$ for the proof of Theorem 4.

GAMES G_5

1 : $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}, (pk, sk) \leftarrow Gen$
2 : $H_1 \xleftarrow{\$} \Omega_{H'}, H_3 \xleftarrow{\$} \Omega_H$
3 : $m^* \xleftarrow{\$} \mathcal{M}$
4 : $c^* \leftarrow Enc(pk, m^*)$
5 : $k^* \xleftarrow{\$} \mathcal{K}$
6 : run $B^{G,H,\text{DECAPS}}(pk, c^*, k^*)$ until the $i-$th query to H
7 : measure the argument $\hat{m}\|\hat{c}$
8 : **return** $\hat{m} =?m^* \wedge \hat{c} =?c^*$

$H(m, c)$

1 : **if** $\text{PCO}(m, c) = 1$
2 : **return** $H_1(c)$
3 : **else return** $H_3(m, c)$

DECAPS $(c \neq c^*)$

1 : **return** $K := H_1(c)$

Fig. 16: Game G_5 for Theorem 4

Let $A^{H,H_1'}$ be an oracle algorithm on input $(pk, (m^*, c^*), k_0^*)$ in Fig. 15. Let $(pk, sk') \leftarrow Gen'$, $H_1 \xleftarrow{\$} \Omega_{H'}$, $H_3 \xleftarrow{\$} \Omega_H$, $m^* \xleftarrow{\$} \mathcal{M}$, $c^* \leftarrow Enc(pk, m^*)$ and H is

simulated as the one in G_3 and G_4. Let H_1' be the function with $H_1'(c^*) = \perp$ and $H_1' = H_1$ everywhere else. Then, if $k_0^* := H(m^*, c^*)$, $A^{H,H_1'}$ perfectly simulates G_3. And, if $k_0^* \xleftarrow{\$} \mathcal{K}$, $A^{H,H_1'}$ perfectly simulates G_4. Let $B^{H,H_1'}$ be an oracle algorithm that on input $(pk, (m^*, c^*))$ does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_H\}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, run $A^{H,H_1'}(pk, (m^*, c^*), k_0^*)$ until the i-th query to H, measure the argument of the query in the computational basis, output the measurement outcome (when $A^{H,H_1'}$ makes less than i queries, output $\perp$). Define game G_5 as Fig. 16. Then, $\Pr[B^{H,H_1'} \Rightarrow (m^*, c^*)] = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$.

Applying Lemma 3 with $\mathcal{O}_1 = H$, $\mathcal{O}_2 = H_1'$, $inp = pk$, $x = (m^*, c^*)$ and $y = k_0^*$, we have

$$\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - \Pr[G_4^{\mathcal{B}} \Rightarrow 1]\right| \leq 2(q_G + q_H)\sqrt{\Pr[G_5^{\mathcal{B}} \Rightarrow 1]}.$$

Then, we construct an adversary $\mathcal{A}$ against the OW-qPCA security of the PKE$'$ scheme such that $\mathtt{Adv}^{\mathrm{OW}-q\mathrm{PCA}}_{\mathrm{PKE}'}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. The adversary $\mathcal{A}$ on input (1^λ, pk, c) does the following:

1. Run the adversary $\mathcal{B}$ in game G_5.
2. Use two different $2q_H$-wise independent functions to simulate the random oracles H_1 and H_3 respectively. The random oracle H is simulated[8] in the same way as the one in game G_5.
3. Answer the decapsulation queries by using the DECAPS oracle in Fig. 16.
4. Select $k^* \xleftarrow{\$} \mathcal{K}$ and respond to $\mathcal{B}$'s challenge query with (c, k^*).
5. Select $i \xleftarrow{\$} \{1, \ldots, q_H\}$, measure the argument $\hat{m}\|\hat{c}$ of the i-th query to H and output $\hat{m}$.

According to Lemma 1, $\mathtt{Adv}^{\mathrm{OW}-q\mathrm{PCA}}_{\mathrm{PKE}'}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. Finally, combing this with the bounds derived above, we can conclude that

$$\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-III}}(\mathcal{B}) \leq 2q_H \frac{1}{\sqrt{\mathcal{M}}} + 2q_H \cdot \sqrt{\mathtt{Adv}^{\mathrm{OW}-q\mathrm{PCA}}_{\mathrm{PKE}'}(\mathcal{A})}.$$

□

4.3 $U^\perp$: from OW-qPVCA to IND-CCA in the QROM

To a public-key encryption PKE$'$=(Gen',Enc',Dec') and a hash function H, we associate KEM-IV = $U^\perp$[PKE$'$, H]. We remark that $U^\perp$ is essentially the transformation [15, Table 2], a KEM variant of the REACT/GEM transformations [36, 37]. The algorithms of KEM-IV=(Gen,$Encaps$,$Decaps^\perp$) are defined in Fig. 17.

