(Tightly) QCCA-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model

Keita Xagawa and Takashi Yamakawa

NTT Secure Platform Laboratories
3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585 Japan
{keita.xagawa.zv,takashi.yamakawa.ga}@hco.ntt.co.jp
August 25, 2021

Abstract. This paper studies indistinguishability against *quantum* chosen-ciphertext attacks (IND-qCCA security) of key-encapsulation mechanisms (KEMs) in quantum random oracle model (QROM). We show that the SXY conversion proposed by Saito, Yamakawa, and Xagawa (EUROCRYPT 2018) and the HU conversion proposed by Jiang, Zhang, and Ma (PKC 2019) turn a weakly-secure deterministic public-key encryption scheme into an IND-qCCA-secure KEM scheme in the QROM. The proofs are very similar to that for the IND-CCA security in the QROM, easy to understand, and as tight as the original proofs.
keywords: Tight security, quantum chosen-ciphertext security, post-quantum cryptography, KEM.

1 Introduction

Quantum Superposition Attacks: Scalable quantum computers will threaten classical cryptography because of efficient quantum algorithms, e.g., Grover's algorithm for DB search [Gro96] and Shor's algorithms for factorization and discrete logarithms [Sho97]. Hence, we study classical cryptography secure against quantum adversaries (see e.g., the technical report from NIST [CJL+16]). Moreover, several researchers studied stronger quantum adversaries that can mount *quantum superposition attacks*, that is, quantum adversaries that can obtain the result of quantum computations with secret. For example, the adversary can obtain $\sum_c \psi_c |c, D(k,c)\rangle$ by querying $\sum_c \psi_c |c\rangle$, where D is a decryption circuit of a symmetric-key encryption scheme and k is a secret key. There are several quantum superposition attacks that break classically-secure cryptographic primitives: Kuwakado and Morii [KM12] presented a quantum chosen-plaintext attack against the Even-Mansour construction of a block cipher if the inner permutation is publicly available as quantum oracle, which employed Simon's algorithm [Sim97] neatly. Kaplan, Leurent, Leverrier, and Naya-Plasencia [KLLN16] also studied quantum superposition attacks against several block ciphers and modes.[1] Boneh and Zhandry [BZ13b] also gave a block cipher that is secure against chosen-plaintext-and-ciphertext attacks but vulnerable against quantum chosen-ciphertext attacks.

The stronger attack model in which adversaries can issue quantum queries is worth investigating. We motivate to investigate this model from following arguments:

- If a source code containing secret information is available, then a quantum adversary can implement a quantum machine containing secret information by itself and mount quantum superposition attacks. For example, a reverse engineering of a physical machine containing secret information allows an adversary to obtain an obfuscated code containing secret information. Moreover, white-box cryptography and obfuscation allows us to publish an obfuscated code containing secret information [GHS16].[2]
- In the future, quantum machines and quantum channels will be ubiquitous. Protocols and primitives will handle quantum data as discussed in Damgård, Funder, Nielsen, and Salvail [DFNS14].
- Even if they handle classical data, we can consider the quantum-ubiquitous world as Boneh and Zhandry discussed [BZ13a, BZ13b]. In this world, the end-user device is quantum and, thus, the device should measure the final quantum state and output a classical information, which prevents the quantum superposition attacks. This last step would be eventually avoided by an implementation bug or be circumvented by a neat hack of a quantum adversary in the future.

[1] We also note that Anand, Targhi, Tabia, and Unruh [ATTU16] showed several modes are secure against quantum superposition attacks if the underlying block cipher is quantumly-secure PRF.

[2] This means that if there is quantum chosen-plaintext or quantum chosen-ciphertext attack that breaks a cryptographic scheme easily, we should not publish an obfuscated code by the white-box cryptography or obfuscation.

– Moreover, if they handle classical data and are implemented in classical machines, one can consider special techniques that force the classical machines behave quantumly. For example, Damgård, Funder, Nielsen, and Salvail [DFNS14] and Gagliardoni, Hülsing, and Schaffner [GHS16] discussed the 'frozen smart-card' scenario.

Security of PKE and KEM against Quantum Chosen-Ciphertext Attacks: Boneh and Zhandry [BZ13b] introduced the security against quantum chosen-ciphertext attacks (qCCA security in short) for public-key encryption (PKE), which is the security against quantum adversaries that make quantum decryption queries. Boneh and Zhandry [BZ13b] showed that a PKE scheme obtained by applying the Canetti-Halevi-Katz conversion [BCHK07] to an identity-based encryption (IBE) scheme and one-time signature is IND-qCCA-secure if the underlying IBE scheme is selectively-secure against quantum chosen-identity queries and the underlying one-time signature scheme is (classically) strongly, existentially unforgeable against chosen-message attacks. They also showed that if there exists an IND-CCA-secure PKE, then there exists an ill-formed PKE that is IND-CCA-secure but not IND-qCCA-secure [BZ13b].

As far as we know, this is the only known PKE scheme that is proven to be IND-qCCA secure (excluding the concurrent work by Zhandry [Zha18, 2018-08-14 ver.]).

1.1 Our Contribution

We show that the SXY conversion in Saito, Yamakawa, and Xagawa[SXY18] and the HU conversion proposed by Jiang, Zhang, and Ma [JZM19] turn a PKE scheme into an IND-qCCA-secure KEM scheme in the QROM, if the underlying PKE scheme is perfectly-correct and disjoint-simulatable. We also observed that the perfect correctness can be relaxed as δ-correctness with negligible δ [HHK17].

Our idea is summarized as follows: In the last step of the IND-CCA security proofs of the above conversions, the challenger should simulate the decapsulation oracle on a query of any ciphertext c except the challenge ciphertext c^*. Roughly speaking, we observe that, if this simulation is "history-free," i.e., if the simulation does not depend on previously made queries at all, this procedure can be quantumly simulated by implementing this procedure in the quantum way. [3] For example, in the last step of the IND-CCA security proof in [SXY18], the decapsulation oracle on input c returns $K = \mathsf{H}_q(c)$ if $c \neq c^*$, where H_q is a random function chosen by the reduction algorithm. Therefore, intuitively speaking, this simulation is "history-free" and can be implemented quantumly.

1.2 Concurrent Works

Zhandry [Zha18, 2018-08-14 ver.] showed that the PKE scheme obtained by applying the Fujisaki-Okamoto conversion [FO13] to a PKE scheme PKE and a DEM scheme DEM is IND-qCCA-secure in the QROM, if PKE is OW-CPA-secure and well-spread, DEM is OT-secure [4]. Zhandry proposed recording and testing techniques to simulate the decryption oracles. We note that his security proof is non-tight unlike ours.

1.3 Organizations

section 2 reviews basic notations and definitions. section 3 reviews security notions of PKE and KEM. section 4 gives our new qCCA-security proof for the KEM in [SXY18] as known as the SXY conversion. section 5 gives our new qCCA-security proof for the KEM in [JZM19] as known as the HU conversion.

2 Preliminaries

2.1 Notation

A security parameter is denoted by κ. We use the standard O-notations: O, Θ, Ω, and ω. DPT and PPT stand for deterministic polynomial time and probabilistic polynomial time. A function $f(\kappa)$ is said to be *negligible* if

[3] Boneh et al. [BDF+11] defined history-free reductions for signature schemes. They also discussed the difficulties to model history-free reductions in the case of (public-key) encryption schemes. We also do not define history-free property of reductions for KEMs.

[4] Any efficient adversary cannot distinguish $E(k, m_0)$ from $E(k, m_1)$ even if it chooses m_0 and m_1 with $|m_0| = |m_1|$.

$f(\kappa) = \kappa^{-\omega(1)}$. We denote a set of negligible functions by $\mathrm{negl}(\kappa)$. For two finite sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathrm{Map}(\mathcal{X}, \mathcal{Y})$ denote a set of all functions whose domain is $\mathcal{X}$ and codomain is $\mathcal{Y}$.

For a distribution χ, we often write "$x \leftarrow \chi$," which indicates that we take a sample x from χ. For a finite set S, $U(S)$ denotes the uniform distribution over S. We often write "$x \leftarrow S$" instead of "$x \leftarrow U(S)$." For a set S and a deterministic algorithm A, $\mathsf{A}(S)$ denotes the set $\{\mathsf{A}(x) \mid x \in S\}$.

