Tighter QCCA-Secure Key Encapsulation Mechanism with Explicit Rejection in the Quantum Random Oracle Model

Jiangxia Ge[1,2], Tianshu Shan[1,2], and Rui Xue[1,2](✉)

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
{gejiangxia, shantianshu, xuerui}@iie.ac.cn

Abstract. Hofheinz et al. (TCC 2017) proposed several key encapsulation mechanism (KEM) variants of Fujisaki-Okamoto (FO) transformation, including $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}_m^{\not\perp}$, $\mathsf{QFO}_m^{\not\perp}$, $\mathsf{FO}^{\perp}$, $\mathsf{FO}_m^{\perp}$ and $\mathsf{QFO}_m^{\perp}$, and they are widely used in the post-quantum cryptography standardization launched by NIST. These transformations are divided into two types, the implicit and explicit rejection type, including $\{\mathsf{FO}^{\not\perp}, \mathsf{FO}_m^{\not\perp}, \mathsf{QFO}_m^{\not\perp}\}$ and $\{\mathsf{FO}^{\perp}, \mathsf{FO}_m^{\perp}, \mathsf{QFO}_m^{\perp}\}$, respectively. The decapsulation algorithm of the implicit (resp. explicit) rejection type returns a pseudorandom value (resp. an abort symbol $\perp$) for an invalid ciphertext.

For the implicit rejection type, the $\mathsf{IND\text{-}CCA}$ security reduction of $\mathsf{FO}^{\not\perp}$ in the quantum random oracle model (QROM) can avoid the quadratic security loss, as shown by Kuchta et al. (EUROCRYPT 2020). However, for the explicit rejection type, the best known $\mathsf{IND\text{-}CCA}$ security reduction in the QROM presented by Hövelmanns et al. (ASIACRYPT 2022) for $\mathsf{FO}_m^{\perp}$ still suffers from a quadratic security loss. Moreover, it is not clear until now whether the implicit rejection type is more secure than the explicit rejection type.

In this paper, a QROM security reduction of $\mathsf{FO}_m^{\perp}$ without incurring a quadratic security loss is provided. Furthermore, our reduction achieves $\mathsf{IND\text{-}qCCA}$ security, which is stronger than the $\mathsf{IND\text{-}CCA}$ security. To achieve our result, two steps are taken: The first step is to prove that the $\mathsf{IND\text{-}qCCA}$ security of $\mathsf{FO}_m^{\perp}$ can be tightly reduced to the $\mathsf{IND\text{-}CPA}$ security of $\mathsf{FO}_m^{\perp}$ by using the online extraction technique proposed by Don et al. (EUROCRYPT 2022). The second step is to prove that the $\mathsf{IND\text{-}CPA}$ security of $\mathsf{FO}_m^{\perp}$ can be reduced to the $\mathsf{IND\text{-}CPA}$ security of the underlying public key encryption (PKE) scheme without incurring quadratic security loss by using the Measure-Rewind-Measure One-Way to Hiding Lemma (EUROCRYPT 2020).

In addition, we prove that (at least from a theoretic point of view), security is independent of whether the rejection type is explicit ($\mathsf{FO}_m^{\perp}$) or implicit ($\mathsf{FO}_m^{\not\perp}$) if the underlying PKE scheme is weakly γ-spread.

Keywords: Fujisaki-Okamoto transformation · quantum random oracle · key encapsulation mechanism · quantum chosen-ciphertext attack.

1 Introduction

The Fujisaki-Okamoto (FO) transformation [11] combines a public key encryption (PKE) scheme and a symmetric key encryption (SKE) scheme to obtain a hybrid scheme that is secure against the indistinguishability under chosen-ciphertext attacks ($\mathsf{IND\text{-}CCA}$) in the random oracle model (ROM) [2]. It is known as the first generic transformation from an arbitrary $\mathsf{OW\text{-}CPA}$-secure PKE to an $\mathsf{IND\text{-}CCA}$-secure PKE in the ROM. Dent [8] introduced the first key encapsulation mechanism (KEM) variant of FO obtaining an $\mathsf{IND\text{-}CCA}$-secure KEM in the ROM. Hofheinz et al. [14] provided a fine-grained and modular toolkit of transformations including $\mathsf{T}, \mathsf{U}^{\not\perp}, \mathsf{U}^{\perp}, \mathsf{U}^{\not\perp}_m$, $\mathsf{U}^{\perp}_m, \mathsf{QU}^{\not\perp}_m$ and $\mathsf{QU}^{\perp}_m$. They then presented the KEM variants of FO as $\mathsf{FO}^{\not\perp}, \mathsf{FO}^{\perp}, \mathsf{FO}^{\not\perp}_m, \mathsf{FO}^{\perp}_m, \mathsf{QFO}^{\not\perp}_m$ and $\mathsf{QFO}^{\perp}_m$ by combining T with $\mathsf{U}^{\not\perp}, \mathsf{U}^{\perp}, \mathsf{U}^{\not\perp}_m, \mathsf{U}^{\perp}_m, \mathsf{QU}^{\not\perp}_m$ and $\mathsf{QU}^{\perp}_m$, respectively. Here $\not\perp$ (resp. $\perp$) indicates that the transformation belongs to the implicit (resp. explicit) rejection type, which means that a pseudorandom value (resp. an abort symbol $\perp$) is returned if the ciphertext fails to decapsulate. In what follows, we refer to above KEM variants of FO as FO-like transformations.

As FO-like transformations are frequently used in the NIST post-quantum cryptography standardisation process [23], the post-quantum security of FO-like transformations have drawn much attention. In the post-quantum setting, the ROM should be lifted to the quantum random oracle model (QROM) [4], and thus the $\mathsf{IND\text{-}CCA}$ security reduction of FO-like transformations in the QROM is more concerned. To this problem, a sequence of works has been given [16,17,18]. The core tool used in their reductions is the One-Way to Hiding (O2H) Lemma [1,26], and their reductions all suffer from a quadratic security loss.

For the implicit rejection type of FO-like transformations, Kuchta et al. proposed a new O2H variant named Measure-Rewind-Measure One-Way to Hiding (MRM O2H) Lemma [20], with which an $\mathsf{IND\text{-}CCA}$ security reduction of $\mathsf{FO}^{\not\perp}$ in the QROM avoiding the quadratic security loss is provided. For the explicit rejection type of FO-like transformations, the best known reduction is provided by Hövelmanns et al. [15]. They proved the $\mathsf{IND\text{-}CCA}$ security of $\mathsf{FO}^{\perp}_m$ in the QROM and their reduction still suffers from a quadratic security loss. The core tool used in their reduction is a new O2H variant named semi-classical OWTH in the eQROM$_f$, which can be considered as the combination of the extractable RO-simulator [9] and the semi-classical O2H [1].

In addition to avoiding the quadratic security loss, Xagawa and Yamakawa [27] also considered the QROM security of FO-like transformations against quantum adversaries that can mount quantum superposition queries to the decapsulation oracle. They introduced a new security notion for KEM named indistinguishability under quantum chosen-ciphertext attacks ($\mathsf{IND\text{-}qCCA}$) by following the notion of Boneh and Zhandry [5], and provided an $\mathsf{IND\text{-}qCCA}$ security reduction of SXY in the QROM. Here SXY designed in [27] is identical to $\mathsf{U}^{\not\perp}_m$. Liu and Wang [21] modified the definition of disjoint simulatability secure proposed in [27] and applied the MRM O2H lemma to prove that the transformation KC defined in [24] can transform a $\mathsf{OW\text{-}CPA}$-secure deterministic public key encryption

(DPKE) scheme with correctness errors into a modified disjoint simulatability secure PKE scheme. Furthermore, they proved that transformation $\mathsf{SXY} \circ \mathsf{KC}$ and $\mathsf{SXY} \circ \mathsf{KC} \circ \mathsf{T}$ can also achieve the IND-qCCA security.

Compared with the implicit rejection type, the explicit rejection type of FO-like transformations is more natural and has a positive performance on the robustness [13]. Unfortunately, the best known QROM reduction of the explicit rejection type FO-like transformations provided by Hövelmanns et al. [15] still suffers from a quadratic security loss, and their IND-CCA security reduction seems to be insufficient to prove the IND-qCCA security[3]. Hence, a natural question arises:

Is it possible to give an IND-qCCA security reduction of the explicit rejection type of FO-like transformations in the QROM avoiding quadratic security loss?

In addition, the impact of the different rejection type of the FO-like transformations on the security of the final scheme is also discussed in the literature. Bindel et al. [3] proved that the transformation $\mathsf{FO}^{\not\perp}$ (resp. $\mathsf{FO}^{\perp}$) is secure iff $\mathsf{FO}_m^{\not\perp}$ (resp. $\mathsf{FO}_m^{\perp}$) is secure. They also showed that the security of $\mathsf{FO}_m^{\perp}$ implies security of $\mathsf{FO}_m^{\not\perp}$, and that the security of $\mathsf{FO}_m^{\not\perp}$ implies security of $\mathsf{QFO}_m^{\perp}$. Further, Hövelmanns et al. [15] showed that the security of $\mathsf{FO}_m^{\perp}$ implies security of all remaining FO-like transformations. However, it is not clear until now whether the security of $\mathsf{FO}_m^{\not\perp}$ implies security of $\mathsf{FO}_m^{\perp}$, and thus the results of [3] and [15] do not imply that the implicit rejection type of FO-like transformations is as secure as their explicit rejection counterparts. Therefore, there still exists an open problem on the implicit and explicit rejection types of FO-like transformations as follows:

Is the explicit rejection type as secure as their implicit rejection counterparts? In other words, does the security of $\mathsf{FO}_m^{\not\perp}$ imply the security of $\mathsf{FO}_m^{\perp}$?

1.1 Our Contribution

Avoiding the quadratic security loss, an IND-qCCA security reduction of $\mathsf{FO}_m^{\perp}$ in the QROM is provided (Corollary 1), and the corresponding security bound is shown in Table 1.1. Compared with security bounds of $\mathsf{FO}_m^{\perp}$ provided in [9,15], our security bound of $\mathsf{FO}_m^{\perp}$ is much tighter, and we achieve a stronger (IND-qCCA) security[4] with the same or even weaker requirements.

[3] Indeed, in the IND-CCA security reduction of [15], **Game $\mathbf{G_1}$** records the decapsulation query c_i $(i = 1, \ldots, q_D)$ and computes $\mathsf{eCO.E}(c_i)$ for each c_i via the extraction interface eCO.E in its end. The record procedure is available in the IND-CCA security reduction. However, due to the quantum no-cloning principle, it is infeasible to perfectly record the quantum decapsulation queries in the IND-qCCA security reduction.

[4] If a PKE/KEM scheme is IND-qCCA-secure, it is also IND-CCA-secure, because classical decryption/decapsulation queries can be implemented by quantum decryption/decapsulation queries. That is why we say that IND-qCCA security is a stronger security.

Table 1. Security bounds of different transformations in the QROM. Here q is the total number of query times to the random oracles, d and w is the query depth and query width of the random oracles, q_D is the adversary's query times to the decapsulation oracle. ϵ is the security bound of the underlying PKE scheme P.

Transformation	Underlying security	Achieved security	Requirement	Security bound($\approx$)
$\mathsf{FO}_m^{\perp}$ [9]	OW-CPA	IND-CCA	P is weakly γ-spread	$q \cdot \sqrt{\epsilon}$
$\mathsf{FO}_m^{\perp}$ [15]	OW-CPA	IND-CCA	P is γ-spread	$(d + q_D) \cdot \sqrt{w \cdot \epsilon}$
$\mathsf{FO}_m^{\perp}$ Our work	IND-CPA	IND-qCCA	P is weakly γ-spread	$d(d + q_D) \cdot \epsilon$

Moreover, in the QROM, we prove that $\mathsf{FO}_m^{\not\perp}$ is IND-qCCA-secure if $\mathsf{FO}_m^{\perp}$ is IND-qCCA-secure (Theorem 5), and conversely that $\mathsf{FO}_m^{\perp}$ is IND-qCCA-secure if $\mathsf{FO}_m^{\not\perp}$ is IND-qCCA-secure (Theorem 6).

In more detail, in the proof of Theorem 5, we tightly reduce the IND-qCCA security of $\mathsf{FO}_m^{\not\perp}$ to the IND-qCCA security of $\mathsf{FO}_m^{\perp}$.