[8] To simulate the quantum random oracle H, we need to make quantum queries to the PCO oracle. This is the reason why we require the scheme PKE$'$ to be OW-qPCA-secure.

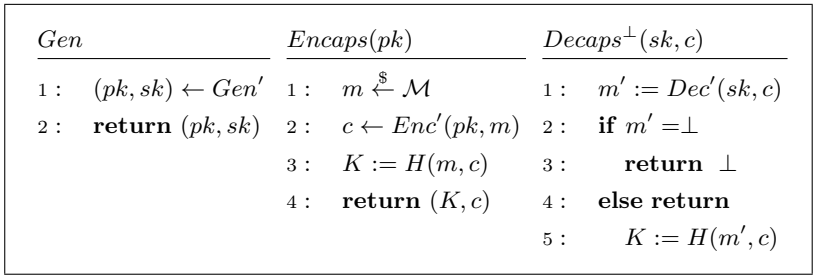

Fig. 17: IND-CCA-secure KEM-IV = $U^{\perp}[\mathrm{PKE}', H]$

Theorem 5 (PKE′ OW-qPVCA $\overset{QROM}{\Rightarrow}$ KEM-IV IND-CCA). *If* PKE′ *is* δ-correct*, for any IND-CCA $\mathcal{B}$ against KEM-IV, issuing at most q_D (classical) queries to the decapsulation oracle* Decaps *and at most q_H queries to the quantum random oracle H, there exists an OW-qPVCA adversary $\mathcal{A}$ against PKE′ that makes at most q_H queries to the* Pco *oracle and at most q_D queries to the* Val *oracle such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-IV}}(\mathcal{B}) \leq 2q_H \cdot \sqrt{\mathtt{Adv}^{\text{OW}-q\text{PVCA}}_{\text{PKE}'}(\mathcal{A})}$.

The only difference between KEM-III and KEM-IV is the response to the invalid ciphertext in the decapsulation algorithm. When the ciphertext c is invalid, the decapsulation algorithm in KEM-III returns a random key related to c. In this way, whatever the ciphertext (valid or invalid) is submitted, the return values have the same distribution. As a result, $\mathcal{A}$ can easily simulate the decapsulation oracle Decaps without recognition of the invalid ciphertexts. While the decapsulation algorithm in KEM-IV returns $\perp$ when the submitted c is invalid. Thus, in order to simulate Decaps, $\mathcal{A}$ needs to judge if the ciphertext c is valid. As we assume that the scheme PKE′ is OW-qPVCA-secure, $\mathcal{A}$ can query the Val oracle to fulfill such a judgement. Then, it is easy to verify that by using the same proof method in Theorem 4 we can obtain the desired security bound.

4.4 $U^{\not\perp}_{m}$/$U^{\perp}_{m}$: from OW-CPA/OW-VA to IND-CCA for Deterministic Encryption in the QROM

The transformation $U^{\not\perp}_{m}$ ($U^{\perp}_{m}$) is a variant of $U^{\not\perp}$ ($U^{\perp}$) that derives the KEM key as $K = H(m)$ instead of $K = H(m, c)$. To a deterministic public-key encryption scheme PKE′ = (Gen', Enc', Dec') with message space $\mathcal{M}$, a hash function $H : \mathcal{M} \to \mathcal{K}$, and a pseudorandom function f with key space $\mathcal{K}^{prf}$, we associate KEM-V=$U^{\not\perp}_{m}[\mathrm{PKE}', H, f]$ and KEM-VI=$U^{\perp}_{m}[\mathrm{PKE}', H]$ shown in Fig. 18 and Fig. 19, respectively.

We note that for a deterministic PKE scheme the OW-PCA security is equivalent to the OW-CPA security as we can simulate the Pco oracle via re-encryption during the proof. Thus, combing the proofs of Theorem 2, Theorem 4 and Theorem 5, we can easily obtain the following two theorems.