If inp is a string, then "out $\leftarrow \mathsf{A}(\mathsf{inp})$" denotes the output of algorithm A when run on input inp. If A is deterministic, then out is a fixed value and we write "$\mathsf{out} := \mathsf{A}(\mathsf{inp})$." We also use the notation "$\mathsf{out} := \mathsf{A}(\mathsf{inp}; r)$" to make the randomness r explicit.

For the Boolean statement P, $\mathrm{boole}(P)$ denotes the bit that is 1 if P is true, and 0 otherwise. For example, $\mathrm{boole}(b' = b)$ is 1 if and only if $b' = b$.

2.2 Quantum Computation

We refer to [NCoo] for basic of quantum computation.

Quantum Random Oracle Model. Roughly speaking, the quantum random oracle model (QROM) is an idealized model where a hash function is modeled as a publicly and quantumly accessible random oracle. See [BDF+11] for a more detailed description of the model.

Lemma. We review useful lemmas regarding the quantum oracles.

Lemma 2.1. *Let ℓ be an integer. Let $\mathsf{H}\colon \{0,1\}^\ell \times \mathcal{X} \to \mathcal{Y}$ and $\mathsf{H}'\colon \mathcal{X} \to \mathcal{Y}$ be two independent random oracles. If an unbounded time quantum adversary $\mathcal{A}$ makes a query to H at most q_{H} times, then we have*

$$\left|\Pr[s \leftarrow \{0,1\}^\ell : \mathcal{A}^{\mathsf{H},\mathsf{H}(s,\cdot)}() \to 1] - \Pr[\mathcal{A}^{\mathsf{H},\mathsf{H}'}() \to 1]\right| \le 2q_{\mathsf{H}} \cdot 2^{-\ell/2},$$

where all oracle accesses of $\mathcal{A}$ can be quantum. [5]

Though this seems to be a folklore, Saito et al. [SXY18] and Jiang et al. [JZC+18] gave the proof.

The second one is the hardness of generic search problem. If the oracle F rarely returns 1, then it is hard to distinguish F from the zero oracle N.

Lemma 2.2 (Generic Search Problem ([ARU14, Lemma 37], [HRS16, Thm.1], [JZC+18])). *Let $\gamma \in [0,1]$. Let $\mathcal{Z}$ be a finite set. Let $F\colon \mathcal{Z} \to \{0,1\}$ be the following function: For each z, $F(z) = 1$ with probability p_z at most γ and $F(z) = 0$ else. Let N be the zero function, that is, $N(z) = 0$ for any $z \in \mathcal{Z}$. If an oracle algorithm $\mathcal{A}$ makes at most Q quantum queries to F (or N), then*

$$\left|\Pr[\mathcal{A}^F() \to 1] - \Pr[\mathcal{A}^N() \to 1]\right| \le 2q\sqrt{\gamma}.$$

Particularly, the probability that $\mathcal{A}$ finds a z satisfying $F(z) = 1$ is at most $2q\sqrt{\gamma}$.

Simulation of Random Oracle. In the original quantum random oracle model introduced by Boneh et al. [BDF+11], they do not allow a reduction algorithm to access a random oracle, so it has to simulate a random oracle by itself. In contrast, in this paper, we give a random oracle access to a reduction algorithm. We remark that this is just a convention and not a modification of the model since we can simulate a random oracle against quantum adversaries in several ways; 1) $2q$-wise independent hash function [Zha12], where q is the maximum nubmer of queries to the random oracle, 2) quantumly-secure PRF [BDF+11], and 3) hash function modeled as quantum random oracle [KLS18]. In addition, Zhandry proposed a new technique to simulate the quantum random oracle, the compressed oracle technique [Zha18]. His new simulation of the quantum random oracle is perfect even for *unbounded* number of queries. In what follows, we use t_{RO} to denote a time needed to simulate a quantum random oracle.

[5] 23 Aug. 2021: We correct the upper bound $q_{\mathsf{H}} \cdot 2^{\frac{-\ell+1}{2}}$ to $2q_{\mathsf{H}} \cdot 2^{-\ell/2}$. See [SXY18, ePrint version].

3 Definitions

3.1 Public-Key Encryption (PKE)

The model for PKE schemes is summarized as follows:

Definition 3.1. *A PKE scheme* PKE *consists of the following triple of polynomial-time algorithms* (Gen, Enc, Dec).

- $\mathsf{Gen}(1^\kappa; r_g) \to (ek, dk)$*: a key-generation algorithm that on input* 1^κ*, where* κ *is the security parameter, outputs a pair of keys* (ek, dk)*.* ek *and* dk *are called the encryption key and decryption key, respectively.*
- $\mathsf{Enc}(ek, m; r_e) \to c$*: an encryption algorithm that takes as input encryption key* ek *and message* $m \in \mathcal{M}$ *and outputs ciphertext* $c \in \mathcal{C}$*.*
- $\mathsf{Dec}(dk, c) \to m/\perp$*: a decryption algorithm that takes as input decryption key* dk *and ciphertext* c *and outputs message* $m \in \mathcal{M}$ *or a rejection symbol* $\perp \notin \mathcal{M}$*.*

Definition 3.2. *We say a PKE scheme* PKE *is deterministic if* Enc *is deterministic. DPKE stands for deterministic public key encryption.*

We review δ-correctness in Hofheinz, Hövelmanns, and Kiltz [HHK17].

Definition 3.3 (δ-Correctness [HHK17]). *Let* $\delta = \delta(\kappa)$*. We say that* PKE $= (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ-correct *if*

$$\mathop{\mathrm{Ex}}_{(ek,dk)\leftarrow\mathsf{Gen}(1^\kappa)}\left[\max_{m\in\mathcal{M}} \Pr[c \leftarrow \mathsf{Enc}(ek, m) : \mathsf{Dec}(dk, c) \neq m]\right] \leq \delta(\kappa).$$

In particular, we say that PKE *is* perfeclty correct *if* $\delta = 0$*.*

We also define key's accuracy.

Definition 3.4 (Accuracy). *We say that a key pair* (ek, dk) *is* accurate *if for any* $m \in \mathcal{M}$*,*

$$\Pr[c \leftarrow \mathsf{Enc}(ek, m) : \mathsf{Dec}(dk, c) = m] = 1.$$

Remark 3.1. We observe that if PKE is deterministic, then δ-correctness implies that

$$\mathop{\mathrm{Ex}}_{(ek,dk)\leftarrow\mathsf{Gen}(1^\kappa)}[(ek, dk) \text{ is inaccurate}] \leq \delta(\kappa).$$

In other words, if PKE is deterministic and δ-correct, then a key pair is accurate with probability $\geq 1 - \delta$. We finally stress that, if PKE is deterministic but derandomized by the random oracle, then we cannot apply the above argument.

Disjoint Simulatability Saito et al. defined *disjoint simulatability* of DPKE [SXY18]. Intuitively speaking, a DPKE scheme is disjoint-simulatable if there exists a simulator that is only given an encryption key and generates a "fake ciphertext" that is computationally indistinguishable from a real ciphertext of a random message. Moreover, we require that a fake ciphertext falls in a valid ciphertext space with negligible probability. The formal definition is as follows.

Definition 3.5 (Disjoint simulatability [SXY18]). *Let* $\mathcal{D}_\mathcal{M}$ *denote an efficiently sampleable distribution on a set* $\mathcal{M}$*. A deterministic PKE scheme* PKE $= (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with plaintext and ciphertext spaces* $\mathcal{M}$ *and* $\mathcal{C}$ *is* $\mathcal{D}_\mathcal{M}$-disjoint-simulatable *if there exists a PPT algorithm* $\mathcal{S}$ *that satisfies the followings.*

- *(Statistical disjointness:)*

$$\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) := \max_{(ek,dk)\in\mathsf{Gen}(1^\kappa;\mathcal{R})} \Pr[c \leftarrow \mathcal{S}(ek) : c \in \mathsf{Enc}(ek, \mathcal{M})]$$

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

$$\mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_\mathcal{M},\mathcal{S},\mathcal{A}}(\kappa) := \left| \Pr\left[\begin{array}{c}(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); m^* \leftarrow \mathcal{D}_\mathcal{M};\\ c^* := \mathsf{Enc}(ek, m^*) : \mathcal{A}(ek, c^*) \to 1\end{array}\right] - \Pr\left[(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); c^* \leftarrow \mathcal{S}(ek) : \mathcal{A}(ek, c^*) \to 1\right] \right|$$

  is negligible.