As for the Theorem 6, let $(\epsilon^{\perp}, T^{\perp}, S^{\perp})$ denote the success probability, running time and memory space of an adversary against the IND-qCCA security of $\mathsf{FO}_m^{\perp}$, respectively, and let $(\epsilon^{\not\perp}, T^{\not\perp}, S^{\not\perp})$ denote the success probability, running time and memory space of a reduction algorithm against the IND-qCCA security of $\mathsf{FO}_m^{\not\perp}$, respectively. In the proof of Theorem 6, suppose that the underlying PKE scheme is weakly γ-spread, we prove that (Here q_D and q is the notion used in Table 1.1.)

$$\epsilon^{\perp} \leq \epsilon^{\not\perp} + O(q_D \cdot 2^{-\gamma/2}), \quad T^{\not\perp} \approx T^{\perp} + O(q^2), \quad S^{\not\perp} \approx S^{\perp} + O(q).$$

This indicates that the IND-qCCA security of $\mathsf{FO}_m^{\perp}$ can be reduced to the IND-qCCA security of $\mathsf{FO}_m^{\not\perp}$ with an additional error of $O(q_D \cdot 2^{-\gamma/2})$, a quadratic running time expansion, and a linear space expansion of the reduction algorithm.

Overall, assuming that the underlying PKE scheme is weakly γ-spread, it can be concluded that the explicit rejection type of FO-like transformations is as secure as their implicit rejection counterparts. This implies that the security of FO-like transformations is independent of the rejection type if the underlying PKE scheme is weakly γ-spread.

1.2 Technical Overview

Our IND-qCCA security reduction of $\mathsf{FO}_m^{\perp}$ in the QROM can be decomposed into two steps as shown in Fig. 1:

1. In the first step, we prove that, in the QROM, the IND-qCCA security of $\mathsf{FO}_m^{\perp}$ can be tightly reduced to the IND-CPA security of $\mathsf{FO}_m^{\perp}$ (Theorem 2).
2. In the second step, we prove that, in the QROM, $\mathsf{U}_m^{\perp}$ can transform a OW-CPA-secure DPKE scheme dPKE into an IND-CPA-secure KEM scheme $\mathsf{U}_m^{\perp}[\mathsf{dPKE}]$ without the quadratic security loss (Theorem 3). Then combining with Lemma 8 and the property that $\mathsf{FO}_m^{\perp} = \mathsf{U}_m^{\perp} \circ \mathsf{T}$, we prove that, in

the QROM, the $\mathsf{IND\text{-}CPA}$ security of $\mathsf{FO}_m^{\perp}$ can be reduced to the $\mathsf{IND\text{-}CPA}$ security of the underlying randomized PKE scheme P without the quadratic security loss.

$$\begin{array}{c}\mathsf{IND\text{-}CPA}\\ \mathsf{FO}_m^{\perp}\end{array} \xrightarrow{\text{Theorem } 2} \begin{array}{c}\mathsf{IND\text{-}qCCA}\\ \mathsf{FO}_m^{\perp}\end{array}$$

$$\begin{array}{c}\mathsf{IND\text{-}CPA}\\ \delta\text{-correct } \mathsf{P}\end{array} \xrightarrow[\text{Lemma } 8\ [3]]{\mathsf{T}} \begin{array}{c}\mathsf{OW\text{-}CPA}\\ \mathsf{dPKE}\end{array} \xrightarrow[\text{Theorem } 3]{\mathsf{U}_m^{\perp}} \begin{array}{c}\mathsf{IND\text{-}CPA}\\ \mathsf{U}_{\mathrm{m}}^{\perp}[\mathsf{dPKE}]\end{array}$$

Fig. 1. Two steps of the $\mathsf{IND\text{-}qCCA}$ security reduction of $\mathsf{FO}_m^{\perp}$ in the QROM.

Here we first consider the second step. Using the MRM O2H lemma, it is straightforward to prove Theorem 3. We stress that this lemma requires the simulator simulates both H and G and we circumvent this problem by using the Lemma 4 in [21] (i.e. Lemma 9 in our paper.).

For the first step, we prove Theorem 2 via a series of hybrid games from $\mathbf{G_0}$ to $\mathbf{G_6}$, where game $\mathbf{G_0}$ is the $\mathsf{IND\text{-}qCCA}$ game of $\mathsf{FO}_m^{\perp}$ with adversary $\mathcal{A}$ in the QROM. Define $\mathrm{Adv}(\mathbf{G_i}, \mathbf{G_{i+1}}) := |\Pr[1 \leftarrow \mathbf{G_i}] - \Pr[1 \leftarrow \mathbf{G_{i+1}}]|$ for $i = 0, \ldots, 5$. In the proof of Theorem 2, our basic idea is to analyze the upper bound of $\mathrm{Adv}(\mathbf{G_i}, \mathbf{G_{i+1}})$ for $i = 0, \ldots, 5$, and finally construct an $\mathsf{IND\text{-}CPA}$ adversary $\tilde{\mathcal{A}}$ against $\mathsf{FO}_m^{\perp}$ in the QROM by the adversary $\mathcal{A}$ in game $\mathbf{G_6}$. The overview of games $\mathbf{G_1}$ to $\mathbf{G_6}$ are as follows.

- Game $\mathbf{G_1}$ is identical with $\mathbf{G_0}$, except the extractable RO-simulator $\mathcal{S}(f_1) := \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_1}\}$ is introduced and the quantum queries to random oracle H is simulated by the RO-interface $\mathsf{eCO.RO}$. In game $\mathbf{G_1}$, $\mathcal{A}$'s quantum queries to H have been recorded in database imperfectly.
- From game $\mathbf{G_2}$ to $\mathbf{G_3}$, we gradually change the simulation of the quantum accessible decapsulation oracle, and finally simulate it without secret key sk in game $\mathbf{G_3}$.
- From game $\mathbf{G_4}$ to $\mathbf{G_6}$, our aim is to make the database just before adversary $\mathcal{A}$ performs its operation be irrelevant to the challenge plaintext m^*.

In the following, we describe the difference between every two adjacent games of games $\mathbf{G_1}, \ldots, \mathbf{G_6}$ and analyze them at a high level.

Game $\mathbf{G_1}$-$\mathbf{G_2}$: In order to simulate the quantum accessible decapsulation oracle qDeca without sk, our idea is to use the extraction-interface of the extractable RO-simulator to read out the information recorded in the database and prepare replies to the qDeca. We emphasize that this simulating can only read the database and cannot update or change it. However, the simulation of qDeca in game $\mathbf{G_1}$ has no such limitation because it can query H (which is simulated by eCO.RO) and update the database at certain points. Therefore, we design the following game $\mathbf{G_2}$ in our proof to clarify the error produced when changing the simulation of qDeca from updating the database to reading it.

– Game $\mathbf{G_2}$: This game is the same as game $\mathbf{G_1}$, except that the operation $\mathsf{eCO.E}_{f_1} \circ O_G \circ \mathsf{eCO.E}_{f_1}$ as shown in Fig. 2 is used to simulate qDeca.

Here $\mathsf{eCO.E}_{f_1}$ maps $|c, D, m\rangle$ to $|c, D, m \oplus x\rangle$, $x = \mathsf{Dec}_{sk}(c)$ if $\mathsf{Dec}_{sk}(c) \neq \perp$ and $\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c), D(\mathsf{Dec}_{sk}(c))) = c$. Otherwise $x = \perp$[5]. Operation O_G simulates the random oracle G and we set $G(\perp) = \perp$.

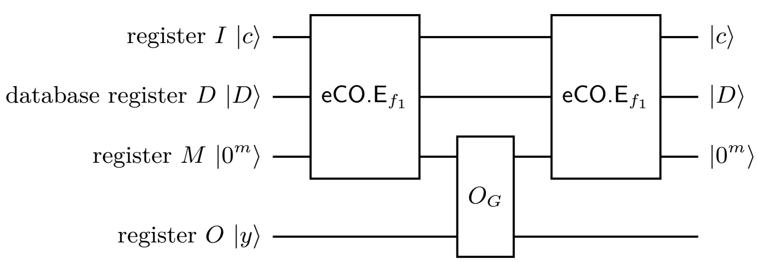

Fig. 2. Operation $\mathsf{eCO.E}_{f_1} \circ O_G \circ \mathsf{eCO.E}_{f_1}$. Here I/O is input/output register of qDeca, M is the internal register used by operation $\mathsf{eCO.E}_{f_1} \circ O_G \circ \mathsf{eCO.E}_{f_1}$.

For any computational basis state $|c, D, y\rangle$ on registers IDO that satisfies $\mathsf{Dec}_{sk}(c) \neq \perp$ and $D(\mathsf{Dec}_{sk}(c)) = \perp$, it is easily verified that the qDeca in game $\mathbf{G_2}$ returns state $|c, D, y \oplus \perp\rangle$ for input state $|c, D, y\rangle$ since $G(\perp) = \perp$. However, the qDeca in game $\mathbf{G_1}$ may not return $|c, D, y \oplus \perp\rangle$, because the simulation of qDeca in game $\mathbf{G_1}$ can update the database to a uniform superposition of database $D \cup (\mathsf{Dec}_{sk}(c), y)$ for $y \in \{0,1\}^n$.

The difference between game $\mathbf{G_1}$ and $\mathbf{G_2}$ above actually corresponds to the classical event GUESS in the ROM reduction of $\mathsf{FO}_m^{\perp}$ provided in [15], i.e., the adversary queries a ciphertext c to the decapsulation oracle satisfying that $\mathsf{Dec}_{sk}(c)(\neq \perp)$ is never queried to H before but $\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c), H(\mathsf{Dec}_{sk}(c))) = c$. The probability that GUESS occurs can be upper bounded by $2^{-\gamma}$ if the underlying PKE scheme is γ-spread, since $H(x)$ is uniformly random in $\{0,1\}^n$ if x is never queried to H, and the maximum number of elements y meeting $\mathsf{Enc}_{pk}(x, y) = c$ in $\{0,1\}^n$ is $2^{n-\gamma}$.

We analyze the difference between game $\mathbf{G_1}$ and $\mathbf{G_2}$ in a similar way, that is to say, even if the database is updated to a uniform superposition of database $D \cup (\mathsf{Dec}_{sk}(c), y)$ for $y \in \{0,1\}^n$ in game $\mathbf{G_1}$, there are not many $y \in \{0,1\}^n$ such that

$$\mathsf{eCO.E}_{f_1}|c, D \cup (\mathsf{Dec}_{sk}(c), y), m\rangle = |c, D \cup (\mathsf{Dec}_{sk}(c), y), m \oplus \mathsf{Dec}_{sk}(c)\rangle$$

[5] For simplify, we do not consider the case of $c = c^*$ here. c^* is the challenge ciphertext.

if the underlying PKE scheme is weakly γ-spread. We stress that we finally (upper) bound $\mathrm{Adv}(\mathbf{G_1}, \mathbf{G_2})$ by $8q_D \cdot 2^{-\gamma/2}$ since decapsulation oracle qDeca is quantum accessible in our reduction.

Game $\mathbf{G_2}$-$\mathbf{G_3}$: Game $\mathbf{G_3}$ is the same as game $\mathbf{G_2}$ except that the extractable RO-simulator is changed to $\mathcal{S}(f_2) := \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_2}\}$.

For computational basis state $|c, D, m\rangle$ on registers IDM, $\mathsf{eCO.E}_{f_2}$ extracts the minimum x satisfying $\mathsf{Enc}_{pk}(x, D(x)) = c$ and returns state $|c, D, m \oplus x\rangle$ if such x exists. Otherwise, returns state $|c, D, m \oplus \perp\rangle$. Note that the implementation of $\mathsf{eCO.E}_{f_2}$ does not need sk because it no longer cares about if above x also equals $\mathsf{Dec}_{sk}(c)$ like $\mathsf{eCO.E}_{f_1}$. However, $\mathsf{eCO.E}_{f_1}$ and $\mathsf{eCO.E}_{f_2}$ may have different effect on state $|c, D, m\rangle$ that triggers decryption errors (x exists s.t. $\mathsf{Enc}_{pk}(x, D(x)) = c$ but $x \neq \mathsf{Dec}_{sk}(c)$).

In the proof of Theorem 2, a database set $R^D_{pk,sk}$ is defined. We find that $\mathsf{eCO.E}_{f_1}$ and $\mathsf{eCO.E}_{f_2}$ have the same effect on state $|c, D, m\rangle$ if $D \notin R^D_{pk,sk}$. Then, we use the compressed semi-classical one-way to hiding theorem[6] proved in [12] to (upper) bound $\mathrm{Adv}(\mathbf{G_2}, \mathbf{G_3})$ by $O(q_H)\sqrt{\delta}$, where q_H is the query times to random oracle H and δ is the correctness error of the underlying PKE scheme.