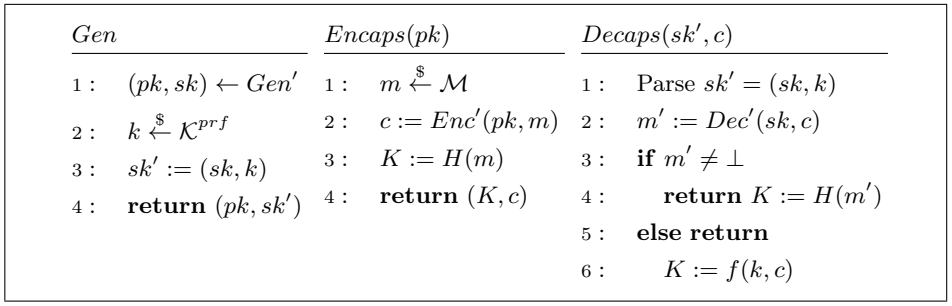

Fig. 18: IND-CCA-secure KEM-V=$U_m^{\not\perp}$[PKE′,H,f]

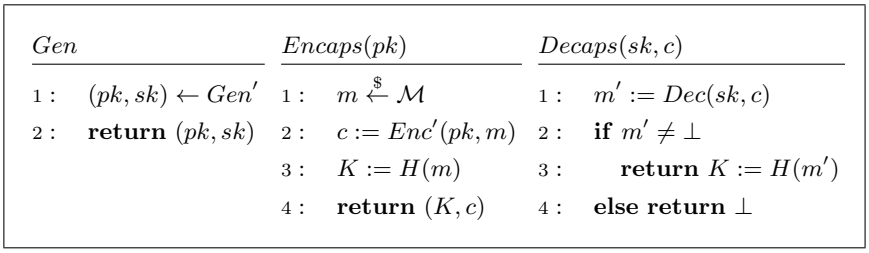

Fig. 19: IND-CCA-secure KEM-VI=$U_m^{\perp}$[PKE′,H]

Theorem 6 (PKE′ OW-CPA $\overset{QROM}{\Rightarrow}$ KEM-V IND-CCA). *If* PKE′ *is* δ-correct *and deterministic, for any IND-CCA* $\mathcal{B}$ *against KEM-V, issuing at most* q_E *quantum queries to the encryption oracle*[9]*, at most* q_D *(classical) queries to the decapsulation oracle* Decaps *and at most* q_H *quantum queries to the random oracle* H*, there exists a quantum OW-CPA adversary* $\mathcal{A}$ *against* PKE′ *and an adversary* $\mathcal{A}'$ *against the security of* PRF *with at most* q_D *classical queries such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-V}}(\mathcal{B}) \leq \mathtt{Adv}_{\text{PRF}}(\mathcal{A}') + 4q_E\sqrt{\delta} + 2q_H \cdot \sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}'}(\mathcal{A})}$.

Theorem 7 (PKE′ OW-VA $\overset{QROM}{\Rightarrow}$ KEM-VI IND-CCA). *If* PKE′ *is* δ-correct *and deterministic, for any IND-CCA* $\mathcal{B}$ *against KEM-VI, issuing at most* q_E *quantum queries to the encryption oracle, at most* q_D *(classical) queries to the decapsulation oracle* Decaps *and at most* q_H *quantum queries to the random oracle* H*, there exists a quantum OW-VA adversary* $\mathcal{A}$ *against* PKE′ *who makes at most* q_D *queries to the* Val *oracle such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-VI}}(\mathcal{B}) \leq 2q_E\sqrt{\delta} + 2q_H \cdot \sqrt{\mathtt{Adv}^{\text{OW-VA}}_{\text{PKE}'}(\mathcal{A})}$.

[9] For the deterministic scheme PKE′, given public key pk, quantum adversary $\mathcal{B}$ can execute the encryption algorithm Enc' in a quantum computer.