$\underline{\mathsf{Expt}^{\text{ind-qcca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0,1\}$

$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$

$(m_0, m_1, st) \leftarrow \mathcal{A}_1^{\mathrm{QDEC}_\perp(\cdot)}(ek)$

$c^* \leftarrow \mathsf{Enc}(ek, m_b)$

$b' \leftarrow \mathcal{A}_2^{\mathrm{QDEC}_{c^*}(\cdot)}(c^*, st)$

return $\mathsf{boole}(b' = b)$

$\underline{\mathrm{QDEC}_a(\sum_{c,z} \phi_{c,z} |c, z\rangle)}$

return $\sum_{c,z} \phi_{c,z} |c, z \oplus f_a(c)\rangle$

$\underline{f_a(c)}$

$m := \mathsf{Dec}(dk, c)$

if $c = a$, set $m := \perp$

return m

Fig. 1: Game for PKE schemes

IND-QCCA Boneh and Zhandry showed that if we consider a quantum challenge oracle, then there exists a quantum adversary that can distinguish the superposition of plaintexts [BZ13b]. They showed that indistinguishability against fully-quantum chosen-plaintext attack (IND-FQCPA) and indistinguishability against fully-quantum chosen-left-right-plaintext attack (IND-FQLRCPA) is impossible. (For the details, see their paper [BZ13b].) Thus, we only consider a classical challenge oracle.

We need to define the result of $m \oplus \perp$, where $\perp \notin \mathcal{M}$. In order to do so, we encode $\perp$ as a bit string outside of the message space. The security definition follows:

Definition 3.6 (IND-QCCA for PKE [BZ13b]). *For any adversary $\mathcal{A}$, we define its* IND-QCCA *advantages against a PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

$$\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) := \left|2\Pr[\mathsf{Expt}^{\text{ind-qcca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1] - 1\right|,$$

where $\mathsf{Expt}^{\text{ind-qcca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$ *is an experiment described in Figure 1.*[6] *We say that* PKE *is* IND-QCCA*-secure if* $\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$ *is negligible for any PPT adversary $\mathcal{A}$.*

3.2 Key Encapsulation Mechanism (KEM)

The model for KEM schemes is summarized as follows:

Definition 3.7. *A KEM scheme* KEM *consists of the following triple of polynomial-time algorithms* (Gen, Encaps, Decaps)*:*

- $\mathsf{Gen}(1^\kappa; r_g) \to (ek, dk)$*: a key-generation algorithm that on input* 1^κ*, where* κ *is the security parameter, outputs a pair of keys* (ek, dk)*. ek and dk are called the encapsulation key and decapsulation key, respectively.*
- $\mathsf{Encaps}(ek; r_e) \to (c, K)$*: an encapsulation algorithm that takes as input encapsulation key ek and outputs ciphertext $c \in \mathcal{C}$ and key $K \in \mathcal{K}$.*
- $\mathsf{Decaps}(dk, c) \to K/\perp$*: a decapsulation algorithm that takes as input decapsulation key dk and ciphertext c and outputs key K or a rejection symbol $\perp \notin \mathcal{K}$.*

Definition 3.8 (δ-Correctness). *Let $\delta = \delta(\kappa)$. We say that* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ *is* δ-correct *if*

$$\Pr[(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); (c, K) \leftarrow \mathsf{Encaps}(ek) : \mathsf{Decaps}(dk, c) \neq K] \leq \delta(\kappa).$$

In particular, we say that KEM *is* perfeclty correct *if* $\delta = 0$.

IND-qCCA We also define indistinguishability under *quantum chosen-ciphertext attacks* (denoted by IND-QCCA) for KEM by follwoing [BZ13b].

Definition 3.9 (IND-QCCA for KEM). *For any adversary $\mathcal{A}$, we define its* IND-QCCA *advantage against a KEM scheme* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ *as follows:*

$$\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) := \left|2\Pr[\mathsf{Expt}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1] - 1\right|,$$

where $\mathsf{Expt}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ *is an experiment described in Figure 2.*[7]

We say that KEM *is* IND-QCCA*-secure if* $\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ *is negligible for any PPT adversary $\mathcal{A}$.*

[6] 23 Aug. 2021: We change $|\Pr[\ldots] - 1/2|$ to $|2\Pr[\ldots] - 1|$.

[7] 23 Aug. 2021: We change $|\Pr[\ldots] - 1/2|$ to $|2\Pr[\ldots] - 1|$.

$\underline{\mathrm{Expt}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0,1\}$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$
$(c^*, K_0^*) \leftarrow \mathsf{Encaps}(ek)$;
$K_1^* \leftarrow \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\mathrm{QDEC}_{c^*}(\cdot)}(ek, c^*, K_b^*)$
return $\mathsf{boole}(b' = b)$

$\underline{\mathrm{QDEC}_a(\sum_{c,z} \phi_{c,z} |c,z\rangle)}$

return $\sum_{c,z} \phi_{c,z} |c, z \oplus f_a(c)\rangle$

$\underline{f_a(c)}$

$K := \mathsf{Decaps}(dk, c)$
if $c = a$, set $K := \bot$
return K

Fig. 2: Game for KEM schemes

Table 1: Summary of Games for the Proof of Theorem 4.1

Game	H	c^*	K_0^*	K_1^*	Decryption of valid c	Decryption of invalid c	justification
Game_0	$\mathsf{H}(\cdot)$	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{H}'(s,c)$	
Game_1	$\mathsf{H}(\cdot)$	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	Lemma 2.1
$\mathsf{Game}_{1.5}$	$\mathsf{H}'_q(\mathsf{Enc}_1(ek,\cdot))$	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	if key is accurate
Game_2	$\mathsf{H}_q(\mathsf{Enc}_1(ek,\cdot))$	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	if key is accurate
Game_3	$\mathsf{H}_q(\mathsf{Enc}_1(ek,\cdot))$	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	if key is accurate
Game_4	$\mathsf{H}_q(\mathsf{Enc}_1(ek,\cdot))$	$\mathcal{S}(ek)$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	DS-IND

4 IND-qCCA Security of SXY

$\underline{\overline{\mathsf{Gen}}(1^\kappa)}$

$(ek, dk) \leftarrow \mathsf{Gen}_1(1^\kappa)$
$s \leftarrow \{0,1\}^\ell$
$\overline{dk} \leftarrow (dk, ek, s)$
return $(ek, \overline{dk})$

$\underline{\overline{\mathsf{Enc}}(ek)}$

$m \leftarrow \mathcal{D}_\mathcal{M}$
$c := \mathsf{Enc}_1(ek, m)$
$K := \mathsf{H}(m)$
return (K, c)

$\underline{\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, s)}$

$m := \mathsf{Dec}_1(dk, c)$
if $m = \bot$, **return** $K := \mathsf{H}'(s, c)$
if $c \neq \mathsf{Enc}_1(ek, m)$, **return** $K := \mathsf{H}'(s, c)$
else return $K := \mathsf{H}(m)$

Fig. 3: $\mathsf{KEM} := \mathsf{SXY}[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$.

Let $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ be a deterministic PKE scheme and let $\mathsf{H}\colon \mathcal{M} \to \mathcal{K}$ and $\mathsf{H}'\colon \{0,1\}^\ell \times \mathcal{C} \to \mathcal{K}$ be random oracles. We review the conversion SXY in Figure 3. We show that $\mathsf{KEM} := \mathsf{SXY}[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$ is IND-qCCA-secure if the underlying PKE_1 is a disjoint-simulatable DPKE.