Game $\mathbf{G_3}$-$\mathbf{G_4}$-$\mathbf{G_5}$: Note that game $\mathbf{G_3}$ uses operation $\mathsf{eCO.E}_{f_2} \circ O_G \circ \mathsf{eCO.E}_{f_2}$, which no longer needs sk, to simulate qDeca. However, the challenge ciphertext c^* ($= \mathsf{Enc}_{pk}(m^*, H(m^*))$) still needs classically query H (which is simulated using $\mathsf{eCO.RO}$) by challenge plaintext m^* to generate. The database state just before adversary $\mathcal{A}$ performs its operations in game $\mathbf{G_3}$ can be written as

$$\mathsf{StdDecomp}_{m^*}|D^\perp \cup (m^*, H(m^*))\rangle,$$

where database $D^\perp$ only contains $(\perp, 0^n)$ pairs, $\mathsf{StdDecomp}_{m^*}$ is the local decompression procedure defined in [29], and we also denote it as S_{m^*} in what follows for convenience. Obviously, this state contains the information of m^*, hence a new adversary without m^* unable to simulate game $\mathbf{G_3}$ for $\mathcal{A}$.

To circumvent this problem, our idea is as follows. Let O be a new random oracle that has the same input/output length as H, roughly speaking, if the extractable RO-simulator $\mathcal{S}(f_2)$ in game $\mathbf{G_3}$ perfectly simulates random oracle H at point m^*, we can equivalently compute c^* as $\mathsf{Enc}_{pk}(m^*, O(m^*))$ and the database state just before adversary $\mathcal{A}$ performs its operation at this time is irrelevant to m^*. What we need to do next is to ensure that $\mathcal{A}$ will get $O(m^*)$ accordingly when querying H (which is simulated using $\mathsf{eCO.RO}$) by m^* and design a simulation method for qDeca following the modification of the computation of c^*.

Unfortunately, the extractable RO-simulator $\mathcal{S}(f_2)$ in game $\mathbf{G_3}$ cannot perfectly simulate the random oracle H at point m^*. Note that state $\mathsf{S}_{m^*}|D^\perp \cup$

[6] Actually, this theorem is a generalization of the compress oracle O2H theorem (Theorem 10) in [7], since the quantum oracle algorithm in this theorem can also make database read queries.

$(m^*, H(m^*))\rangle$ is a superposition of $|D^\perp \cup (m^*, y)\rangle$ for $y \in \{0,1\}^n$ and $|D^\perp\rangle$ [29], the extraction-interface $\mathsf{eCO.E}_{f_2}$ used in game $\mathbf{G_3}$ may disturb this superposition state. Then, we design game $\mathbf{G_4}$ as follows in our reduction.

– Game $\mathbf{G_4}$: It is the same as game $\mathbf{G_3}$ except that S_{m^*} is performed before and after the applying of $\mathsf{eCO.E}_{f_2}$. Thus, a new extractable RO-simulator

$$\mathcal{S}'(f_2) := \{\mathsf{eCO.RO}, \mathsf{S}_{m^*} \circ \mathsf{eCO.E}_{f_2} \circ \mathsf{S}_{m^*}\}$$

is applied in this game.

The $\mathrm{Adv}(\mathbf{G_3}, \mathbf{G_4})$ can be easily upper bounded by using the operator norm $\|[\mathsf{eCO.E}_{f_2}, \mathsf{S}_{m^*}]\|$ since S_{m^*} is an involution [29].

In contrast to game $\mathbf{G_3}$, the extractable RO-simulator $\mathcal{S}'(f_2)$ in game $\mathbf{G_4}$ perfectly simulates the random oracle H at point m^*. Intuitively, the operation $\mathsf{S}_{m^*} \circ \mathsf{eCO.E}_{f_2} \circ \mathsf{S}_{m^*}$ seems to implement one classical compressed standard oracle query at point m^*, except that the operation CNOT is changed to $\mathsf{eCO.E}_{f_2}$. Indeed, it is precisely because of this query-like structure, $\mathsf{S}_{m^*} \circ \mathsf{eCO.E}_{f_2} \circ \mathsf{S}_{m^*}$ will not cause disturbance to $\mathsf{S}_{m^*}|D^\perp \cup (m^*, H(m^*))\rangle$ like $\mathsf{eCO.E}_{f_2}$. We observe that the internal joint state of game $\mathbf{G_4}$ before and after the implementation of operation $\mathsf{S}_{m^*} \circ \mathsf{eCO.E}_{f_2} \circ \mathsf{S}_{m^*}$ can always be written as

$$\sum_{Z,D} \mathsf{S}_{m^*}|Z, D \cup (m^*, H(m^*))\rangle^{7}.$$

Hence, the random oralce H in game $\mathbf{G_4}$, which is simulated using $\mathsf{eCO.RO}$, will always return $H(m^*)$ for the input m^* and $H(m^*)$ is a uniformly random value in $\{0,1\}^n$. Thus, the extractable RO-simulator $\mathcal{S}'(f_2)$ in game $\mathbf{G_4}$ perfectly simulates the random oracle H at the point m^*.

As for the decapsulation oracle qDeca, it is simulated by operation

$$\underline{\mathsf{S}_{m^*} \circ \mathsf{eCO.E}_{f_2} \circ \mathsf{S}_{m^*}} \circ O_G \circ \underline{\mathsf{S}_{m^*} \circ \mathsf{eCO.E}_{f_2} \circ \mathsf{S}_{m^*}}$$

in game $\mathbf{G_4}$. In our reduction, we prove that the extraction result of the operation $\mathsf{S}_{m^*} \circ \mathsf{eCO.E}_{f_2} \circ \mathsf{S}_{m^*}$ acting on state $\mathsf{S}_{m^*}|c, D \cup (m^*, H(m^*)), m\rangle$ is the same as the extraction result of the operation $\mathsf{eCO.E}_{f_2}$ acting on state $|c, D, m\rangle$. Therefore, if c^* is computed as $\mathsf{Enc}_{pk}(m^*, O(m^*))$ in game $\mathbf{G_4}$, we can equivalently use the operation $\mathsf{eCO.E}_{f_2} \circ O_G \circ \mathsf{eCO.E}_{f_2}$ to simulate qDeca. That is to say, game $\mathbf{G_4}$ and following game $\mathbf{G_5}$ are identical.

– Game $\mathbf{G_5}$: This game is like game $\mathbf{G_4}$, except for the following modifications: A new random oracle O is introduced and the challenge ciphertext c^* is generated as $\mathsf{Enc}_{pk}(m^*, O(m^*))$. The decapsulation oracle qDeca in this game is simulated by the operation $\mathsf{eCO.E}_{f_2} \circ O_G \circ \mathsf{eCO.E}_{f_2}$. When adversary $\mathcal{A}$ queries H by $|x, y\rangle$, a conditional operation U as follows is applied.

$$\mathrm{U}|x, y, D\rangle = \begin{cases} \mathsf{eCO.RO}|x, y, D\rangle & (x \neq m^*) \\ |x, y \oplus O(m^*), D\rangle & (x = m^*). \end{cases}$$

[7] Here we abbreviate other registers that may entangled with the database register (e.g. registers of the adversary) as Z.

Game $\mathbf{G_5}$-$\mathbf{G_6}$: However, another problem arises in game $\mathbf{G_5}$, the conditional operation U still needs m^* to perform a test checking if $x = m^*$. In game $\mathbf{G_6}$, the conditional operation U is replaced by a new conditional operation U′ as

$$\mathrm{U}'|x, y, D\rangle = \begin{cases} \mathsf{eCO.RO}|x, y, D\rangle & (\mathsf{Enc}_{pk}(x, O(x)) \neq c^*) \\ |x, y \oplus O(m^*), D\rangle & (\mathsf{Enc}_{pk}(x, O(x)) = c^*). \end{cases}$$

Obviously, if x' satisfying $\mathsf{Enc}_{pk}(x', O(x')) = \mathsf{Enc}_{pk}(m^*, O(m^*))$ does not exist, games $\mathbf{G_5}$ and $\mathbf{G_6}$ are identical. Indeed, if the underlying PKE scheme is δ-correct, the probability that such x' exists is at most 2δ by using the Lemma 4 in [21].

As for the relation between the security of $\mathsf{FO}_m^{\perp}$ and $\mathsf{FO}_m^{\not\perp}$, it is easy to prove that the IND-qCCA security of $\mathsf{FO}_m^{\perp}$ implies the IND-qCCA security of $\mathsf{FO}_m^{\not\perp}$[8]. The proof in the opposite direction heavily relies on Theorem 2 and contains the following two steps:

1. By using Theorem 2, we obtain that any IND-qCCA adversary against $\mathsf{FO}_m^{\perp}$ can be transformed to an IND-CPA adversary against $\mathsf{FO}_m^{\perp}$.
2. Then we prove that any IND-CPA adversary against $\mathsf{FO}_m^{\perp}$ can be efficiently transformed to an IND-qCCA adversary against $\mathsf{FO}_m^{\not\perp}$.

Related Work The reduction from the IND-CCA security of $\mathsf{FO}_m^{\perp}$ in the QROM to the IND-CPA security of $\mathsf{FO}_m^{\perp}$ has been argued in [15]. Their IND-CPA security of $\mathsf{FO}_m^{\perp}$ is in the eQROM$_{\mathsf{Enc}}$, in which the random oracle H is simulated by an extractable RO-simulator $\mathcal{S}(\mathsf{Enc}) := \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{\mathsf{Enc}}\}$ and the decapsulation oracle is simulated by using the extraction-interfaces $\mathsf{eCO.E}_{\mathsf{Enc}}$. They then reduced the IND-CPA security of $\mathsf{FO}_m^{\perp}$ in the eQROM$_{\mathsf{Enc}}$ to the OW-CPA security of the underlying PKE by using the semi-classical OWTH in the eQROM$_f$, which brings a quadratic security loss to their reduction.

In contrast, we reduce the IND-qCCA security of $\mathsf{FO}_m^{\perp}$ in the QROM to the IND-CPA security of $\mathsf{FO}_m^{\perp}$ in the QROM (not eQROM$_{\mathsf{Enc}}$), which enables us to use the MRM O2H lemma and avoid the quadratic security loss.

Recently, Ge et al. [12] proved a lifting theorem for a class of games called the oracle-hiding game, and then proved the IND-qCCA security of $\mathsf{FO}_m^{\perp}$ in the QROM by directly applying that lifting theorem. However, their reduction still has a quadratic security loss. Additionally, by combining Theorem 2 of [21] and Theorem 5.1 of [27], the transformation $\mathsf{HU} \circ \mathsf{KC}$ can transform an OW-CPA-secure DPKE scheme into an IND-qCCA-secure KEM scheme in the QROM. The corresponding reduction also avoids the quadratic security loss, and $\mathsf{HU} \circ \mathsf{KC}$ is also an explicit rejection type KEM transformation. However, compared with the $\mathsf{FO}_m^{\perp}$, the encapsulation and decapsulation algorithms of $\mathsf{HU} \circ \mathsf{KC}$ are more complicated, and the underlying PKE scheme of $\mathsf{HU} \circ \mathsf{KC}$ is restricted to DPKE scheme.

[8] Note that any IND-qCCA adversary against $\mathsf{FO}_m^{\not\perp}$ can be efficiently transformed to an IND-qCCA adversary against $\mathsf{FO}_m^{\perp}$.

2 Preliminaries

2.1 Notation

By $[x = y]$ we denote a bit that is 1 if $x = y$ and 0 otherwise. $H : \mathcal{X} \to \mathcal{Y}$ represents a function with domain $\mathcal{X}$ and codomain $\mathcal{Y}$, and Ω_H is the set of all such functions. For a finite set S, we denote the sampling of a uniformly random element x by $x \xleftarrow{\$} \mathrm{S}$. $x \leftarrow \mathcal{D}$ represents that the chosen x is subject to distribution $\mathcal{D}$. Let $y \leftarrow \mathcal{A}(x)$ denote that the algorithm $\mathcal{A}$ outputs y on input x, and let $y \leftarrow \mathbf{G}$ denote that the game $\mathbf{G}$ finally returns y. For a function or algorithm $\mathcal{A}$, $\mathsf{Time}(\mathcal{A})$ (resp. $\mathsf{Space}(\mathcal{A})$) denotes the time complexity (resp. memory space) of (an algorithm computing) $\mathcal{A}$.

2.2 Quantum Random Oracle Model

We refer to [22] for detailed basics of quantum computation and quantum information. In Appendix A, we provide an overview of important quantum notions that are used in this paper.

Here we first briefly introduce the quantum random oracle model (QROM). The random oracle model (ROM) is an ideal model in which a uniformly random function $H : \mathcal{X} \to \mathcal{Y}$ is selected and all parties have access to H. In the quantum setting, the QROM is considered and the adversary has quantum access to the random oracle in this model [4]. In the QROM, we take the random oracle H as a unitary operation O_H such that $O_H : |x, y\rangle \mapsto |x, y \oplus H(x)\rangle$.

Next, we introduce two lemmas that are used throughout this paper.