References

1. NIST: National institute for standards and technology. Postquantum crypto project (2017) http://csrc.nist.gov/groups/ST/post-quantum-crypto/.
2. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing **33**(1) (2003) 167–226
3. Boyd, C., Cliff, Y., Gonzalez Nieto, J., Paterson, K.G.: Efficient one-round key exchange in the standard model. In Mu, Y., Susilo, W., Seberry, J., eds.: Information Security and Privacy, 13th Australasian Conference– ACISP 2008. Volume 5107 of LNCS., Springer-Verlag (2008) 69–83
4. Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Strongly secure authenticated key exchange from factoring, codes, and lattices. Designs, Codes and Cryptography **76**(3) (2015) 469–504
5. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Stehlé, D.: Crystals–kyber: a cca-secure module-lattice-based kem. Technical report, Cryptology ePrint Archive, Report 2017/634, 2017. http://eprint. iacr. org/2017/634
6. Albrecht, M.R., Orsini, E., Paterson, K.G., Peer, G., Smart, N.P.: Tightly secure ring-lwe based key encapsulation with short ciphertexts. In Foley, S.N., Gollmann, D., Snekkenes, E., eds.: 22nd European Symposium on Research in Computer Security–ESORICS 2017. Volume 10492 of LNCS., Springer (2017) 29–46
7. 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
8. Stam, M.: A key encapsulation mechanism for ntru. In Smart, N.P., ed.: Proceedings of the 10th international conference on Cryptography and Coding. Volume 3796 of LNCS., Springer-Verlag (2005) 410–427
9. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: Ntru prime: reducing attack surface at low cost. SAC 2017 (to appear) (2017) http://eprint.iacr.org/2016/461.
10. Peikert, C.: Lattice cryptography for the internet. In Mosca, M., ed.: International Workshop on Post-Quantum Cryptography–PQCrypto 2014. Volume 8772 of LNCS., Springer (2014) 197–219
11. Cheon, J.H., Han, K., Kim, J., Lee, C., Son, Y.: A practical post-quantum publickey cryptosystem based on splwe. In Hong, S., Park, J.H., eds.: International Conference on Information Security and Cryptology–ICISC 2016. Volume 10157 of LNCS., Springer (2016) 51–74
12. Barreto, P.S.L.M., Gueron, S., Gueneysu, T., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J.P.: Cake: Code-based algorithm for key encapsulation. Cryptology ePrint Archive, Report 2017/757 (2017) http://eprint.iacr.org/2017/757.
13. Bernstein, D.J., Chou, T., Schwabe, P.: Mcbits: Fast constant-time code-based cryptography. In Bertoni, G., Coron, J., eds.: Cryptographic Hardware and Embedded Systems–CHES 2013. Volume 8086 of LNCS., Springer (2013) 250–272
14. 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
15. 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

16. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki-okamoto transformation. In: Theory of Cryptography Conference-TCC 2017 (to appear). (2017) http://eprint.iacr.org/2017/604.
17. 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
18. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of cryptology **26**(1) (2013) 1–22
19. 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
20. 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
21. 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
22. 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
23. Grover, L.K.: A fast quantum mechanical algorithm for database search. In Miller, G.L., ed.: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing–STOC 1996, ACM (1996) 212–219
24. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In Gilbert, H., ed.: Advances in Cryptology–EUROCRYPT 2010. Volume 6110 of LNCS., Springer (2010) 1–23
25. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum private queries. Physical review letters **100**(23) (2008) 230502
26. De Martini, F., Giovannetti, V., Lloyd, S., Maccone, L., Nagali, E., Sansoni, L., Sciarrino, F.: Experimental quantum private queries with linear optics. Physical Review A **80**(1) (2009) 010302
27. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. Technical report, Cryptology ePrint Archive, Report 2017/1005, 2017. http://eprint. iacr. org/2017/1005
28. 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
29. 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
30. Cheng, C., Chung, K., Persiano, G., Yang, B., eds.: 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)
31. Unruh, D.: Revocable quantum timed-release encryption. Journal of the ACM **62**(6) (2015) 49:1–49:76
32. Yuen, H.: A quantum lower bound for distinguishing random functions from random permutations. Quantum Information & Computation **14**(13-14) (2014) 1089–1097
33. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Information & Computation **15**(7-8) (2015) 557–567

34. Targhi, E.E., Tabia, G.N., Unruh, D.: Quantum collision-resistance of non-uniformly distributed functions. In Takagi, T., ed.: International Workshop on Post-Quantum Cryptography–PQCrypto 2016. LNCS, Springer (2016) 79–85
35. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Advances in Cryptology–CRYPTO 2013, Springer (2013) 361–379
36. Okamoto, T., Pointcheval, D.: React: Rapid enhanced-security asymmetric cryptosystem transform. In Naccache, D., ed.: Topics in CryptologyCT-RSA 2001. Volume 2020 of LNCS., Springer (2001) 159–174
37. 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 CryptologyCT-RSA 2002. Volume 2271 of LNCS., Springer (2002) 263–276
38. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Number 2. Cambridge University Press (2000)
39. Unruh, D.: Quantum position verification in the random oracle model. In Garay, J.A., Gennaro, R., eds.: Advances in Cryptology–CRYPTO 2014. Volume 8617 of LNCS., Springer (2014) 1–18

A Quantum Computation

We give a short introduction to quantum computation. For a more thorough discussion, please see [38].