Theorem 4.1 (IND-qCCA security of SXY in the QROM). *Let PKE_1 be a δ-correct DPKE scheme. Suppose that PKE_1 is $\mathcal{D}_\mathcal{M}$-disjoint-simulatable with a simulator $\mathcal{S}$. For any IND-qCCA quantum adversary $\mathcal{A}$ against KEM issuing q_H and $q_{\mathsf{H}'}$ quantum random oracle queries to H and H' and $q_{\overline{\mathsf{Dec}}}$ decapsulation queries, there exists an adversary $\mathcal{B}$ against the disjoint simulatability of PKE_1 such that*

$$\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) \le 2\mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1,\mathcal{D}_\mathcal{M},\mathcal{S},\mathcal{B}}(\kappa) + \mathsf{Disj}_{\mathsf{PKE}_1,\mathcal{S}}(\kappa) + 4\delta + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2}$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q_\mathsf{H} \cdot \mathsf{Time}(\mathsf{Enc}_1) + (q_\mathsf{H} + q_{\mathsf{H}'} + q_{\overline{\mathsf{Dec}}}) \cdot t_{\mathsf{RO}}$.

We note that the proof of Theorem 4.1 is essentially equivalent to that of the CCA security in the QROM in [SXY18] except that at the final game we require quantum simulation of decapsulation oracle.

Security Proof. We use a game-hopping proof. The overview of all games is given in Table 1.

Game_0: This is the original game, $\mathsf{Expt}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$.

Game_1: This game is the same as Game_0 except that $\mathsf{H}'(s,c)$ in the decapsulation oracle is replaced with $\mathsf{H}_q(c)$ where $\mathsf{H}_q : \mathcal{C} \to \mathcal{K}$ is another random oracle. We remark that $\mathcal{A}$ is not given direct access to H_q.

$\mathsf{Game}_{1.5}$: This game is the same as Game_1 except that the random oracle $\mathsf{H}(\cdot)$ is simulated by $\mathsf{H}'_q(\mathsf{Enc}_1(ek,\cdot))$ where H'_q is yet another random oracle. We remark that a decapsulation oracle and generation of K^*_0 also use $\mathsf{H}'_q(\mathsf{Enc}_1(ek,\cdot))$ as $\mathsf{H}(\cdot)$ and that $\mathcal{A}$ is not given direct access to H'_q.

Game_2: This game is the same as $\mathsf{Game}_{1.5}$ except that the random oracle $\mathsf{H}(\cdot)$ is simulated by $\mathsf{H}_q(\mathsf{Enc}_1(ek,\cdot))$ instead of $\mathsf{H}'_q(\mathsf{Enc}_1(ek,\cdot))$. We remark that the decapsulation oracle and generation of K^*_0 also use $\mathsf{H}_q(\mathsf{Enc}_1(ek,\cdot))$ as $\mathsf{H}(\cdot)$.

Game_3: This game is the same as Game_2 except that K^*_0 is set as $\mathsf{H}_q(c^*)$ and the decapsulation oracle always returns $\mathsf{H}_q(c)$ as long as $c \neq c^*$. We denote the modified decapsulation oracle by $\textsc{qDec}'$.

Game_4: This game is the same as Game_3 except that c^* is set as $\mathcal{S}(ek)$.

The above completes the descriptions of games. We clearly have

$$\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 2|\Pr[\mathsf{Game}_0 = 1] - 1/2|$$

by the definition. We upperbound this by the following lemmas.

Lemma 4.1. *We have*

$$|\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_1 = 1]| \leq q_{\mathsf{H}'} \cdot 2^{\frac{-\ell+1}{2}}.$$

Proof. This is obvious from Lemma 2.1. □

Lemma 4.2. *Let* Acc *and* $\overline{\mathsf{Acc}}$ *denote the event that the key pair* (ek, dk) *is accurate and inaccurate, respectively. We have*

$$|\Pr[\mathsf{Game}_1 = 1] - 1/2| \leq |\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}] - 1/2| + \delta.$$

Proof. By the definition, we have

$$\Pr[\mathsf{Acc}] \geq 1 - \delta \text{ and } \Pr[\overline{\mathsf{Acc}}] \leq \delta.$$

We have

$$\begin{aligned}
&|\Pr[\mathsf{Game}_1 = 1] - 1/2| \\
&= \left|\Pr[\overline{\mathsf{Acc}}] \cdot \Pr[\mathsf{Game}_1 = 1 \mid \overline{\mathsf{Acc}}] + \Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}] - 1/2\right| \\
&\leq \Pr[\overline{\mathsf{Acc}}] \cdot \Pr[\mathsf{Game}_1 = 1 \mid \overline{\mathsf{Acc}}] + |\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}] - 1/2| \\
&\leq \Pr[\overline{\mathsf{Acc}}] + |\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}] - 1/2| \\
&\leq |\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}] - 1/2| + \delta
\end{aligned}$$

as we wanted. □

Lemma 4.3. *We have*

$$\Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}] = \Pr[\mathsf{Game}_{1.5} = 1 \mid \mathsf{Acc}].$$

Proof. Since we assume that the key pair (ek, dk) of PKE_1 is accurate, $\mathsf{Enc}_1(ek,\cdot)$ is injective. Therefore, if $\mathsf{H}'_q(\cdot)$ is a random function, then $\mathsf{H}'_q(\mathsf{Enc}_1(ek,\cdot))$ is also a random function. Remarking that access to H'_q is not given to $\mathcal{A}$, it causes no difference from the view of $\mathcal{A}$ if we replace $\mathsf{H}(\cdot)$ with $\mathsf{H}'_q(\mathsf{Enc}_1(ek,\cdot))$. □

Lemma 4.4. *We have*

$$\Pr[\mathsf{Game}_{1.5} = 1 \mid \mathsf{Acc}] = \Pr[\mathsf{Game}_2 = 1 \mid \mathsf{Acc}].$$

Proof. We say that a ciphertext c is valid if we have $\mathsf{Enc}_1(ek, \mathsf{Dec}_1(dk, c)) = c$ and invalid otherwise. We remark that H_q is used only for decrypting an invalid ciphertext c as $\mathsf{H}_q(c)$ in $\mathsf{Game}_{1.5}$. This means that a value of $\mathsf{H}_q(c)$ for a valid c is not used at all in $\mathsf{Game}_{1.5}$.

On the other hand, any output of $\mathsf{Enc}_1(ek, \cdot)$ is valid due to the accuracy of (ek, dk). Since H'_q is only used for evaluating an output of $\mathsf{Enc}_1(ek, \cdot)$, a value of $\mathsf{H}'_q(c)$ for an invalid c is not used at all in $\mathsf{Game}_{1.5}$.

Hence, it causes no difference from the view of $\mathcal{A}$ if we use the same random oracle H_q instead of two independent random oracles H_q and H'_q. □

Lemma 4.5. *We have*

$$\Pr[\mathsf{Game}_2 = 1 \mid \mathsf{Acc}] = \Pr[\mathsf{Game}_3 = 1 \mid \mathsf{Acc}].$$

Proof. Since we set $\mathsf{H}(\cdot) := \mathsf{H}_q(\mathsf{Enc}_1(ek, \cdot))$, for any valid c and $m := \mathsf{Dec}_1(dk, c)$, we have $\mathsf{H}(m) = \mathsf{H}_q(\mathsf{Enc}_1(ek, m)) = \mathsf{H}_q(c)$. Therefore, responses of the decapsulation oracle are unchanged. We also have $\mathsf{H}(m^*) = \mathsf{H}_q(c^*)$. □

Lemma 4.6. *We have*

$$|\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \mathsf{Acc}] - 1/2| \leq |\Pr[\mathsf{Game}_3 = 1] - 1/2| + \delta.$$

Proof. We have

$$\begin{aligned}
&|\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \mathsf{Acc}] - 1/2| \\
&\leq \left| \begin{aligned} &\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \mathsf{Acc}] + \Pr[\overline{\mathsf{Acc}}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \overline{\mathsf{Acc}}] \\ &\qquad - \Pr[\overline{\mathsf{Acc}}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \overline{\mathsf{Acc}}] - 1/2 \end{aligned} \right| \\
&\leq \left| \Pr[\mathsf{Game}_3 = 1] - 1/2 - \Pr[\overline{\mathsf{Acc}}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \overline{\mathsf{Acc}}] \right| \\
&\leq |\Pr[\mathsf{Game}_3 = 1] - 1/2| + \Pr[\overline{\mathsf{Acc}}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \overline{\mathsf{Acc}}] \\
&\leq |\Pr[\mathsf{Game}_3 = 1] - 1/2| + \Pr[\overline{\mathsf{Acc}}] \\
&\leq |\Pr[\mathsf{Game}_3 = 1] - 1/2| + \delta.
\end{aligned}$$

In the third inequality, we use the fact that for any reals a, b, and c with $c \geq 0$, we have $|a - b - c| \leq |a - b| + c$. (See Lemma A.1 for the proof.) We use this inequality by setting $a = \Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \mathsf{Acc}]$, $b = 1/2$ and $c = \Pr[\overline{\mathsf{Acc}}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \overline{\mathsf{Acc}}]$. □

Lemma 4.7. *There exists a quantum adversary $\mathcal{B}$ such that*

$$|\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_4 = 1]| \leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1, \mathcal{D}_{\mathcal{M}}, \mathcal{S}, \mathcal{B}}(\kappa).$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q_{\mathsf{H}} \cdot \mathsf{Time}(\mathsf{Enc}_1) + (q_{\mathsf{H}} + q_{\mathsf{H}'} + q_{\overline{\mathsf{Dec}}}) \cdot t_{\mathsf{RO}}$.