Lemma 1 (Simulate the QROM [28]). *Let O be a random oracle, and H be a function uniformly chosen from the set of $2q$-wise independent functions. For any adversary $\mathcal{A}$ with any input z and at most q quantum queries, we have*

$$\Pr[1 \leftarrow \mathcal{A}^H(z)] = \Pr[1 \leftarrow \mathcal{A}^O(z)].$$

Lemma 2 (Measure-Rewind-Measure One-Way to Hiding [20], Lemma 3.3). *Let $H, G : \mathcal{X} \to \mathcal{Y}$ be random functions, z be a random value, and $S \subseteq \mathcal{X}$ be a random set such that $H(x) = G(x)$ for every $x \notin S$. The tuple (H, G, S, z) may have arbitrary joint distribution $\mathcal{D}$. Furthermore, let $\mathcal{A}^O$ be a quantum oracle algorithm (not necessarily unitary) that makes at most q queries to oracle O. Let d be the query depth of $\mathcal{A}$'s oracle O queries. Then we can construct an algorithm $\mathcal{B}^{H,G}(z)$ such that $\mathsf{Time}(\mathcal{B}) \approx 2 \cdot \mathsf{Time}(\mathcal{A})$, $\mathsf{Space}(\mathcal{B}) \approx O(\mathsf{Space}(\mathcal{A}) + \mathsf{Time}(\mathcal{A}))$ and*

$$\begin{aligned} &|\Pr[1 \leftarrow \mathcal{A}^H(z) : (H, G, S, z) \leftarrow \mathcal{D}] - \Pr[1 \leftarrow \mathcal{A}^G(z) : (H, G, S, z) \leftarrow \mathcal{D}]| \\ &\qquad \le 4d \cdot \Pr\left[T \cap S \neq \varnothing : T \leftarrow \mathcal{B}^{H,G}(z), (H, G, S, z) \leftarrow \mathcal{D}\right]. \end{aligned}$$

Here $\mathcal{B}^{H,G}(z)$ makes at most $3q$ queries in total to random functions H and G.

Remark 1. Here we omit the detailed construction of algorithm $\mathcal{B}^{H,G}(z)$ since it is slightly complicated. We emphasize that the property that $\mathsf{Time}(\mathcal{B}) \approx 2 \cdot \mathsf{Time}(\mathcal{A})$ and the fact that $\mathcal{B}^{H,G}(z)$ makes at most $3q$ queries in total are both easily obtained from the detailed construction of $\mathcal{B}^{H,G}(z)$ as presented in [20]. The property $\mathsf{Space}(\mathcal{B}) \approx O(\mathsf{Space}(\mathcal{A}) + \mathsf{Time}(\mathcal{A}))$ is proved by Jiang et al. in [19]. According to the analysis in [19], $\mathcal{B}^{H,G}(z)$ requires $\mathcal{A}$'s quantum gate operations to be explicitly described and accessed, resulting in the need for additional quantum memory space (or quantum register) to implement a unitary variant[9] of $\mathcal{A}$ if $\mathcal{A}$ is not unitary.

2.3 Compressed Oracle Technique

The compressed oracle technique was introduced by Zhandry in [29]. Roughly speaking, its core idea is to purify the quantum random oracle and use the purified version to record information about the quantum queries. In this section, we only introduce the database model and a specific version of the compressed oracle called the compressed standard oracle. Additionally, we set the query upper bound for the compressed standard oracle to a constant value of $q > 0$.

Definition of the database: Let $\perp \notin \{0,1\}^m$ and $\perp \notin \{0,1\}^n$. A database D is a q-pair collection of pairs $(x, y) \in \{0,1\}^m \times \{0,1\}^n$ and $(\perp, 0^n)$ as:

$$D = ((x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), (\perp, 0^n), \dots, (\perp, 0^n)),$$

where $(x_j, y_j) \in \{0,1\}^m \times \{0,1\}^n$ $(j = 1, \dots, i)$, $x_1 < x_2 < \cdots < x_i$, and all $(\perp, 0^n)$ pairs are at the end of the collection. Let $\mathbf{D}_q$ be the set of all these databases. For a $x \in \{0,1\}^m$, we will write $D(x) = y$ if y exists such that $(x, y) \in D$, and $D(x) = \perp$ otherwise. Let $n(D)$ be the number of pairs $(x, y) \in D$ that $x \neq \perp$.

For a pair $(x, y) \in \{0,1\}^m \times \{0,1\}^n$ and a database $D \in \mathbf{D}_q$ with $n(D) < q$ and $D(x) = \perp$, write $D \cup (x, y)$ to be the new database obtained by first deleting a $(\perp, 0^n)$ pair, then inserting (x, y) appropriately into D and maintain the ordering of the x values.

A quantum register D_q defined over set $\mathbf{D}_q$ is a complex Hilbert space with orthonormal basis $\{|D\rangle\}_{D \in \mathbf{D}_q}$, where the basis state $|D\rangle$ is labeled by the elements of $\mathbf{D}_q$. As mentioned in Appendix A, this basis is the computational basis. We also refer to D_q as the database register. For a database $D \in \mathbf{D}_q$ that $n(D) < q$ and $D(x) = \perp$, define a superposition state on the database register D_q as

$$|D \cup (x, \hat{r})\rangle := \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{y \cdot r} |D \cup (x, y)\rangle,$$

where $x \in \{0,1\}^m$ and $r \in \{0,1\}^n$.

For a $x \in \{0,1\}^m$, the local decompression procedure $\mathsf{StdDecomp}_x$ acts on the database register D_q as follows:

[9] The unitary variant of a quantum oracle algorithm is explained in Appendix A.

– For $D \in \mathbf{D}_q$, if $D(x) = \perp$ and $n(D) < q$, $\mathsf{StdDecomp}_x|D\rangle = |D \cup (x, \hat{0^n})\rangle$.
– For $D \in \mathbf{D}_q$, if $D(x) = \perp$ and $n(D) < q$, $\mathsf{StdDecomp}_x|D \cup (x, \hat{0^n})\rangle = |D\rangle$ and

$$\mathsf{StdDecomp}_x|D' \cup (x, \hat{r})\rangle = |D' \cup (x, \hat{r})\rangle \ (r \neq 0^n).$$

– For $D \in \mathbf{D}_q$ that $D(x) = \perp$ and $n(D) = q$, $\mathsf{StdDecomp}_x|D\rangle = |D\rangle$.

For any $x \in \{0,1\}^m$, it is obvious that $\mathsf{StdDecomp}_x$ is a unitary operation and

$$\mathsf{StdDecomp}_x \circ \mathsf{StdDecomp}_x = \mathbf{I}.$$

Here $\mathbf{I}$ is the identity operator.

Definition 1 (Compressed Standard Oracle). *Let* X *(resp.* Y*) be the quantum register defined over* $\{0,1\}^m$ *(resp.* $\{0,1\}^n$*). Let* $|D^\perp\rangle$ *be the initial state on database register* D_q*, where* $D^\perp \in \mathbf{D}_q$ *is the database containing* q *pairs* $(\perp, 0^n)$*. A query to the compressed standard oracle with input/output register* X/Y *is implemented by performing the following unitary operation* CStO *on registers* XYD_q*.*

$$\mathsf{CStO} := \sum_{x \in \{0,1\}^m} |x\rangle\langle x|_{\mathsf{X}} \otimes \mathsf{StdDecomp}_x \circ \mathsf{CNOT}^x_{\mathsf{YD}_q} \circ \mathsf{StdDecomp}_x.$$

For state $|y, D\rangle$ *(*$y \in \{0,1\}^n$*,* $D \in \mathbf{D}_q$*),* $\mathsf{CNOT}^x_{\mathsf{YD}_q}|y, D\rangle = |y \oplus D(x), D\rangle$ *if* $D(x) \neq \perp$*,* $\mathsf{CNOT}^x_{\mathsf{YD}_q}|y, D\rangle = |y, D\rangle$ *if* $D(x) = \perp$[10]*.*

Zhandry proved that the compressed standard oracle is perfectly indistinguishable from the quantum random oracle.

Lemma 3 ([29]). *For any adversary making at most* q *queries, the compressed standard oracle defined in Definition 1 and quantum random oracle* $H : \{0,1\}^m \to \{0,1\}^n$ *are perfectly indistinguishable.*

Let X (resp. Y) be the quantum register defined over a finite set $\mathcal{X}$ (resp. $\mathcal{Y}$). For any function f with domain $\mathcal{X} \times \mathbf{D}_q$ and codomain $\mathcal{Y}$, define the unitary operation Read_f acting on registers $\mathsf{XD}_q\mathsf{Y}$ as

$$\mathsf{Read}_f|x, D, y\rangle = |x, D, y + f(x, D)\rangle, \tag{1}$$

where $+ : \mathcal{Y} \times \mathcal{Y} \to \mathcal{Y}$ is a group operation on $\mathcal{Y}$. Note that Read_f does not change the database in the computational basis state, it only computes $f(x, D)$ and returns the result in register Y. We call Read_f a database read operation.

We now recall the compressed semi-classical oracle and the compressed semi-classical one-way to hidding lemma from [12].

Compressed semi-classical oracle: Let S be a subset of $\mathbf{D}_q$. Define a function f_S such that $f_S(D) = 1$ if $D \in S$, and $f_S(D) = 0$ otherwise. The compressed semi-classical oracle $\mathcal{O}_S^{CSC}$ performs the following operation on input state $\sum \alpha_{z,D}|z, D\rangle$:

[10] The property that $\mathsf{CNOT}^x_{\mathsf{YD}_q}$ acts trivially on the state $|y, D\rangle$ satisfies $D(x) = \perp$, as defined in [9], is actually equivalent to the property that "$y \oplus \perp = y$" defined in [29].

1. Initialize a single qubit register L with $|0\rangle_{\mathsf{L}}$, transform state $\sum \alpha_{z,D}|z, D\rangle|0\rangle_{\mathsf{L}}$ into state $\sum \alpha_{z,D}|z, D\rangle|f_S(D)\rangle_{\mathsf{L}}$.
2. Measure L and output the measurement outcome.

Denote by Find the event that $\mathcal{O}_S^{CSC}$ ever returns 1.

Theorem 1 (Compressed Semi-Classical One-Way to Hidding [12], Theorem 3). *Let $H : \{0,1\}^m \to \{0,1\}^n$ be a quantum random oracle that is implemented by the compressed standard oracle with database register D_q. Let S be a subset of $\mathbf{D}_q$ that $D^\perp \notin S$ and z be a random string. The tuple (S, z) may have arbitrary joint distribution $\mathcal{D}$. Let $H\backslash S$ be an oracle that first queries H and then queries $\mathcal{O}_S^{CSC}$.*

Let $\mathcal{A}$ be a quantum oracle algorithm (not necessarily unitary) that makes at most $q_1 \leq q$[11] (resp. q_2) queries to oracle H (resp. oRead_f). Here f is a function with domain $\mathcal{X} \times \mathbf{D}_q$ and codomain $\mathcal{Y}$, and oracle oRead_f is implemented by the database read operation Read_f defined in (1). Define

$$\begin{aligned} P_{\text{left}} &:= \Pr\left[1 \leftarrow \mathcal{A}^{H,\mathsf{oRead}_f}(z) : (S, z) \leftarrow \mathcal{D}\right], \\ P_{\text{right}} &:= \Pr[1 \leftarrow \mathcal{A}^{H\backslash S,\mathsf{oRead}_f}(z) : (S, z) \leftarrow \mathcal{D}], \\ P_{\text{find}} &:= \Pr[\mathsf{Find} \textit{ occurs in } \mathcal{A}^{H\backslash S,\mathsf{oRead}_f}(z) : (S, z) \leftarrow \mathcal{D}]. \end{aligned}$$

Then

$$|P_{\text{left}} - P_{\text{right}}| \leq \sqrt{(q_1 + 1) \cdot P_{\text{find}}}, \quad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq \sqrt{(q_1 + 1) \cdot P_{\text{find}}}.$$

Define $\mathrm{J}_S := \sum_{D \in S} |D\rangle\langle D|$ as a projector on the database register D_q, let CStO be as in Definition 1. Then we have

$$P_{\text{find}} \leq q_1 \cdot \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}} \|[\mathrm{J}_S, \mathsf{CStO}]\|^2.$$

2.4 The Extractable RO-Simulator

In [9], Don et al. generalized the compressed standard oracle and defined the extractable RO-simulator. Roughly speaking, this simulator simulates the quantum random oracle H by using the compressed standard oracle, and has an extraction-interface that can output a x satisfying $f(x, H(x)) = t$ for an input t. In the following, we present the details of the extractable RO-simulator and introduce a lemma that will be used in the next section. We stress that, similar to Section 2.3, the database register used here is also D_q. Therefore, unlike the inefficient version defined in [9], the extractable RO-simulator described here is efficient.

[11] In fact, even if $q_1 > q$, Theorem 1 is still valid. We require $q_1 \leq q$ here because we have set the query upper bound for the compressed standard oracle to a constant value of q.