A quantum system A is a complex Hilbert space $\mathcal{H}$ with an inner product $\langle\cdot|\cdot\rangle$. The state of a quantum system is given by a vector $|\Psi\rangle$ of unit norm ($\langle\Psi|\Psi\rangle = 1$). Given quantum systems A and B over spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively, we define the joint or composite quantum system through the tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$. The product state of $|\varphi_A\rangle \in \mathcal{H}_A$ and $|\varphi_B\rangle \in \mathcal{H}_B$ is denoted by $|\varphi_A\rangle \otimes |\varphi_B\rangle$ or simply $|\varphi_A\rangle|\varphi_B\rangle$. A n-qubit system lives in the joint quantum system of n two-dimensional Hilbert spaces. The standard orthonormal computational basis $B = \{|x\rangle\}$ for such a system is given by $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$ for $x = x_1 \cdots x_n$. Any (classical) bit string x is encoded into a quantum state by $|x\rangle$. Denote $TD(|\Psi\rangle, |\varphi\rangle)$ as the trace distance between quantum states $|\Psi\rangle$ and $|\varphi\rangle$.

Quantum measurement. Given a state $|\varphi\rangle$, we can measure $|\varphi\rangle$ in the basis B, obtaining the value x with probability $|\langle x|\varphi\rangle|^2$. Thus, to each $|\varphi\rangle$, we associate a distribution D_φ where $D_\varphi(x) = |\langle x|\varphi\rangle|^2$. The normalization constant and the fact that B is an orthonormal basis ensure that D_φ is exactly a valid distribution. After measurement, the system is in state $|x\rangle$.

Quantum algorithm. A quantum algorithm A over a Hilbert space $\mathcal{H}$ with a standard orthonormal basis B is specified by unitary transformation U. The input to A is the initial state $|x_0\rangle$. Then U is applied to the system, and the final state is obtained $|\varphi\rangle = U|x_0\rangle$. At last, A's output is obtained by performing a measurement on $|\varphi\rangle$.

Quantum algorithm usually operates on a product space $S \otimes K \otimes V$, where S represents the work space, K the input space, and V the output space. Given

a function $H : K \to V$, define the standard orthonormal basis B as the set $|s, k, v\rangle$ for $s \in S$, $k \in K$, and $v \in V$. Define the unitary transformation O_H over the Hilbert space spanned by B as the transformation that takes $|s, k, v\rangle$ into $|s, k, v \oplus H(k)\rangle$. O_H is unitary, its own inverse, and Hermitian.

A quantum algorithm A making q quantum queries to H is then specified by a sequence of unitary transformations $U_0, \ldots, U_q$. The evaluation of A then consists of alternately applying U_i and O_H to the initial state $U_0|x_0\rangle$. The final state of the algorithm is

$$U_q O_H \ldots U_1 O_H U_0 |x_0\rangle.$$

We say that a quantum algorithm is efficient if q is a polynomial, and all the U_is are composed of a polynomial number of universal basis gates (the Hadamard, CNOT, and phase shift gates are commonly used).

B Proof of Lemma 3

Proof. Assume that A uses three quantum systems S, K and V for its state, oracle input and oracle output, where K has two subsystems $K = K_1 \otimes K_2$ and V has two subsystems $V = V_1 \otimes V_2$. Let $x_i, y_i \in \{0, 1\}$ $(i \in \{1, 2, \ldots, q\}, q = q_1 + q_2)$ such that $\sum x_i = q_1, \sum y_i = q_2, x_i + y_i = 1$. Then an execution of A leads to the final state

$$|\Psi_q\rangle := \prod_{i=1}^{q} (U_2^i O_2^{y_i} U_1^i O_1^{x_i}) |\Psi_0\rangle,$$

where $|\Psi_0\rangle$ is the initial state, U_1 and U_2 are A's state transition operations, O_1 and O_2 are the oracle queries such that $O_1|s, k_1, k_2, v_1, v_2\rangle := |s, k_1, k_2, v_1 \oplus \mathcal{O}_1(k_1), v_2\rangle, O_2|s, k_1, k_2, v_1, v_2\rangle := |s, k_1, k_2, v_1, v_2 \oplus \mathcal{O}_2(k_2)\rangle$. A's output is produced by applying a measurement M to A's final state. Then,

$$\Pr[E_1] = \sum_{(\mathcal{O}_1, \mathcal{O}_2, inp, x)y} \alpha b,$$

where α is the probability of each particular pair $(\mathcal{O}_1, \mathcal{O}_2, inp, x)y$ and $b = \Pr[M \text{ outputs 1 on state } |\Psi_q\rangle$.