Proof. We construct an adversary $\mathcal{B}$, which is allowed to access two random oracles H_q and H', against the disjoint simulatability as follows [8].

$\mathcal{B}^{\mathsf{H}_q, \mathsf{H}'}(ek, c^*)$: It picks $b \leftarrow \{0, 1\}$, sets $K_0^* := \mathsf{H}_q(c^*)$ and $K_1^* \leftarrow \mathcal{K}$, and invokes $b' \leftarrow \mathcal{A}^{\mathsf{H}, \mathsf{H}', \mathrm{QDEC'}}(ek, c^*, K_b^*)$ where $\mathcal{A}'s$ oracles are simulated as follows.

- $\mathsf{H}(\cdot)$ is simulated by $\mathsf{H}_q(\mathsf{Enc}_1(ek, \cdot))$.
- H' can be simulated because $\mathcal{B}$ has access to an oracle H'.
- QDEC'$(\cdot)$ is simulated by filtering c^* and using $\mathsf{H}_q(\cdot)$; that is, on input $\sum_{c,z} \phi_{c,z} |c, z\rangle$, $\mathcal{B}$ returns $\sum_{c \neq c^*, z} \phi_{c,z} |c, z \oplus \mathsf{H}_q(c)\rangle + \sum_z \phi_{c^*,z} |c^*, z \oplus \bot\rangle$.

Finally, $\mathcal{B}$ returns $\mathsf{boole}(b = b')$.

This completes the description of $\mathcal{B}$. It is easy to see that $\mathcal{B}$ perfectly simulates Game_3 if $c^* = \mathsf{Enc}_1(ek, m^*)$ and Game_4 if $c^* = \mathcal{S}(ek)$. Therefore, we have

$$|\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_4 = 1]| \leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1, \mathcal{D}_{\mathcal{M}}, \mathcal{S}, \mathcal{B}}(\kappa)$$

as wanted. Since H is simulated by one evaluation of Enc_1 plus one evaluation of a random oracle H_q, and H' and QDEC' are simulated by one evaluation of random oracles, we have $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q_{\mathsf{H}} \cdot \mathsf{Time}(\mathsf{Enc}_1) + (q_{\mathsf{H}} + q_{\mathsf{H}'} + q_{\overline{\mathsf{Dec}}}) \cdot t_{\mathsf{RO}}$. □

[8] We allow a reduction algorithm to access the random oracles. See subsection 2.2 for details.

Lemma 4.8. *We have*

$$|2\Pr[\mathsf{Game}_4 = 1] - 1| \le \mathsf{Disj}_{\mathsf{PKE}_1,\mathcal{S}}(\kappa).$$

Proof. Let Bad denote the event that c^* is in $\mathsf{Enc}_1(ek, \mathcal{M})$ in Game_4. It is easy to see that we have

$$\Pr[\mathsf{Bad}] \le \mathsf{Disj}_{\mathsf{PKE}_1,\mathcal{S}}(\kappa).$$

When Bad does not occur, i.e., $c^* \notin \mathsf{Enc}_1(ek, \mathcal{M})$, $\mathcal{A}$ obtains no information about $K_0^* = \mathsf{H}_q(c^*)$. This is because queries to H only reveal $\mathsf{H}_q(c)$ for $c \in \mathsf{Enc}_1(ek, \mathcal{M})$, and $\textsc{qDec}'(c)$ returns $\perp$ if $c = c^*$. Therefore, we have

$$\Pr[\mathsf{Game}_4 = 1 \mid \overline{\mathsf{Bad}}] = 1/2.$$

Combining the above, we have

$$\begin{aligned}
&|2\Pr[\mathsf{Game}_4 = 1] - 1| \\
&= \left|\Pr[\mathsf{Bad}] \cdot 2(\Pr[\mathsf{Game}_4 = 1 \mid \mathsf{Bad}] - 1/2) + \Pr[\overline{\mathsf{Bad}}] \cdot 2(\Pr[\mathsf{Game}_4 = 1 \mid \overline{\mathsf{Bad}}] - 1/2)\right| \\
&\le \Pr[\mathsf{Bad}] \cdot |2\Pr[\mathsf{Game}_4 = 1 \mid \mathsf{Bad}] - 1| + \Pr[\overline{\mathsf{Bad}}] \cdot 2\left|\Pr[\mathsf{Game}_4 = 1 \mid \overline{\mathsf{Bad}}] - 1/2\right| \\
&\le \Pr[\mathsf{Bad}] + 2 \cdot \left|\Pr[\mathsf{Game}_4 = 1 \mid \overline{\mathsf{Bad}}] - 1/2\right| \\
&\le \mathsf{Disj}_{\mathsf{PKE}_1,\mathcal{S}}(\kappa)
\end{aligned}$$

as we wanted. □

Proof (Proof of Theorem 4.1). Combining all lemmas in this section, we obtain the following inequality:

$$\begin{aligned}
\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) &= |2\Pr[\mathsf{Game}_0 = 1] - 1| \\
&\le 2|\Pr[\mathsf{Game}_1 = 1] - 1/2| + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2} \\
&\le 2|\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}] - 1/2| + 2\delta + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2} \\
&= 2|\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_{1.5} = 1 \mid \mathsf{Acc}] - 1/2| + 2\delta + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2} \\
&= 2|\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_2 = 1 \mid \mathsf{Acc}] - 1/2| + 2\delta + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2} \\
&= 2|\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_3 = 1 \mid \mathsf{Acc}] - 1/2| + 2\delta + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2} \\
&\le 2|\Pr[\mathsf{Game}_3 = 1] - 1/2| + 4\delta + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2} \\
&\le 2|\Pr[\mathsf{Game}_4 = 1] - 1/2| + 2\mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1,\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{B}}(\kappa) + 4\delta + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2} \\
&\le \mathsf{Disj}_{\mathsf{PKE}_1,\mathcal{S}}(\kappa) + 2\mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1,\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{B}}(\kappa) + 4\delta + 4q_{\mathsf{H}'} \cdot 2^{-\ell/2}.
\end{aligned}$$

□

5 IND-qCCA Security of HU

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_1, c_2))$, where $\overline{dk} = (dk, ek)$
$(ek, dk) \leftarrow \mathsf{Gen}_1(1^\kappa)$	$m \leftarrow \mathcal{D}_{\mathcal{M}}$	$m := \mathsf{Dec}_1(dk, c_1)$
$dk \leftarrow (dk, ek)$	$c_1 := \mathsf{Enc}_1(ek, m)$	**if** $m = \perp$, **return** $K := \perp$
return $(ek, \overline{dk})$	$c_2 := \mathsf{H}'(m)$	**if** $c_1 \neq \mathsf{Enc}_1(ek, m)$, **return** $K := \perp$
	$K := \mathsf{H}(m)$	**if** $c_2 \neq \mathsf{H}'(m)$, **return** $K := \perp$
	return $(K, (c_1, c_2))$	**else return** $K := \mathsf{H}(m)$

Fig. 4: $\mathsf{KEM} := \mathsf{HU}[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$.

Table 2: Summary of Games for the Proof of Theorem 5.1. We let $g(\cdot) = \mathsf{Enc}_1(ek, \cdot)$.