Let f be an arbitrary but fixed function with domain $\{0,1\}^m \times \{0,1\}^n$ and codomain $\mathcal{Y}$. For a fixed $t \in \mathcal{Y}$, we define relation $R_t^f \subset \{0,1\}^m \times \{0,1\}^n$ and corresponding parameter $\Gamma_{R_t^f}$ as

$$
\begin{aligned}
R_t^f &:= \{(x,y) \in \{0,1\}^m \times \{0,1\}^n | f(x,y) = t\},\\
\Gamma_{R_t^f} &:= \max_{x \in \{0,1\}^m} |\{y \in \{0,1\}^n | f(x,y) = t\}|.
\end{aligned}
$$

For relation R_t^f, we define following projectors on the database register D_q:

$$
\Sigma^x := \sum_{\substack{D\ s.t.\ (x,D(x)) \in R_t^f \\ x'<x, (x',D(x')) \notin R_t^f}} |D\rangle\langle D| \ (x \in \{0,1\}^m), \quad \Sigma^\perp := \mathbf{I} - \sum_{x \in \{0,1\}^m} \Sigma^x.
$$

Then we define a measurement $\mathbb{M}^{R_t^f}$ on database register D_q to be the set of projectors $\{\Sigma^x\}_{x \in \{0,1\}^m \cup \perp}$.

Indeed, the measurement $\mathbb{M}^{R_t^f}$ returns the smallest x such that $(x, D(x)) \in R_t^f$. If such x does not exist, $\mathbb{M}^{R_t^f}$ will return $\perp$. Similar to [9], we also consider the purified measurement $\mathrm{M}_{\mathsf{D}_q\mathsf{P}}^{R_t^f}$ corresponding to $\mathbb{M}^{R_t^f}$, which is a unitary operation that acts on registers $\mathsf{D}_q\mathsf{P}$ as

$$
\mathrm{M}_{\mathsf{D}_q\mathsf{P}}^{R_t^f} |D, p\rangle = \sum_{x \in \{0,1\}^m \cup \perp} \Sigma^x |D\rangle |p \oplus x\rangle.
$$

Here P is a quantum register defined over $\{0,1\}^{m+1}$[12], $D \in \mathbf{D}_q$ and $p \in \{0,1\}^{m+1}$.

Definition 2 (The Extractable RO-Simulator (efficient version)). *The extractable RO-simulator $\mathcal{S}(f)$ with an internal database register D_q is a black-box oracle with two interfaces: the RO-interface* eCO.RO *and the extraction-interface* $\mathsf{eCO.E}_f$. *$\mathcal{S}(f)$ prepares its database register D_q to be in state $|D^\perp\rangle$ at the beginning, where $D^\perp \in \mathbf{D}_q$ is the database containing q pairs $(\perp, 0^n)$. Then, the RO-interface* eCO.RO *and the extraction-interface* $\mathsf{eCO.E}_f$ *act as follows:*

– *Let* X *(resp.* Y*) be the quantum register defined over $\{0,1\}^m$ (resp. $\{0,1\}^n$),* T *be the quantum register defined over $\mathcal{Y}$.*
– eCO.RO*: For any quantum RO-query on query registers* XY*, $\mathcal{S}(f)$ implements a compressed standard oracle query on registers* XYD_q *by the* CStO *defined in Definition 1.*
– $\mathsf{eCO.E}_f$*: For any quantum extraction-query on query registers* TP*, $\mathcal{S}(f)$ applies*

$$
\mathsf{Ext}_f := \sum_{t \in \mathcal{Y}} |t\rangle\langle t|_{\mathsf{T}} \otimes \mathrm{M}_{\mathsf{D}_q\mathsf{P}}^{R_t^f} \tag{2}
$$

to registers $\mathsf{TD}_q\mathsf{P}$.

[12] Here we embed the set $\{0,1\}^m \cup \perp$ into the set $\{0,1\}^{m+1}$ as explained in Appendix A.

Moreover, by the Theorem 4.3 of [9], the total runtime of $\mathcal{S}(f)$ is bounded[13] *by*

$$T_{\mathcal{S}} = O(q_{RO} \cdot q_E \cdot \mathrm{Time}[f] + q_{RO}^2),$$

where $q_{RO}(\leq q)$[14] *and q_E are the number of queries to* eCO.RO *and* eCO.E$_f$*, respectively.*

The eCO.RO (resp. eCO.E$_f$) can also be classically queried. In this case, the query registers XY (resp. TP) are measured after applying the unitary operation CStO (resp. Ext$_f$). The eCO.RO can also be queried in parallel, and k-parallel queries to eCO.RO are processed by sequentially implementing CStO k times [6].

In addition, for any computational basis state $|t, D, p\rangle$ on register $\mathsf{TD}_q\mathsf{P}$, it is straightforward to check that

$$\mathsf{Ext}_f|t, D, p\rangle = |t, D, p \oplus g(t, D)\rangle. \tag{3}$$

Here function $g : \mathcal{Y} \times \mathbf{D}_q \to \{0,1\}^{m+1}$ on input (t, D) outputs the smallest value x that satisfies $(x, D(x)) \in R_t^f$. If such x does not exist, function g outputs $\perp$. Therefore, by the definition of the database read operation given in Section 2.3, Ext$_f$ can also be considered as a database read operation.

Lemma 4 ([12] Lemma 2). *For any $x \in \{0,1\}^m$, let* StdDecomp$_x$ *and* CStO *be the unitary operation defined in Section 2.3, then*

$$\|[\mathsf{Ext}_f, \mathsf{StdDecomp}_x]\| \leq 16 \cdot \sqrt{\max_{t \in \mathcal{Y}} \Gamma_{R_t^f}/2^n}, \quad \|[\mathsf{CStO}, \Sigma^{\perp}]\| \leq 8 \cdot \sqrt{\Gamma_{R_t^f}/2^n}.$$

Here $[A, B] := AB - BA$ is the commutator of two operations A, B acting on a quantum register.

3 From IND-CPA$_{\mathsf{FO}_m^{\perp}[\mathsf{P}]}$ to IND-qCCA$_{\mathsf{FO}_m^{\perp}[\mathsf{P}]}$

In this section, we prove that, in the QROM, the IND-qCCA security of KEM scheme $\mathsf{FO}_m^{\perp}[\mathsf{P}, H, G]$ can be tightly reduced to its IND-CPA security. Particularly, our reduction does not require the perfect correctness property of the underlying randomized PKE scheme P. The formal definitions of cryptographic primitives, correctness and spreadness used in this section are shown in Appendix B.

Transformation $\mathsf{FO}_m^{\perp}$: Let $\mathsf{P} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a randomized PKE with message space $\mathcal{M}(= \{0,1\}^m)$, randomness space $\{0,1\}^n$ and ciphertext space $\mathcal{C}$. Let $H : \mathcal{M} \to \{0,1\}^n$ and $G : \{0,1\}^* \to \{0,1\}^{n'}$ be hash functions. We associate

$$\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{P}, H, G] = (\mathsf{Gen}, \mathsf{Enca}_m, \mathsf{Deca}_m^{\perp}).$$

The constituting algorithms of $\mathsf{KEM}_m^{\perp}$ are given in Fig. 3.

[13] Although [9] defined an inefficient version of the extractable RO-simulator, the total runtime of the efficient version is given instead in the Theorem 4.3 of [9].

[14] This is because we have set the query upper bound for the compressed standard oracle to a constant value of q.

$$
\begin{array}{lll}
\underline{\mathsf{Gen}} & \underline{\mathsf{Encap}_m\,(pk)} & \underline{\mathsf{Deca}_m^{\perp}\,(sk,c)} \\
(pk,sk) \leftarrow \mathsf{Gen} & m \xleftarrow{\$} \mathcal{M} & m' = \mathsf{Dec}_{sk}\,(c) \\
\textbf{return } (pk,sk) & c = \mathsf{Enc}_{pk}\,(m, H(m)) & \textbf{if } m' = \perp \\
 & K = G(m) & \quad \textbf{return } \perp \\
 & \textbf{return } (K,c) & \textbf{else if } c \neq \mathsf{Enc}_{pk}\,(m'; H(m')) \\
 & & \quad \textbf{return } \perp \\
 & & \textbf{return } K = G(m')
\end{array}
$$

Fig. 3. Key Encapsulation Mechanism $\mathsf{KEM}_m^{\perp} = (\mathsf{Gen}, \mathsf{Enca}_m, \mathsf{Deca}_m^{\perp})$.

Before we prove the main result of this section, we first describe how to simulate a quantum accessible decapsulation oracle qDeca for $\mathsf{KEM}_m^{\perp}$.

Denote by I/O the input/output register of qDeca, where I is defined over $\mathcal{C}$ and O is defined over $\{0,1\}^{n'+1}$[15]. As shown in Fig. 3, decapsulation algorithm $\mathsf{Deca}_m^{\perp}$ needs to query H and G in its process. Specifically, it queries H to perform the re-encryption check (i.e., check if $c = \mathsf{Enc}_{pk}(m', H(m'))$), and then queries G by m' to produce the key K if m' passes the re-encryption check. Following this process, a unitary operation U_m acting on registers IM is presented as follows:

$$
\mathrm{U}_m |c\rangle_{\mathsf{I}} |0^m\rangle_{\mathsf{M}} = \begin{cases} |c\rangle_{\mathsf{I}} |m'\rangle_{\mathsf{M}} \text{ if } m' := \mathsf{Dec}_{sk}(c) \neq \perp \wedge \mathsf{Enc}_{pk}(m', H(m')) = c \\ |c\rangle_{\mathsf{I}} |\perp\rangle_{\mathsf{M}} \;\; \text{otherwise.} \end{cases}
$$

Here M is a quantum register defined over $\{0,1\}^{m+1}$[15]. With this operation, the re-encryption check can be performed in superposition. The quantum circuit implementation of U_m is shown in Appendix C, which two queries to H is needed.

To simulate qDeca on input state $|c\rangle_{\mathsf{I}}|y\rangle_{\mathsf{O}}$, the following unitary operation is performed on state $|c\rangle_{\mathsf{I}}|y\rangle_{\mathsf{O}}|0^m\rangle_{\mathsf{M}}$:

$$
\mathrm{U}_{\mathsf{qD}} := (\mathrm{U}_m)^{\dagger} \circ O_G \circ \mathrm{U}_m, \tag{4}
$$

where unitary operation O_G maps $|m'\rangle_{\mathsf{M}}|y\rangle_{\mathsf{O}}$ to $|m'\rangle_{\mathsf{M}}|y \oplus G(m')\rangle_{\mathsf{O}}$, and we set $G(\perp) = \perp$. The register M used by U_m can be viewed as the internal register of U_{qD}, it stores the plaintext m'. Note that this register is always in state $|0^m\rangle_{\mathsf{M}}$ before and after once simulation of qDeca.

Theorem 2 (IND-$\mathsf{CPA}_{\mathsf{KEM}_m^{\perp}} \overset{\mathrm{QROM}}{\Rightarrow}$ IND-$\mathsf{qCCA}_{\mathsf{KEM}_m^{\perp}}$). *Let P be a randomized PKE scheme that is δ-correct and weakly γ-spread. Let $\mathcal{A}$ be an IND-qCCA adversary against $\mathsf{KEM}_m^{\perp}$ in the QROM, making at most q_H, q_G and q_D queries to random oracle H, random oracle G and decapsulation oracle qDeca^*[16], respectively. Let d_H (resp. d_G) be the query depth of $\mathcal{A}$'s random oracle H (resp. G)*

[15] Here we embed the set $\{0,1\}^{n'} \cup \perp$ (resp. $\{0,1\}^m \cup \perp$) into the set $\{0,1\}^{n'+1}$ (resp. $\{0,1\}^{m+1}$) as explained in Appendix A.

[16] Here and in what follows, we following [16] to make the convention that q_H and q_G counts the total number of times H and G is queried in the security game, respectively.

queries. Let w_H (resp. w_G) be the query width of $\mathcal{A}$'s random oracle H (resp. G) queries.

Then there exists an $\mathsf{IND\text{-}CPA}$ *adversary $\tilde{\mathcal{A}}$ against* $\mathsf{KEM}_m^{\perp}$ *in the QROM such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}_m^{\perp},\mathcal{A}} \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}_m^{\perp},\tilde{\mathcal{A}}} + 8\sqrt{q_H(q_H+1)\cdot\delta} + (64q_H+2)\cdot\delta + 40q_D\cdot 2^{-\gamma/2}.$$

The adversary $\tilde{\mathcal{A}}$ makes at most $2q_H$ (resp. q_G+q_D) queries to random oracle H (resp. G). The query depth of $\tilde{\mathcal{A}}$ to random oracle H (resp. G) is $2d_H$ (resp. d_G+q_D). The running time and memory space of $\tilde{\mathcal{A}}$ is bounded as $\mathsf{Time}(\tilde{\mathcal{A}}) \approx \mathsf{Time}(\mathcal{A}) + O(q_Hq_D + q_H^2)$ *and* $\mathsf{Space}(\tilde{\mathcal{A}}) \approx \mathsf{Space}(\mathcal{A}) + O(q_H)$*, respectively.*

Proof. To prove this theorem, a series of hybrid games are defined (see also Fig. 4).