Reprogram $\mathcal{O}_1$ at x. Denote O_1' as the function that $O_1'(x) = y$ and $O_1' = O_1$ everywhere else. Then, the final state becomes

$$|\Psi_q'\rangle := \prod_{i=1}^{q} (U_2^i O_2^{y_i} U_1^i {O_1'}^{x_i}) |\Psi_0\rangle.$$

Thus,

$$\Pr[E_2] = \sum_{(\mathcal{O}_1, \mathcal{O}_2, inp, x)y} \alpha b',$$

where $b' = \Pr[M \text{ outputs 1 on state} |\Psi_q'\rangle$.

According to [38, Theorem 9.1], we know that

$$|\Pr[E_1] - \Pr[E_2]| \leq \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)y} \alpha \left|b - b'\right| \leq \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)y} \alpha D_q, \tag{2}$$

where $D_q := TD(|\Psi_q\rangle, |\Psi'_q\rangle)$ is the trace distance between quantum states $|\Psi_q\rangle$ and $|\Psi'_q\rangle$.

Note the fact that the difference between $|\Psi_q\rangle$ and $|\Psi'_q\rangle$ just comes from the difference between O_1 and O'_1. Thus, the formulas of $|\Psi_q\rangle$ and $|\Psi'_q\rangle$ can be simplified by $|\Psi_q\rangle := \prod_{i=1}^{q_1}(U_iO_1)U_0|\Psi_0\rangle$ and $|\Psi'_q\rangle := \prod_{i=1}^{q_1}(U_iO'_1)U_0|\Psi_0\rangle$, where U_i is the product of the transformations between the i-th O_1 and $(i+1)$-th O_1. Specifically, $U_0 = \prod_{l<j_1}(U_2^l O_2^{y_l} U_1^l {O_l}^{x_l})$, $U_i = \prod_{j_i \leq l < j_{i+1}}(U_2^l O_2^{y_l} U_1^l {O_l}^{x_l}) \times {O_l}^{x_{j_i}}$ $(1 \leq i < q_1)$ and $U_{q_1} = \prod_{l>j_{q_1}}(U_2^l O_2^{y_l} U_1^l {O_l}^{x_l}) \times U_2^{j_{q_1}} O_2^{y_{j_{q_1}}} U_1^{j_{q_1}}$ $(j_i \in \{i : x_i = 1\}$, $j_1 < j_2 \ldots < j_{q_1})$.

Define $|\Phi_i\rangle := \prod_{j=1}^{i}(U_jO_1)U_0|\Psi_0\rangle$ and $|\Phi'_i\rangle := \prod_{j=1}^{i}(U_jO'_1)U_0|\Psi_0\rangle$ $(i \in \{1,\ldots,q_1\})$. Then, $|\Phi_{q_1}\rangle = |\Psi_q\rangle$, $|\Phi'_{q_1}\rangle = |\Psi'_q\rangle$ and $D_q = TD(|\Psi_q\rangle, |\Psi'_q\rangle) = TD(|\Phi_{q_1}\rangle, |\Phi'_{q_1}\rangle)$.

Describe B as follows: $B^{\mathcal{O}_1,\mathcal{O}_2}(inp,x)$ picks $i \xleftarrow{\$} \{1,\ldots,q_1\}$ and $y \xleftarrow{\$} \{0,1\}^m$, measures the quantum system K_1 of the state $|\Phi'_{i-1}\rangle$, and outputs the result. Thus,

$$P_B := \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)yi} \frac{\alpha}{q_1} \left\|Q_x|\Phi'_{i-1}\rangle\right\|^2,$$

where Q_x is the projector projecting K_1 onto $|x\rangle$ (i.e., $Q_x = I \otimes |x\rangle\langle x| \otimes I \otimes I$).