Game	H	H′	c_1^*	c_2^*	K_0^*	K_1^*	Decryption K	condition	justification
Game_0	H	H′	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}'(m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	if $c_1 = \mathsf{Enc}_1(ek, m)$ and $c_2 = \mathsf{H}'(m)$	
Game_1	$\mathsf{H}_q \circ g$	$\mathsf{H}'_q \circ g$	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}'_q(c_1^*)$	$\mathsf{H}_q(c_1^*)$	random	$\mathsf{H}(m)$	if $c_1 = \mathsf{Enc}_1(ek, m)$ and $c_2 = \mathsf{H}'(m)$	if key is accurate
Game_2	$\mathsf{H}_q \circ g$	$\mathsf{H}'_q \circ g$	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}'_q(c_1^*)$	$\mathsf{H}_q(c_1^*)$	random	$\mathsf{H}_q(c_1)$	if $c_1 = \mathsf{Enc}_1(ek, m)$ and $c_2 = \mathsf{H}'_q(c_1)$	if key is accurate
Game_3	$\mathsf{H}_q \circ g$	$\mathsf{H}'_q \circ g$	$\mathsf{Enc}_1(ek, m^*)$	$\mathsf{H}'_q(c_1^*)$	$\mathsf{H}_q(c_1^*)$	random	$\mathsf{H}_q(c_1)$	if $c_2 = \mathsf{H}'_q(c_1)$	Statistical
Game_3	$\mathsf{H}_q \circ g$	$\mathsf{H}'_q \circ g$	$\mathcal{S}(ek)$	$\mathsf{H}'_q(c_1^*)$	$\mathsf{H}_q(c_1^*)$	random	$\mathsf{H}_q(c_1)$	if $c_2 = \mathsf{H}'_q(c_1)$	DS-IND

Very recently, Jiang, Zhang, and Ma [JZM19] proposed a conversoin HU, which allows an explicit rejection but requires additional hash value c_2 of m. Let $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ be a deterministic PKE scheme and let $\mathsf{H}\colon \mathcal{M} \to \mathcal{K}$ and $\mathsf{H}'\colon \mathcal{M} \to \mathcal{H}$ be random oracles. We review the conversion HU in Figure 4. We show that $\mathsf{KEM} := \mathsf{HU}[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$ is IND-QCCA-secure if the underlying PKE_1 is a disjoint-simulatable DPKE.

Theorem 5.1 (IND-QCCA security of HU in the QROM). *Let PKE_1 be a δ-correct DPKE scheme. Suppose that PKE_1 is $\mathcal{D}_{\mathcal{M}}$-disjoint-simulatable with a simulator $\mathcal{S}$. For any IND-QCCA quantum adversary $\mathcal{A}$ against KEM issuing q_{H} and $q_{\mathsf{H}'}$ quantum random oracle queries to H and H′ and $q_{\overline{\mathsf{Dec}}}$ decapsulation queries, there exists an adversary $\mathcal{B}$ against the disjoint simulatability of PKE_1 such that*

$$\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) \le 2\mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1,\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{B}}(\kappa) + \mathsf{Disj}_{\mathsf{PKE}_1,\mathcal{S}}(\kappa) + 4q_{\overline{\mathsf{Dec}}}|\mathcal{H}|^{-1/2} + 4\delta$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + (q_{\mathsf{H}} + q_{\mathsf{H}'}) \cdot \mathsf{Time}(\mathsf{Enc}_1) + (q_{\mathsf{H}} + q_{\mathsf{H}'} + 2q_{\overline{\mathsf{Dec}}}) \cdot t_{\mathsf{RO}}$.

The proof of Theorem 5.1 follows.

Security Proof. We use a game-hopping proof. The overview of all games is given in Table 2.

Game_0: This is the original game, $\mathsf{Expt}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$.

Game_1: This game is the same as Game_0 except that the random oracle $\mathsf{H}(\cdot)$ and $\mathsf{H}'(\cdot)$ are simulated by $\mathsf{H}_q(\mathsf{Enc}_1(ek, \cdot))$ and $\mathsf{H}'_q(\mathsf{Enc}_1(ek, \cdot))$, respectively, where $\mathsf{H}_q\colon \mathcal{C} \to \mathcal{K}$ and $\mathsf{H}'_q\colon \mathcal{C} \to \mathcal{H}$ are random oracles. We remark that a decapsulation oracle and generation of K_0^* also use $\mathsf{H}_q(\mathsf{Enc}_1(ek, \cdot))$ as $\mathsf{H}(\cdot)$, and generation of c_2^* uses $\mathsf{H}'_q(\mathsf{Enc}_1(ek, \cdot))$ as $\mathsf{H}'(\cdot)$. We also remark that $\mathcal{A}$ is not given direct access to H_q and H'_q.

Game_2: This game is the same as Game_1 except that the decapsulation oracle returns $K := \mathsf{H}_q(c_1)$ if $c_1 = \mathsf{Enc}_1(ek, m)$ and $\mathsf{H}'_q(c_1) = c_2$, instead returns $K := \mathsf{H}(m)$ if $c_1 = \mathsf{Enc}_1(ek, m)$ and $\mathsf{H}'(m) = c_2$.

Game_3: This game is the same as Game_2 except that the decapsulation oracle returns $K := \mathsf{H}_q(c_1)$ if $\mathsf{H}'_q(c_1) = c_2$. That is, the decapsulation oracle never use the re-encryption check.

Game_4: This game is the same as Game_3 except that c_1^* is set as $\mathcal{S}(ek)$.

The above completes the descriptions of games. We clearly have

$$\mathsf{Adv}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 2|\Pr[\mathsf{Game}_0 = 1] - 1/2|$$

by the definition. We upperbound this by the following lemmas.

Lemma 5.1. *Let* Acc *denote the event that the key pair (ek, dk) is accurate. We have*

$$|\Pr[\mathsf{Game}_0 = 1] - 1/2| \le |\Pr[\mathsf{Acc}] \cdot \Pr[\mathsf{Game}_0 = 1 \mid \mathsf{Acc}] - 1/2| + \delta.$$

We omit the proof, since the proof is the same as that of Lemma 4.2.

Lemma 5.2. *We have*

$$\Pr[\mathsf{Game}_0 = 1 \mid \mathsf{Acc}] = \Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}].$$

Proof. Since we assume that the key pair is accurate, $\mathsf{Enc}_1(ek,\cdot)$ is injective. Therefore, if $\mathsf{H}_q(\cdot)$ (and $\mathsf{H}'_q(\cdot)$, resp.) is a random function, then $\mathsf{H}_q(\mathsf{Enc}_1(ek,\cdot))$ (and $\mathsf{H}'_q(\mathsf{Enc}_1(ek,\cdot))$, resp.) is also a random function. Remarking that access to H_q and H'_q is not given to $\mathcal{A}$, it causes no difference from the view of $\mathcal{A}$ if we replace $\mathsf{H}(\cdot)$ (and $\mathsf{H}'(\cdot)$, resp.) with $\mathsf{H}_q(\mathsf{Enc}_1(ek,\cdot))$ (and $\mathsf{H}'_q(\mathsf{Enc}_1(ek,\cdot))$, resp.). □

Lemma 5.3. *We have*

$$\Pr[\mathsf{Game}_1 = 1 \mid \mathsf{Acc}] = \Pr[\mathsf{Game}_2 = 1 \mid \mathsf{Acc}].$$

Proof. This change is just conceptual. Suppose that $c_1 = \mathsf{Enc}_1(ek,m)$. We have that $c_2 = \mathsf{H}'(m)$ holds if and only if $c_2 = \mathsf{H}'_q(c_1)$ and $K = \mathsf{H}(m) = \mathsf{H}_q(c_1)$. □

Lemma 5.4. *We have*

$$|\Pr[\mathsf{Game}_2 = 1 \mid \mathsf{Acc}] - \Pr[\mathsf{Game}_3 = 1 \mid \mathsf{Acc}]| \le 2q\sqrt{\epsilon_{\mathsf{Dec}}}|\mathcal{H}|^{-1/2}.$$

Proof. Recall that we have $\mathsf{H}'(m) = \mathsf{H}'_q(\mathsf{Enc}(ek,m))$ and $\mathsf{H}'_q(c_1) = c_2$.