GAMES $\mathbf{G_0}$-$\mathbf{G_6}$

1, $(pk, sk) \leftarrow \mathsf{Gen}$ //$\mathbf{G_0}$-$\mathbf{G_6}$

2, $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G, O \xleftarrow{\$} \Omega_H$ //$\mathbf{G_0}$-$\mathbf{G_6}$

3, $b \xleftarrow{\$} \{0,1\}$, $m^* \xleftarrow{\$} \mathcal{M}$ //$\mathbf{G_0}$-$\mathbf{G_6}$

4, $c^* = \mathsf{Enc}_{pk}(m^*, H(m^*))$ //$\mathbf{G_0}$-$\mathbf{G_6}$

5, $K_0^* = G(m^*)$, $K_1^* \xleftarrow{\$} \mathcal{K}$ //$\mathbf{G_0}$-$\mathbf{G_6}$

6, $b' \leftarrow \mathcal{A}^{H,G,\mathsf{qDeca}^*}(pk, c^*, K_b^*)$ //$\mathbf{G_0}$-$\mathbf{G_1}$

$b' \leftarrow \mathcal{A}^{H,G,\mathsf{qDeca}^\diamond}(pk, c^*, K_b^*)$ //$\mathbf{G_2}$-$\mathbf{G_6}$

7, **return** $[b = b']$ //$\mathbf{G_0}$-$\mathbf{G_6}$

$\underline{H(|x_H, y_H\rangle)}$ //$\mathbf{G_0}$-$\mathbf{G_6}$

8, **return** $|x_H, y_H \oplus H(x_H)\rangle$ //$\mathbf{G_0}$

9, **query** $\mathsf{eCO.RO}$ by $|x_H, y_H\rangle$ //$\mathbf{G_1}$-$\mathbf{G_4}$

10, **if** $x_H = m^*$ //$\mathbf{G_5}$

return $|x_H, y_H \oplus O(x_H)\rangle$

else query $\mathsf{eCO.RO}$ by $|x_H, y_H\rangle$

11, **if** $\mathsf{Enc}_{pk}(x_H, O(x_H)) = c^*$ //$\mathbf{G_6}$

return $|x_H, y_H \oplus O(x_H)\rangle$

else query $\mathsf{eCO.RO}$ by $|x_H, y_H\rangle$

$\underline{G(|x_G, y_G\rangle)}$ //$\mathbf{G_0}$-$\mathbf{G_6}$

12, **return** $O_G|x_G, y_G\rangle = |x_G, y_G \oplus G(x_G)\rangle$

$\underline{\mathsf{qDeca}^*(|c, y\rangle)}$ //$\mathbf{G_0}$-$\mathbf{G_1}$

13, **if** $c = c^*$ **return** $|c, y \oplus \perp\rangle$

else return

$(\mathrm{U}_m)^\dagger \circ O_G \circ \mathrm{U}_m|c, y\rangle$ //$\mathbf{G_0}$

$(\tilde{\mathrm{U}}_m)^\dagger \circ O_G \circ \tilde{\mathrm{U}}_m|c, y\rangle$ //$\mathbf{G_1}$

$\underline{\mathsf{qDeca}^\diamond(|c, y\rangle)}$ //$\mathbf{G_2}$-$\mathbf{G_6}$

14, **if** $c = c^*$ **return** $|c, y \oplus \perp\rangle$

else return

$\mathsf{eCO.E}_f \circ O_G \circ \mathsf{eCO.E}_f|c, y\rangle$

$\underline{\mathcal{S}(f) = \{\mathsf{eCO.RO}, \mathsf{eCO.E}_f\}}$ //$\mathbf{G_1}$-$\mathbf{G_6}$

15, $\mathsf{eCO.RO}$: **apply** CStO //$\mathbf{G_1}$-$\mathbf{G_6}$

16, $\mathsf{eCO.E}_f$: $f = f_1$, **apply** Ext_{f_1} //$\mathbf{G_1}$-$\mathbf{G_2}$

$\mathsf{eCO.E}_f$: $f = f_2$, **apply** Ext_{f_2} //$\mathbf{G_3}$

$\mathsf{eCO.E}_f$: $f = f_2$, //$\mathbf{G_4}$

apply $\mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}$

$\mathsf{eCO.E}_f$: $f = f_2$, **apply** Ext_{f_2} //$\mathbf{G_5}$-$\mathbf{G_6}$

Fig. 4. Games $\mathbf{G_0}$ to $\mathbf{G_6}$ in the proof of Theorem 2. In these games, the adversary $\mathcal{A}$ can make parallel quantum queries to H and G and quantum queries to qDeca^*. In this figure, for brevity, we just write the input state of H, G and qDeca^* as $|x_H, y_H\rangle$, $|x_G, y_G\rangle$ and $|c, y\rangle$, respectively. We also stress that the $H(m^*)$ used to compute c^* ($= \mathsf{Enc}_{pk}(m^*, H(m^*))$) in game $\mathbf{G_1}$ to $\mathbf{G_4}$ is generated by classically query $\mathsf{eCO.RO}$ with input m^*.

Game $\mathbf{G_0}$: This is the $\mathsf{IND\text{-}qCCA}$ game of $\mathsf{KEM}_m^{\perp}$ with adversary $\mathcal{A}$ in the QROM. The decapsulation oracle qDeca^* in this game is identical to qDeca that is simulated by U_{qD} as defined in (4), except that qDeca^* returns $\perp$ if $c = c^*$.

$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}^{\perp}_{m},\mathcal{A}} = \left|\Pr[1 \leftarrow \mathbf{G_0}] - \frac{1}{2}\right|. \tag{5}$$

We recall that the input/output register of the decapsulation oracle is denoted as $\mathsf{I/O}$, and U_{qD} also has an internal register M. Here we denote the private register of adversary $\mathcal{A}$ as A, which contains the query registers of the random oracle H and G.

Define $\mathrm{P}_{c^*} := |c^*\rangle\langle c^*|$ as a projector on the input register I, $\mathrm{U}_\perp$ as a unitary operation that acts on the output register O and maps $|y\rangle$ to $|y \oplus \perp\rangle$. Then the decapsulation oracle qDeca^* in game $\mathbf{G_0}$ is simulated by the unitary operation

$$\mathrm{U}^0_{\mathsf{qD}} := \mathrm{U}_\perp \circ \mathrm{P}_{c^*} + (\mathrm{U}_m)^\dagger \circ O_G \circ \mathrm{U}_m \circ (\mathbf{I} - \mathrm{P}_{c^*}).$$

Let D_{q_H} be the database register defined over set $\mathbf{D}_{q_H}$ (Section 2.3). Let $\mathcal{S}(f_1) := \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_1}\}$ be the extractable RO-simulator with internal database register D_{q_H} (Definition 2), where function $f_1 : \mathcal{M} \times \{0,1\}^n \cup \perp \to \mathcal{C} \cup \perp$ is that

$$f_1(x,y) = \begin{cases} c \text{ if } y \neq \perp \wedge \mathsf{Enc}_{pk}(x,y) = c \wedge x = \mathsf{Dec}_{sk}(c) \\ \perp \text{ otherwise.} \end{cases}$$

Game $\mathbf{G_1}$: This game is identical to game $\mathbf{G_0}$, except that the extractable RO-simulator $\mathcal{S}(f_1) := \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_1}\}$ is introduced and the queries to random oracle H are answered by querying the RO-interface $\mathsf{eCO.RO}$.

In game $\mathbf{G_1}$, the decapsulation oracle qDeca^* is simulated by the unitary operation

$$\mathrm{U}^1_{\mathsf{qD}} := \mathrm{U}_\perp \circ \mathrm{P}_{c^*} + (\tilde{\mathrm{U}}_m)^\dagger \circ O_G \circ \tilde{\mathrm{U}}_m \circ (\mathbf{I} - \mathrm{P}_{c^*}).$$

Here $\tilde{\mathrm{U}}_m$ acts the same as U_m, except that the internal two random oracle H queries are answered by querying $\mathsf{eCO.RO}$.

In game $\mathbf{G_1}$, although the extractable RO-simulator $\mathcal{S}(f_1)$ is used to answer the queries to random oracle H, the extraction-interface $\mathsf{eCO.E}_{f_1}$ is never queried. By using Lemma 3, we have

$$\Pr[1 \leftarrow \mathbf{G_0}] = \Pr[1 \leftarrow \mathbf{G_1}]. \tag{6}$$

Game $\mathbf{G_2}$: This game is identical to game $\mathbf{G_1}$, except that the decapsulation oracle qDeca^* is replaced with $\mathsf{qDeca}^\diamond$.

Instead of using $\tilde{\mathrm{U}}_m$ to perform the re-encryption check in superposition, the decapsulation oracle $\mathsf{qDeca}^\diamond$ in game $\mathbf{G_2}$ queries $\mathsf{eCO.E}_{f_1}$ to directly extract plaintext m' that passes the re-encryption check from the database register. Moreover, the decapsulation oracle $\mathsf{qDeca}^\diamond$ in game $\mathbf{G_2}$ is simulated by the unitary operation

$$\mathrm{U}^2_{\mathsf{qD}} := \mathrm{U}_\perp \circ \mathrm{P}_{c^*} + \mathsf{Ext}_{f_1} \circ O_G \circ \mathsf{Ext}_{f_1} \circ (\mathbf{I} - \mathrm{P}_{c^*})$$

that acts on the registers $\mathsf{IOD}_{q_H}\mathsf{M}$, where $\mathsf{Ext}_{f_1} := \sum_{c\in\mathcal{C}} |c\rangle\langle c|_{\mathsf{I}} \otimes \mathrm{M}^{R_c^{f_1}}_{\mathsf{D}_{q_H}\mathsf{M}}$ acts on registers $\mathsf{ID}_{q_H}\mathsf{M}$[17]. Similar to (3) in Section 2.4, the unitary operation Ext_{f_1} can also be rewritten as

$$\mathsf{Ext}_{f_1}|c, D, m\rangle_{\mathsf{ID}_{q_H}\mathsf{M}} = |c, D, m \oplus x\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}.$$

Here x is the smallest value that satisfies $f_1(x, D(x)) = c$. If such x does not exist, Ext_{f_1} returns $\perp$ in register M.

Indeed, we can prove the following lemma and the detailed proof is shown in Appendix D.1.

Lemma 5. $|\Pr[1 \leftarrow \mathbf{G_1}] - \Pr[1 \leftarrow \mathbf{G_2}]| \leq 8q_D \cdot 2^{-\gamma/2}$.

Game $\mathbf{G_3}$: This game is the same as game $\mathbf{G_2}$, except that the extractable RO-simulator is replaced to $\mathcal{S}(f_2) := \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_2}\}$, where function $f_2 : \mathcal{M} \times \{0,1\}^n \to \mathcal{C} \cup \perp$ is that $f_2(x, y) = \mathsf{Enc}_{pk}(x, y)$.

In game $\mathbf{G_3}$, the decapsulation oracle $\mathsf{qDeca}^{\diamond}$ is simulated by the unitary operation

$$\mathrm{U}^3_{\mathsf{qD}} := \mathrm{U}_\perp \circ \mathrm{P}_{c^*} + \mathsf{Ext}_{f_2} \circ O_G \circ \mathsf{Ext}_{f_2} \circ (\mathbf{I} - \mathrm{P}_{c^*}) \tag{7}$$

that acts on registers $\mathsf{IOD}_{q_H}\mathsf{M}$, where $\mathsf{Ext}_{f_2} := \sum_{c\in\mathcal{C}} |c\rangle\langle c|_{\mathsf{I}} \otimes \mathrm{M}^{R_c^{f_2}}_{\mathsf{D}_{q_H}\mathsf{M}}$ acts on registers $\mathsf{ID}_{q_H}\mathsf{M}$. Similar with Ext_{f_1}, the unitary operation Ext_{f_2} can be rewritten as

$$\mathsf{Ext}_{f_2}|c, D, m\rangle_{\mathsf{ID}_{q_H}\mathsf{M}} = |c, D, m \oplus x\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}.$$

Here x is the smallest value satisfies $f_2(x, D(x)) = c$. If such x does not exist, Ext_{f_2} returns $\perp$ in register M. We note that the implementation of Ext_{f_2} does not require sk since the computation of function f_2 only uses pk. Therefore, the implementation of $\mathrm{U}^3_{\mathsf{qD}}$ also does not require sk.