In fact, we can view K_2 and V_2 as the subsystems of the auxiliary quantum system S. Then, according to the proof of the OW2H lemma in [31, Lemma 6.2], we can directly obtain $|\Pr[E_1] - \Pr[E_2]| \leq 2q_1\sqrt{P_B}$. But, for completeness, we also preset the complete proof here.

Let $D_i := TD(|\Phi_i\rangle, |\Phi'_i\rangle)$. $D_0 = TD(U_0|\Phi_0\rangle, U_0|\Phi_0\rangle)$ and

$$\begin{aligned} D_i &= TD(U_iO_1|\Phi_{i-1}\rangle, U_iO'_1|\Phi'_{i-1}\rangle) \\ &\leq TD(U_iO_1|\Phi_{i-1}\rangle, U_iO_1|\Phi'_{i-1}\rangle) + TD(U_iO_1|\Phi'_{i-1}\rangle, U_iO'_1\Phi'_{i-1}\rangle) \\ &\leq D_{i-1} + TD(O_1|\Phi'_{i-1}\rangle, O'_1|\Phi'_{i-1}\rangle). \end{aligned}$$

Hence,

$$D_q \leq \sum_{i=1}^{q} TD(O_1|\Phi'_{i-1}\rangle, O'_1|\Phi'_{i-1}\rangle). \tag{3}$$

Let $V_y|s,k_1,k_2,v_1,v_2\rangle := |s,k_1,k_2,v_1 \oplus y,v_2\rangle$. Then $O'_1 = O_1(1-Q_x) + V_yQ_x$. By using [39, Lemma 12], we can get that

$$\begin{aligned} &TD(O_1|\Phi'_{i-1}\rangle, O'_1|\Phi'_{i-1}\rangle) \\ &= TD(O_1(1-Q_x)|\Phi'_{i-1}\rangle + O_1Q_x|\Phi'_{i-1}\rangle, O_1(1-Q_x)|\Phi'_{i-1}\rangle + V_yQ_x|\Phi'_{i-1}\rangle) \\ &\leq 2\left\|O_1Q_x|\Phi'_{i-1}\rangle\right\| = 2\left\|Q_x|\Phi'_{i-1}\rangle\right\|. \end{aligned} \tag{4}$$

Combing the equations (2, 3, 4), we obtain that

$$\begin{aligned}|\Pr[E_1]-\Pr[E_2]| &\leq \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)y} \alpha D_q \leq \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)yi} \alpha TD(O_1|\Phi'_{i-1}\rangle, O'_1|\Phi'_{i-1}\rangle)\\ &\leq \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)yi} \alpha 2\left\|Q_x|\Phi'_{i-1}\rangle\right\|\\ &\overset{(*)}{\leq} 2q_1\sqrt{\sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)yi}\frac{\alpha}{q_1}\left\|Q_x|\Phi'_{i-1}\rangle\right\|^2} = 2q_1\sqrt{P_B},\end{aligned}$$

where (*) uses Jensen's inequality. □

C Proof of Lemma 4

Proof. Assume that A uses three quantum systems S, K and V for its state, oracle input and oracle output, where K has two subsystems $K = K_1 \otimes K_2$. K_1, K_2 and V have n_1, n_2 and m qubits respectively. Then an execution of A leads to the final state $(UO_H)^q|\Psi_{xH'}\rangle$, where $|\Psi_{xH'}\rangle$ is the initial state, $O_H \mid s, k_1 \otimes k_2, v\rangle := |s, k_1 \otimes k_2, v \oplus H(k_1, k_2)\rangle$, and U is A's state transition operation. We assume that all the transition operations U_i are identical and equal to U (the proof in the general case is essentially identical). A's output is produced by applying a measurement M to A's final state.

Define $|\Psi^i_{HxH'}\rangle := (UO_H)^i|\Psi_{xH'}\rangle$. Then, we can obtain

$$\Pr[E_1] = \sum_{HxH'} \alpha b_{HxH'},$$

where $b_{HxH'} = \Pr[M \textit{ outputs } 1 \textit{ on state } |\Psi^q_{HxH'}\rangle]$, $\alpha = 2^{-m2^{(n_1+n_2)}-n_1-m2^{n_2}}$.