Let us see the details how the decapsulation oracle treats the query $|c_1,c_2,z\rangle$. Let $m = \mathsf{Dec}_1(dk,c_1)$.

- Case 1 that $c_1 = \mathsf{Enc}_1(ek,m)$: in this case, the decapsulation oracles in both games return $|c_1,c_2,z\oplus K\rangle$, where $K := \mathsf{H}_q(c_1)$ or $\perp$ depending on that $c_2 = \mathsf{H}'_q(c_1)$.
- Case 2 that $c_1 \neq \mathsf{Enc}_1(ek,m)$ and $c_2 \neq \mathsf{H}'_q(c_2)$: In this case, the decapsulation oracles in both games return $|c_1,c_2,z\oplus\perp\rangle$.
- Case 3 that $c_1 \neq \mathsf{Enc}_1(ek,m)$ and $c_2 = \mathsf{H}'_q(c_1)$: In this case, the decapsulation oracle in Game_2 returns $|c_1,c_2,z\oplus\perp\rangle$, but the decapsulation oracle in Game_3 returns $|c_1,c_2,z\oplus\mathsf{H}_q(c_1)\rangle$.

If the query is classical, we can argue the difference as in [JZM19]: Since the adversary cannot access to H'_q directly, it cannot know the value of $\mathsf{H}'_q(c_1)$ if c_1 lies outside of $\mathsf{Enc}(ek,\cdot)$. Therefore, any c_2 hits the value $\mathsf{H}'_q(c_1)$ with probability at most $1/|\mathcal{H}|$.

Even if the query is quantum, the problem is distinguishing problem and we invoke Lemma 2.2. We now reduce from generic search problem to distinguishing Game_2 with Game_3. We define the distribution $\mathcal{D}_F$ over $F := \{f : \mathcal{C}\times\mathcal{H}\to\{0,1\}\}$ as follows: for each $c_1 \in \mathcal{C}$, choose $h_{c_1} \leftarrow \mathcal{H}$ uniformly at random and set

$$f(c_1,h) := \begin{cases} 1 & \text{if } h = h_{c_1} \\ 0 & \text{otherwise.} \end{cases}$$

For each (c_1,h), we have $\Pr[f(c_1,h)=1] \le |\mathcal{H}|^{-1}$.

The reduction algorithm is defined as follows: Suppose that we are given $f : \mathcal{C}\times\mathcal{H}\to\{0,1\}$, which is chosen according to $\mathcal{D}_F$ or set as the zero function N. We construct H, H', and the decapsulation oracle as follows:

- H_q and H'_q: we choose $\mathsf{H}_q|_{\mathsf{Enc}_1(ek,\mathcal{M})}$ and $\mathsf{H}'_q|_{\mathsf{Enc}_1(ek,\mathcal{M})}$ uniformly at random.
- H: on input $|m,z\rangle$, it returns $|m,z\oplus\mathsf{H}_q(\mathsf{Enc}_1(ek,m))\rangle$.
- H': on input $|m,z\rangle$, it returns $|m,z\oplus\mathsf{H}'_q(\mathsf{Enc}_1(ek,m))\rangle$.
- QDec_{c^*}: On input $|c_1,c_2,z\rangle$, it computes $m = \mathsf{Dec}_1(dk,c_1)$ and computes K as follows:
  - if $c_1 = c_1^*$ and $c_2 = c_2^*$, then let $K = \perp$.
  - if $c_1 = \mathsf{Enc}_1(ek,m)$ and $c_2 = \mathsf{H}'_q(c_1)$, then let $K = \mathsf{H}_q(c_1)$.
  - if $c_1 = \mathsf{Enc}_1(ek,m)$ and $c_2 \neq \mathsf{H}'_q(c_1)$, then let $K = \perp$.
  - if $c_1 \neq \mathsf{Enc}_1(ek,m)$ and $f(c_1,c_2) = 1$, then let $K = \mathsf{H}_q(c_1)$.
  - if $c_1 \neq \mathsf{Enc}_1(ek,m)$ and $f(c_1,c_2) = 0$, then let $K = \perp$.

  it returns $|c_1,c_2,z\oplus K\rangle$.

If $f = N$, then this algorithm perfectly simulates Game_2. On the other hand, if $f \leftarrow \mathcal{D}_F$, then this algorithm perfectly simulates Game_3, since any adversary cannot access H'_q on $\mathcal{C}\setminus\mathsf{Enc}(ek,\mathcal{M})$. Thus, according to Lemma 2.2, we have upperbound $2q\sqrt{\epsilon_{\mathsf{Dec}}}|\mathcal{H}|^{-1/2}$ as we wanted. □

Lemma 5.5. *We have*

$$|\Pr[\mathsf{Acc}]\cdot\Pr[\mathsf{Game}_3 = 1 \mid \mathsf{Acc}] - 1/2| \le |\Pr[\mathsf{Game}_3 = 1] - 1/2| + \delta.$$

We omit the proof, since the proof is the same as that of Lemma 4.6.

Lemma 5.6. *There exists an adversary $\mathcal{B}$ such that*

$$|\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_4 = 1]| \leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1, \mathcal{D}_{\mathcal{M}}, \mathcal{S}, \mathcal{B}}(\kappa).$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q_{\mathsf{H}} \cdot \mathsf{Time}(\mathsf{Enc}_1) + (q_{\mathsf{H}} + q_{\mathsf{H}'} + q_{\overline{\mathsf{Dec}}}) \cdot t_{\mathsf{RO}}$.

Proof. Let $g(\cdot) := \mathsf{Enc}_1(ek, \cdot)$. For ease of notation, we define a new function $f_{\mathsf{H}_q, \mathsf{H}'_q} : \mathcal{C} \times \mathcal{H} \to \mathcal{K} \cup \{\perp\}$ as follows:

$$f_{\mathsf{H}_q, \mathsf{H}'_q}(c_1, c_2) := \begin{cases} \mathsf{H}_q(c_1) & \text{if } \mathsf{H}'_q(c_1) = c_2 \\ \perp & \text{otherwise.} \end{cases}$$

We construct an adversary $\mathcal{B}$, which is allowed to access two random oracles H_q and H'_q, against the disjoint simulatability as follows [9].

$\mathcal{B}^{\mathsf{H}_q, \mathsf{H}'_q}(ek, c_1^*)$: It picks $b \leftarrow \{0,1\}$, sets $K_0^* := \mathsf{H}_q(c_1^*)$ and $K_1^* \leftarrow \mathcal{K}$, and invokes $b' \leftarrow \mathcal{A}^{\mathsf{H}, \mathsf{H}', \overline{\mathsf{Dec}}'}(ek, c_1^*, K_b^*)$ where $\mathcal{A}'s$ oracles are simulated as follows.

- $\mathsf{H}(\cdot)$ is simulated by $\mathsf{H}_q(\mathsf{Enc}_1(ek, \cdot))$.
- $\mathsf{H}'(\cdot)$ is simulated by $\mathsf{H}'_q(\mathsf{Enc}_1(ek, \cdot))$.
- $\overline{\mathsf{Dec}}'(\cdot)$ is simulated by filtering c_1^*; on input $\sum_{c_1, c_2, z} \phi_{c_1, c_2, z} |c_1, c_2, z\rangle$, $\mathcal{B}$ returns

$$\sum_{c_1 \neq c_1^*, z} \phi_{c_1, c_2, z} |c_1, c_2, z \oplus f_{\mathsf{H}_q, \mathsf{H}'_q}(c_1, c_2)\rangle + \sum_{c_2, z} \phi_{c_1^*, c_2, z} |c_1^*, c_2, z \oplus \perp\rangle$$

Finally, $\mathcal{B}$ returns $\mathsf{boole}(b = b')$.

This completes the description of $\mathcal{B}$.

Since $c_2^* := \mathsf{H}_q(c_1^*)$, if $c_2 \neq c_2^*$, then the decapsulation oracle in both games and $f_{\mathsf{H}_q, \mathsf{H}'_q}$ return $\perp$ on input (c_1^*, c_2). Thus, we have

$$\sum_{c_2, z} \phi_{c_1^*, c_2, z} |c_1^*, c_2, z \oplus \perp\rangle = \sum_{c_2 \neq c_2^*, z} \phi_{c_1^*, c_2, z} |c_1^*, c_2, z \oplus \perp\rangle + \phi_{c_1^*, c_2^*, z} |c_1^*, c_2^*, z \oplus \perp\rangle$$

and $\mathcal{B}$ perfectly simulate the decapsulation oracle.