Compared with f_1, function f_2 directly computes $\mathsf{Enc}_{pk}(x, y)$ and ignores the check of whether x equals $\mathsf{Dec}_{sk}(c)$, where $c = \mathsf{Enc}_{pk}(x, y)$. Hence, for any computational basis state $|c, D, m\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$, if Ext_{f_1} does not map it to $|c, D, m \oplus \perp\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$, then Ext_{f_2} will also be unable to map it to $|c, D, m \oplus \perp\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$. Indeed, Ext_{f_2} may have a different return than Ext_{f_1} only on the following type of input state:

(a) $|c, D, m\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$: $c \neq c^*$, Ext_{f_1} maps it to $|c, D, m \oplus \perp\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$, but Ext_{f_2} does not.

(b) $|c, D, m\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$: $c \neq c^*$, neither Ext_{f_1} nor Ext_{f_2} maps it to $|c, D, m \oplus \perp\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$, but the return state of Ext_{f_1} and Ext_{f_2} is different.

[17] Note that the codomain of function f_1 is the union of $\mathcal{C}$ and $\perp$. However, we ignore the extraction with input $\perp$ in Ext_{f_1}, which is different from its definition as shown in Definition 2. That is to say, we restrict the adversary $\mathcal{A}$ from querying the decapsulation oracle by $\perp$ in our reduction. Indeed, this is reasonable since $\perp \notin \mathcal{C}$.

For a fixed (pk, sk) pair, define a set of database as

$$R^D_{pk,sk} := \{D|D \in \mathbf{D}_{q_H}, \exists x \ s.t. \ D(x) \neq \bot \wedge \mathsf{Enc}_{pk}(x, D(x)) = c \wedge \mathsf{Dec}_{sk}(c) \neq x\}. \tag{8}$$

It is straightforward to check that the database D in state $|c, D, m\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$ of types (a) and (b) above must satisfy $D \in R^D_{pk,sk}$. Hence, we can conclude that the extraction-interfaces $\mathsf{eCO.E}_{f_1}$ and $\mathsf{eCO,E}_{f_2}$ proceed identically for any input state $|c, D, m\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$ if $D \notin R^D_{pk,sk}$.

By using Theorem 1, we can prove the following lemma. The detailed proof is shown in Appendix D.3.

Lemma 6. $|\Pr[1 \leftarrow \mathbf{G_2}] - \Pr[1 \leftarrow \mathbf{G_3}]| \leq 8 \cdot \sqrt{q_H(q_H+1)\cdot\delta} + 64 q_H \cdot \delta.$

Game $\mathbf{G_4}$: This game is the same as game $\mathbf{G_3}$, except that the extraction interface $\mathsf{eCO.E}_{f_2}$ is implemented by unitary operation $\mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}$. Here, S_{m^*} is the abbreviation of $\mathsf{StdDecomp}_x$ defined in Section 2.3.

Obviously, the decapsulation oracle $\mathsf{qDeca}^{\diamond}$ in game $\mathbf{G_4}$ is simulated by unitary operation

$$\mathrm{U}^4_{\mathsf{qD}} := \mathrm{U}_\bot \circ \mathrm{P}_{c^*} + \underline{\mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}} \circ O_G \circ \underline{\mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}} \circ (\mathbf{I} - \mathrm{P}_{c^*}).$$

For a fixed (pk, sk) pair, one can check that the parameter $\Gamma_{R_c^{f_2}}$ related to function f_2 defined in Section 2.4 satisfies

$$\max_{c\in\mathcal{C}} \Gamma_{R_c^{f_2}}/2^n \leq \gamma(pk, sk),$$

since the underlying PKE scheme P is weakly γ-spread. Then, by Lemma 4,

$$\|[\mathsf{Ext}_{f_2}, \mathsf{S}_{m^*}]\| \leq 16 \cdot \sqrt{\max_{c\in\mathcal{C}} \Gamma_{R_c^{f_2}}/2^n} \leq 16 \cdot \sqrt{\gamma(pk, sk)}.$$

Notice that $\mathsf{S}_{m^*} \circ \mathsf{S}_{m^*} = \mathbf{I}$, thus we can conclude that $\mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}$ is indistinguishable from Ext_{f_2} except for an error of $16 \cdot \sqrt{\gamma(pk, sk)}$.

In game $\mathbf{G_4}$, the query times to decapsulation oracle $\mathsf{qDeca}^{\diamond}$ are at most q_D, thus the unitary operation $\mathrm{U}^4_{\mathsf{qD}}$ is implemented at most q_D times. Then, for a fixed (pk, sk) pair, it is easy to obtain

$$|\Pr[1 \leftarrow \mathbf{G_3} : (pk, sk)] - \Pr[1 \leftarrow \mathbf{G_4} : (pk, sk)]| \leq 32 q_D \cdot \sqrt{\gamma(pk, sk)}.$$

Here $\Pr[1 \leftarrow \mathbf{G} : (pk, sk)]$ is the probability that game $\mathbf{G}$ returns 1 for fixed (pk, sk). By averaging the (pk, sk), we obtain

$$|\Pr[1 \leftarrow \mathbf{G_3}] - \Pr[1 \leftarrow \mathbf{G_4}]| \leq 32 q_D \cdot \sqrt{\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}} \gamma(pk, sk)} \overset{(a)}{\leq} 32 q_D \cdot 2^{-\gamma/2}. \tag{9}$$

Here (a) uses the fact that the underlying PKE scheme P is weakly γ-spread.

In game $\mathbf{G_4}$, c^* is computed by $H(m^*)$, which is generated by classically querying the RO-interface $\mathsf{eCO.RO}$ with m^*. As defined in Definition 2, $\mathsf{eCO.RO}$

is implemented by the unitary operation CStO. Indeed, by the definition of the CStO (Definition 1), the joint state of game $\mathbf{G_4}$ just before $\mathcal{A}$ performs its first query to $\mathsf{qDeca}^{\diamond}$ can be written as

$$\sum_{z,c,y,D} \alpha_{z,c,y,D} \mathsf{S}_{m^*} |z, c, y, D \cup (m^*, H(m^*))\rangle_{\mathsf{AIOD}_{q_H}} |0^m\rangle_{\mathsf{M}}.$$

Then, for any basis state

$$|\psi\rangle := \mathsf{S}_{m^*}|z, c, y, D \cup (m^*, H(m^*)), 0^m\rangle,$$

suppose unitary operation Ext_{f_2} maps state $|z, c, y, D \cup (m^*, H(m^*)), 0^m\rangle$ to state $|z, c, y, D \cup (m^*, H(m^*)), m\rangle$, we have

$$\begin{aligned} \mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}|\psi\rangle &= \mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2}|z, c, y, D \cup (m^*, H(m^*)), 0^m\rangle \\ &= \mathsf{S}_{m^*}|z, c, y, D \cup (m^*, H(m^*)), m\rangle. \end{aligned} \tag{10}$$

Therefore, if we abbreviate the other registers as R, the internal joint state of game $\mathbf{G_4}$ before and after the implementation of $\mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}$ always can be written as

$$\sum_{r,D} \beta_{r,D} |r, D \cup \mathsf{S}_{m^*}(m^*, H(m^*))\rangle_{\mathsf{RD}_{q_H}}.$$

Now, by the definition of the CStO (Definition 1), we can conclude that the random oracle H query (which is simulated by $\mathsf{eCO.RO}$) with m^* makes by $\mathcal{A}$ in game $\mathbf{G_4}$ will return $H(m^*)$ again, thus the extractable RO-simulator $\mathcal{S}(f_2) = \{\mathsf{eCO.RO}, \mathsf{S}_{m^*} \circ \mathsf{eCO.E}_{f_2} \circ \mathsf{S}_{m^*}\}$ of game $\mathbf{G_4}$ perfectly simulates the random oracle H at point m^*.

In addition, we can prove the following lemma.

Lemma 7. *For any basis state $|z, c, y, D\rangle$, suppose $c \neq c^*$,*

$$\mathsf{Ext}_{f_2}|z, c, y, D \cup (m^*, H(m^*)), 0^m\rangle = |z, c, y, D \cup (m^*, H(m^*)), m\rangle$$

and $\mathsf{Ext}_{f_2}|z, c, y, D, 0^m\rangle = |z, c, y, D, m'\rangle$, then we have $m = m'$.

Proof. We recall that $c^* = \mathsf{Enc}_{pk}(m^*, H(m^*))$. Denote database $D \cup (m^*, H(m^*))$ as D', then we have $D'(m^*) = H(m^*)$. By the definition of function f_2, if the value $m \neq \perp$, it satisfies that $D'(m) \neq \perp$ and $\mathsf{Enc}_{pk}(m, D'(m)) = c$. Then we can conclude that m cannot be m^*, because $m = m^*$ implies $\mathsf{Enc}_{pk}(m^*, D'(m^*)) = c$, which is contradictory to $c \neq c^*$.

So even if database $D \cup (m^*, H(m^*))$ contains more information than D, the return of Ext_{f_2} on input state $|z, c, y, D \cup (m^*, H(m^*)), 0^m\rangle$ is irrelevant to that additional information. Thus, Ext_{f_2} returns the same value on state $|z, c, y, D \cup (m^*, H(m^*)), 0^m\rangle$ and $|z, c, y, D, 0^m\rangle$, i.e., $m = m'$. ☐

By using above lemma and (10), we obtain that the return of operation $\mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}$ acting on state

$$\mathsf{S}_{m^*}|z, c, y, D \cup (m^*, H(m^*)), 0^m\rangle \ (c \neq c^*)$$

is identical to the return of operation Ext_{f_2} acting on state $|z, c, y, D, 0^m\rangle$. This implies that even if we do not query $\mathsf{eCO.RO}$ by m^* to generate c^* in game $\mathbf{G_4}$, and generate it as $\mathsf{Enc}_{pk}(m^*, O(m^*))$ instead ($O \xleftarrow{\$} \Omega_H$), the operation $\mathsf{S}_{m^*} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*}$ in game $\mathbf{G_4}$ can then be reduced to operation Ext_{f_2} directly. In other words, we can transform game $\mathbf{G_4}$ to the following game $\mathbf{G_5}$ equivalently.

Game $\mathbf{G_5}$: Compared with game $\mathbf{G_4}$, this game has two modifications:

- The simulation of random oracle H is changed. Let $O \xleftarrow{\$} \Omega_H$ be a new random oracle, when H is queried with state $|x, y\rangle_{\mathsf{XY}}$, a conditional operation to registers XY is applied:
  - Query $\mathsf{eCO.RO}$ if $x \neq m^*$, query random oracle O with input/output register X/Y if $x_i = m^*$.

  The simulation of parallel queries can be done in a similar manner. We note that the c^* in this game is computed as $\mathsf{Enc}_{pk}(m^*, O(m^*))$.
- The extraction-interface $\mathsf{eCO.E}_{f_2}$ is implemented by the unitary operation Ext_{f_2}. Therefore, the decapsulation oracle $\mathsf{qDeca}^{\diamond}$ in this game is simulated by the unitary operation $\mathrm{U}^3_{\mathsf{qD}}$ defined in (7).

$$\Pr[1 \leftarrow \mathbf{G_4}] = \Pr[1 \leftarrow \mathbf{G_5}]. \tag{11}$$

Notice that game $\mathbf{G_5}$ needs m^* to implement a conditional operation when simulating H. In the following game $\mathbf{G_6}$, a new conditional operation without using m^* is implemented instead.

Game $\mathbf{G_6}$: This game is the same as $\mathbf{G_5}$, except that a new conditional operation as follows is implemented to simulate random oracle H.

- Query $\mathsf{eCO.RO}$ if $\mathsf{Enc}_{pk}(x, O(x)) \neq c^*$, query random oracle O with input/output register X/Y if $\mathsf{Enc}_{pk}(x, O(x)) = c^*$.

Define a subset of message space $\mathcal{M}$ as

$$\mathrm{S}^{collision}_{pk,sk,O} := \{m | \exists m' \neq m, \mathsf{Enc}_{pk}(m, O(m)) = \mathsf{Enc}_{pk}(m', O(m'))\}.$$

It is obvious that games $\mathbf{G_5}$ and $\mathbf{G_6}$ are identical if $m^* \notin \mathrm{S}^{collision}_{pk,sk,O}$ for $(pk, sk) \leftarrow \mathsf{Gen}$ and $O \xleftarrow{\$} \Omega_H$. By using Lemma 9 and the δ-correct property of the underlying PKE scheme P, we obtain

$$|\Pr[1 \leftarrow \mathbf{G_5}] - \Pr[1 \leftarrow \mathbf{G_6}]| \leq 2\delta. \tag{12}$$

Now, we define an $\mathsf{IND\text{-}CPA}$ adversary $\tilde{\mathcal{A}}$ against $\mathsf{KEM}^{\perp}_{m}$ in the QROM as follows. To avoid confusion, we denote the two random oracles quantum accessible to $\tilde{\mathcal{A}}$ in the $\mathsf{IND\text{-}CPA}$ game of $\mathsf{KEM}^{\perp}_{m}$ as H' and G'.