Reprogram H at $(x, \cdot)$. Denote $H_{xH'}$ as the function that $H_{xH'}(x, \cdot) = H'(\cdot)$ and $H_{xH'} = H$ everywhere else. Thus,

$$\Pr[E_2] = \sum_{HxH'} \alpha b_{H_{xH'}xH'}.$$

According to [38, Theorem 9.1], we know that

$$|\Pr[E_1]-\Pr[E_2]| \leq \sum_{HxH'} \alpha\left|b_{HxH'} - b_{H_{xH'}xH'}\right| \leq \sum_{HxH'} \alpha D_q, \tag{5}$$

where $D_i := TD(|\Psi^i_{HxH'}\rangle, |\Psi^i_{H_{xH'}xH'}\rangle)$ is the trace distance between quantum states $|\Psi^i_{HxH'}\rangle$ and $|\Psi^i_{H_{xH'}xH'}\rangle$.

Note that $D_0 = TD(|\Psi_{xH'}\rangle, |\Psi_{xH'}\rangle) = 0$ and

$$\begin{aligned}D_i &= TD(UO_H|\Psi^{i-1}_{HxH'}\rangle, UO_{H_{xH'}}|\Psi^{i-1}_{H_{xH'}xH'}\rangle)\\ &\leq TD(UO_H|\Psi^{i-1}_{HxH'}\rangle, UO_{H_{xH'}}|\Psi^{i-1}_{HxH'}\rangle) + TD(UO_{H_{xH'}}|\Psi^{i-1}_{HxH'}\rangle, UO_{H_{xH'}}|\Psi^{i-1}_{H_{xH'}xH'}\rangle)\\ &\leq D_{i-1} + TD(O_H|\Psi^{i-1}_{HxH'}\rangle, O_{H_{xH'}}|\Psi^{i-1}_{HxH'}\rangle).\end{aligned}$$

Hence,

$$D_q \leq \sum_{i=1}^{q} TD(O_H|\Psi_{HxH'}^{i-1}\rangle, O_{H_{xH'}}|\Psi_{HxH'}^{i-1}\rangle) \tag{6}$$

Let $O_{H'}|a, k_1 \otimes k_2, v\rangle := |a, k_1 \otimes k_2, v \oplus H'(k_2)\rangle$. Q_x is the projector projecting K_1 onto $|x\rangle$ (i.e., $Q_x = I \otimes |x\rangle\langle x| \otimes I \otimes I$). Then, $O_{H_{xH'}} = O_H(1-Q_x) + O_{H'}Q_x$. By using [39, Lemma 12], we can get that

$$\begin{aligned}
&TD(O_H|\Psi_{HxH'}^{i-1}\rangle, O_{H_{xH'}}|\Psi_{HxH'}^{i-1}\rangle) \\
&= TD(O_H(1-Q_x)|\Psi_{HxH'}^{i-1}\rangle + O_HQ_x|\Psi_{HxH'}^{i-1}\rangle, O_H(1-Q_x)|\Psi_{HxH'}^{i-1}\rangle + O_{H'}Q_x|\Psi_{HxH'}^{i-1}\rangle) \\
&\leq 2\left\|O_HQ_x|\Psi_{HxH'}^{i-1}\rangle\right\| = 2\left\|Q_x|\Psi_{HxH'}^{i-1}\rangle\right\|.
\end{aligned} \tag{7}$$

Combing the equations (5, 6, 7), we obtain that

$$|\Pr[E_1] - \Pr[E_2]| \leq \sum_{HxH'i} 2\alpha \left\|Q_x|\Psi_{HxH'}^{i-1}\rangle\right\| \overset{(*)}{\leq} 2q\sqrt{\sum_{HxH'i} \frac{\alpha}{q}\left\|Q_x|\Psi_{HxH'}^{i-1}\rangle\right\|^2},$$

where (*) uses Jensen's inequality.

Define algorithm B as follows: pick $i \overset{\$}{\leftarrow} \{1, \ldots, q\}$, measure the quantum system K_1 of A's i-th query state $|\Psi_{HxH'}^{i-1}\rangle$, obtain $\hat{x}$ and output $\hat{x} = ?x$. Thus, $\Pr[B \Rightarrow 1]$ is exactly $\sum_{HxH'i} \frac{\alpha}{q}\left\|Q_x|\Psi_{HxH'}^{i-1}\rangle\right\|^2$. Because x is chosen uniformly at random and independent from A's view, $\Pr[B \Rightarrow 1] = \frac{1}{2^{n_1}}$. Therefore,

$$|\Pr[E_1] - \Pr[E_2]| \leq 2q\frac{1}{\sqrt{2^{n_1}}}.$$

□