It is easy to see that $\mathcal{B}$ perfectly simulates Game_3 if $c_1^* = \mathsf{Enc}_1(ek, m^*)$ and Game_4 if $c_1^* \leftarrow \mathcal{S}(ek)$. Therefore, we have

$$|\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_4 = 1]| \leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1, \mathcal{D}_{\mathcal{M}}, \mathcal{S}, \mathcal{B}}(\kappa)$$

as wanted. We have $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + (q_{\mathsf{H}} + q_{\mathsf{H}'}) \cdot \mathsf{Time}(\mathsf{Enc}_1) + (q_{\mathsf{H}} + q_{\mathsf{H}'} + 2q_{\overline{\mathsf{Dec}}}) \cdot t_{\mathsf{RO}}$, since $\mathcal{B}$ invokes $\mathcal{A}$ once, H is simulated by one evaluation of Enc_1 plus one evaluation of a random oracle, and H' and $\overline{\mathsf{Dec}}'$ are simulated by two evaluations of random oracles. □

Lemma 5.7. *We have*

$$|2\Pr[\mathsf{Game}_4 = 1] - 1| \leq \mathsf{Disj}_{\mathsf{PKE}_1, \mathcal{S}}(\kappa).$$

Proof. Let Bad denote the event that $c_1^* \in \mathsf{Enc}_1(ek, \mathcal{M})$ happens in Game_4. It is easy to see that we have

$$\Pr[\mathsf{Bad}] \leq \mathsf{Disj}_{\mathsf{PKE}_1, \mathcal{S}}(\kappa).$$

When Bad does not occur, i.e., $c_1^* \notin \mathsf{Enc}_1(ek, \mathcal{M})$, $\mathcal{A}$ obtains no information about $K_0^* = \mathsf{H}_q(c_1^*)$. This is because queries to H only reveal $\mathsf{H}_q(c)$ for $c \in \mathsf{Enc}_1(ek, \mathcal{M})$, and $\overline{\mathsf{Dec}}'(c)$ returns $\perp$ if $c = c_1^*$. Therefore, we have

$$\Pr[\mathsf{Game}_4 = 1 \mid \overline{\mathsf{Bad}}] = 1/2.$$

[9] We allow a reduction algorithm to access the random oracles. See subsection 2.2 for details.

Combining the above, we have

$$\begin{aligned}
&|2\Pr[\mathrm{Game}_4 = 1] - 1| \\
&= \left|\Pr[\mathrm{Bad}] \cdot 2(\Pr[\mathrm{Game}_4 = 1 \mid \mathrm{Bad}] - 1/2) + \Pr[\overline{\mathrm{Bad}}] \cdot 2(\Pr[\mathrm{Game}_4 = 1 \mid \overline{\mathrm{Bad}}] - 1/2)\right| \\
&\leq \Pr[\mathrm{Bad}] + 2\left|\Pr[\mathrm{Game}_4 = 1 \mid \overline{\mathrm{Bad}}] - 1/2\right| \\
&\leq \mathrm{Disj}_{\mathsf{PKE}_1,\mathcal{S}}(\kappa)
\end{aligned}$$

as we wanted. □

Proof (Proof of Theorem 5.1). Combining all lemmas in this section, we obtain the following inequality:

$$\begin{aligned}
\mathrm{Adv}^{\text{ind-qcca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) &= 2|\Pr[\mathrm{Game}_0 = 1] - 1/2| \\
&\leq 2|\Pr[\mathrm{Acc}] \cdot \Pr[\mathrm{Game}_0 = 1 \mid \mathrm{Acc}] - 1/2| + 2\delta \\
&= 2|\Pr[\mathrm{Acc}] \cdot \Pr[\mathrm{Game}_1 = 1 \mid \mathrm{Acc}] - 1/2| + 2\delta \\
&= 2|\Pr[\mathrm{Acc}] \cdot \Pr[\mathrm{Game}_2 = 1 \mid \mathrm{Acc}] - 1/2| + 2\delta \\
&\leq 2|\Pr[\mathrm{Acc}] \cdot \Pr[\mathrm{Game}_3 = 1 \mid \mathrm{Acc}] - 1/2| + 4q_{\overline{\mathrm{Dec}}}|\mathcal{H}|^{-1/2} + 2\delta \\
&\leq 2|\Pr[\mathrm{Game}_3 = 1] - 1/2| + 4q_{\overline{\mathrm{Dec}}}|\mathcal{H}|^{-1/2} + 4\delta \\
&\leq 2|\Pr[\mathrm{Game}_4 = 1] - 1/2| + 2\mathrm{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1,\mathcal{D}_M,\mathcal{S},\mathcal{B}}(\kappa) + 4q_{\overline{\mathrm{Dec}}}|\mathcal{H}|^{-1/2} + 4\delta \\
&\leq \mathrm{Disj}_{\mathsf{PKE}_1,\mathcal{S}}(\kappa) + 2\mathrm{Adv}^{\text{ds-ind}}_{\mathsf{PKE}_1,\mathcal{D}_M,\mathcal{S},\mathcal{B}}(\kappa) + 4q_{\overline{\mathrm{Dec}}}|\mathcal{H}|^{-1/2} + 4\delta.
\end{aligned}$$

□

Acknowledgments

We would like to thank Haodong Jiang and anonymous reviewers of PQCrypto 2019 for insightful comments.

References

ARU14. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th FOCS*, pages 474–483. IEEE Computer Society Press, October 2014. 3

ATTU16. Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, pages 44–63. Springer, Heidelberg, 2016. 1

BCHK07. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007. 2

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011. 2, 3

BZ13a. Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, Heidelberg, May 2013. 1

BZ13b. Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013. 1, 2, 5

CJL+16. Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Technical report, National Institute of Standards and Technology (NIST), 2016. 1

DFNS14. Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition attacks on cryptographic protocols. In Carles Padró, editor, *ICITS 13*, volume 8317 of *LNCS*, pages 142–161. Springer, Heidelberg, 2014. 1, 2

FO13. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013. 2

GHS16. Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 60–89. Springer, Heidelberg, August 2016. 1, 2

Gro96. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *28th ACM STOC*, pages 212–219. ACM Press, May 1996. 1

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, Heidelberg, November 2017. 2, 4

HRS16. Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 387–416. Springer, Heidelberg, March 2016. 3

JZC+18. Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 96–125. Springer, Heidelberg, August 2018. 3

JZM19. Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model, 2019. To appear PKC 2019. 2, 10, 11

KLLN16. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 207–237. Springer, Heidelberg, August 2016. 1

KLS18. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, Heidelberg, April / May 2018. 3

KM12. Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type even-mansour cipher. In *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012*, pages 312–316. IEEE, 2012. 1

NC00. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 3

Sho97. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. 1

Sim97. Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997. 1

SXY18. Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 520–551. Springer, Heidelberg, April / May 2018. 2, 3, 4, 6

Zha12. Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, August 2012. 3

Zha18. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. Cryptology ePrint Archive, Report 2018/276, 2018. https://eprint.iacr.org/2018/276. 2, 3

A Simple Lemma

Lemma A.1. *For any reals a, b, and c with $c \geq 0$, we have*

$$|a - b - c| \leq |a - b| + c.$$

Proof. We consider the three cases below:

- Case $a - b \geq c \geq 0$: In this case, we have $a - b - c \geq 0$. Thus, we have $|a - b - c| = a - b - c \leq a - b + c = |a - b| + c$.
- Case $a - b \leq 0 \leq c$: In this case, we have $a - b - c \leq 0$. We have $|a - b - c| = -(a - b - c) = -(a - b) + c = |a - b| + c$.
- Case $0 \leq a - b \leq c$: Again, we have $a - b - c \leq 0$. We have $|a - b - c| = -(a - b - c) = -(a - b) + c \leq a - b + c = |a - b| + c$.

In all three cases, we have $|a - b - c| \leq |a - b| + c$ as we wanted. □