1. The input of $\tilde{\mathcal{A}}$ is (pk, c^*, K^*_b), where $c^* = \mathsf{Enc}_{pk}(m^*, H'(m^*))$.
2. $\tilde{\mathcal{A}}$ initializes register M with state $|0^m\rangle$, prepares database register D_{q_H}, and implements the extractable RO-simulator $\mathcal{S}(f_2) = \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_2}\}$. Then $\tilde{\mathcal{A}}$ runs adversary $\mathcal{A}$, simulates game $\mathbf{G_6}$ for it, and output $\mathcal{A}$'s output.

(a) When $\mathcal{A}$ queries random oracle H in parallel with state $|x_1, y_1\rangle_{\mathsf{X}_1\mathsf{Y}_1} \cdots |x_{w_H}, y_{w_H}\rangle_{\mathsf{X}_{w_H}\mathsf{Y}_{w_H}}$ on w_H pairs input/output registers, $\tilde{\mathcal{A}}$ answers it by applying a conditional operation to registers $\mathsf{X}_i\mathsf{Y}_i (i = 1, \ldots, w_H)$ sequentially:
   i. For the registers $\mathsf{X}_i\mathsf{Y}_i\mathsf{D}_{q_H}$, implement the RO-interface $\mathsf{eCO.RO}$ if $\mathsf{Enc}_{pk}(x_i, H'(x_i)) \neq c^*$, query random oracle H' with input/output register $\mathsf{X}_i/\mathsf{Y}_i$ if $\mathsf{Enc}_{pk}(x_i, H'(x_i)) = c^*$.

(b) When $\mathcal{A}$ queries random oracle G, $\tilde{\mathcal{A}}$ answers it by querying random oracle G' directly.

(c) When $\mathcal{A}$ queries decapsulation oracle qDeca with input state $|c, y\rangle_{\mathsf{IO}}$, $\tilde{\mathcal{A}}$ answers it by implementing unitary operation
$$\mathrm{U}_\perp \circ \mathrm{P}_{c^*} + \mathsf{Ext}_{f_2} \circ O_{G'} \circ \mathsf{Ext}_{f_2} \circ (\mathbf{I} - \mathrm{P}_{c^*})$$
on registers $\mathsf{IOD}_{q_H}\mathsf{M}$. Here, $O_{G'}$ represents querying random oracle G' with input/output register M/O.

One can check that, adversary $\tilde{\mathcal{A}}$ makes at most $2q_H$ (resp. $q_G + q_D$) queries to H' (resp. G'), the query depth of $\tilde{\mathcal{A}}$ to H' (resp. G') is $2d_H$ (resp. $d_G + q_D$). As for the running time, since $\tilde{\mathcal{A}}$ implements $\mathsf{eCO.RO}$ and $\mathsf{eCO.E}_{f_2}$ at most q_H and $2q_D$ times, respectively, the running time of $\tilde{\mathcal{A}}$ can be bounded as $\mathsf{Time}(\tilde{\mathcal{A}}) \approx \mathsf{Time}(\mathcal{A}) + O(q_H q_D + q_H^2)$ by the Definition 2. As for the memory space, note that $\tilde{\mathcal{A}}$ needs to prepare database register D_{q_H} to implement the extractable RO-simulator $\mathcal{S}(f_2)$, hence, we have $\mathsf{Space}(\tilde{\mathcal{A}}) \approx \mathsf{Space}(\mathcal{A}) + O(q_H)$.

Obviously, we have
$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}^{\perp}_{m}, \tilde{\mathcal{A}}} = \left| \Pr[1 \leftarrow \mathbf{G_6}] - \frac{1}{2} \right|. \tag{13}$$

Finally, combining Lemma 5 and Lemma 6 with (5), (6), (9), (11), (12) and (13), we obtain
$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}^{\perp}_{m}, \mathcal{A}} \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}^{\perp}_{m}, \tilde{\mathcal{A}}} + 8\sqrt{q_H(q_H + 1) \cdot \delta} + (64q_H + 2) \cdot \delta + 40q_D \cdot 2^{-\gamma/2}.$$

□

4 From $\mathsf{IND\text{-}CPA}_{\mathsf{P}}$ to $\mathsf{IND\text{-}CPA}_{\mathsf{FO}^{\perp}_{m}[\mathsf{P}]}$

In this section, we prove that, in the QROM, the $\mathsf{IND\text{-}CPA}$ security of KEM scheme $\mathsf{FO}^{\perp}_{m}[\mathsf{P}, H, G]$ can be reduced to the $\mathsf{IND\text{-}CPA}$ security of PKE scheme P without the quadratic security loss. Similar to Theorem 2, our reduction does not require the perfect correctness property of the PKE scheme P.

Before we prove the main result of this section, we first review the transformation T and $\mathsf{U}^{\perp}_{m}$ introduced in [14].

Transformation T: Let $\mathsf{P} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a randomized PKE scheme with message space $\mathcal{M}(= \{0,1\}^m)$ and randomness space $\{0,1\}^n$. Let $H : \mathcal{M} \to \{0,1\}^n$ be a hash function. We associate PKE scheme $\mathsf{T}[\mathsf{P}, H] := (\mathsf{Gen}, \mathsf{Enc}_1, \mathsf{Dec}_1)$. The constituting algorithms of $\mathsf{T}[\mathsf{P}, H]$ are given in Fig. 5.

$$
\begin{array}{ll}
\underline{\mathsf{Gen}} & \underline{\mathsf{Dec}_1\,(sk,c)} \\
(pk,sk) \leftarrow \mathsf{Gen} & m' = \mathsf{Dec}_{sk}\,(c) \\
\mathbf{return}\ (pk,sk) & \mathbf{if}\ m' = \perp \\
 & \quad \mathbf{return}\ \perp \\
\underline{\mathsf{Enc}_1\,(pk,m)} & \mathbf{else\ if}\ c \neq \mathsf{Enc}_{pk}\,(m';H(m')) \\
c = \mathsf{Enc}_{pk}\,(m;H(m)) & \quad \mathbf{return}\ \perp \\
\mathbf{return}\ c & \mathbf{return}\ m'
\end{array}
$$

Fig. 5. PKE scheme $\mathsf{T}[\mathsf{P},H] = (\mathsf{Gen},\mathsf{Enc}_1,\mathsf{Dec}_1)$.

We introduce the following two lemmas about transformation T. Note that the final upper bound of the first lemma avoids the quadratic security loss.

Lemma 8 (Security of T in the QROM [3], Theorem 1). *For any adversary $\mathcal{A}$ against the* OW-CPA *security of PKE scheme* $\mathsf{T}[\mathsf{P},H]$ *making q_H queries to H with depth d_H, there exists an adversary $\mathcal{B}$ against the* IND-CPA *security of PKE scheme* P *such that*

$$
\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{T}[\mathsf{P},H],\mathcal{A}} \leq (d_H+2)\cdot\left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{P},\mathcal{B}} + \frac{8(q_H+1)}{|\mathcal{M}|}\right),
$$

$\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$ *and* $\mathsf{Space}(\mathcal{B}) \approx \mathsf{Space}(\mathcal{A})$.

Lemma 9 ([21], Lemma 4). *Let* P=(Gen,Enc,Dec) *with message space $\mathcal{M}$ and randomness space $\{0,1\}^n$ be δ-correct. Define a set with respect to fixed $(pk,sk) \leftarrow \mathsf{Gen}$ and $H:\mathcal{M}\rightarrow\{0,1\}^n$:*

$$
S^{collision}_{pk,sk,H} := \{m \in \mathcal{M} | \exists m' \neq m, \mathsf{Enc}_{pk}(m',H(m')) = \mathsf{Enc}_{pk}(m,H(m))\}\,.
$$

Then we have

$$
\Pr[m \in S^{collision}_{pk,sk,H} | (pk,sk) \leftarrow \mathsf{Gen}, H \xleftarrow{\$} \Omega_H, m \xleftarrow{\$} \mathcal{M}] \leq 2\delta.
$$

$$
\begin{array}{lll}
\underline{\mathsf{Gen}} & \underline{\mathsf{Enca}\,(pk)} & \underline{\mathsf{Deca}\,(sk,c)} \\
(pk,sk) \leftarrow \mathsf{Gen} & m \xleftarrow{\$} \mathcal{M} & m' = \mathsf{dDec}_{sk}\,(c) \\
\mathbf{return}\ (pk,sk) & c = \mathsf{dEnc}_{pk}\,(m) & \mathbf{if}\ m' = \perp \\
 & K = G(m) & \quad \mathbf{return}\ \perp \\
 & \mathbf{return}\ (K,c) & \mathbf{else\ return}\ K = G(m')
\end{array}
$$

Fig. 6. KEM scheme $\mathsf{U}^{\perp}_{m}[\mathsf{dPKE},G] = (\mathsf{Gen},\mathsf{Enca},\mathsf{Deca})$.

Transformation $\mathsf{U}^{\perp}_{m}$: Let dPKE = (Gen,dEnc,dDec) be a DPKE scheme with message space $\mathcal{M}(=\{0,1\}^m)$. Let $G:\mathcal{M}\rightarrow\{0,1\}^n$ be a hash function. We associate KEM scheme $\mathsf{U}_m^{\perp}[\mathsf{dPKE}, G] := (\mathsf{Gen}, \mathsf{Enca}, \mathsf{Deca})$. The constituting algorithms of $\mathsf{U}_m^{\perp}[\mathsf{dPKE}, G]$ are given in Fig. 6.

Obviously, we have $\mathsf{FO}_m^{\perp}[\mathsf{P}, H, G] = \mathsf{U}_m^{\perp}[\mathsf{T}[\mathsf{P}, H], G]$. Next, we prove the following theorem, which indicates that the IND-CPA security of $\mathsf{U}_m^{\perp}[\mathsf{dPKE}, G]$ in the QROM can be reduced to the OW-CPA security of dPKE without the quadratic security loss.

Theorem 3 ($\mathbf{OW\text{-}CPA}_{\mathsf{dPKE}} \overset{\mathrm{QROM}}{\Rightarrow} \mathbf{IND\text{-}CPA}_{\mathsf{U}_m^{\perp}[\mathsf{dPKE,G}]}$). *Let $\mathcal{A}$ be an* IND-CPA *adversary against* $\mathsf{U}_m^{\perp}[\mathsf{dPKE,G}]$ *in the QROM making at most q_G queries to random oracle G with depth d_G. Then there exists an* OW-CPA *adversary $\tilde{\mathcal{A}}$ against* dPKE *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{U}_m^{\perp}[\mathsf{dPKE,G}],\mathcal{A}} \leq 2d_G \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{dPKE},\tilde{\mathcal{A}}} + 2d_G \cdot \Pr[E_{\mathsf{dPKE}}].$$

Here E_{dPKE} is the event that

$$E_{\mathsf{dPKE}} : m \xleftarrow{\$} \mathcal{M},\ \exists m' \neq m,\ \mathsf{dEnc}_{pk}(m) = \mathsf{dEnc}_{pk}(m').$$

The running time and memory space of $\tilde{\mathcal{A}}$ is bounded as $\mathsf{Time}(\tilde{\mathcal{A}}) \approx 2\cdot\mathsf{Time}(\mathcal{A}) + O(q_G)$ and $\mathsf{Space}(\tilde{\mathcal{A}}) \approx O(\mathsf{Space}(\mathcal{A}) + \mathsf{Time}(\mathcal{A}))$, respectively.

Proof. Define two games $\mathbf{G}_{b=0}$ and $\mathbf{G}_{b=1}$ as shown in Fig. 7. Here $\mathcal{D}$ is a joint distribution of (G, H, m^*, pk), where $G \xleftarrow{\$} \Omega_G$, $m^* \xleftarrow{\$} \mathcal{M}$, H is identical to G, except that $H(m^*)$ is a fresh random value uniformly sampled from $\{0,1\}^n$, and pk is sampled by $(pk, sk) \leftarrow \mathsf{Gen}$. Then we have

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{U}_m^{\perp}[\mathsf{dPKE},G],\mathcal{A}} = \frac{1}{2}|\Pr[1 \leftarrow \mathbf{G}_{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{b=1}]|. \tag{14}$$

$\underline{\mathbf{G}_{b=0}}$	$\underline{\mathbf{G}_{b=1}}$
1, $(G, H, m^*, pk) \leftarrow \mathcal{D}$	1, $(G, H, m^*, pk) \leftarrow \mathcal{D}$
2, $b = 0$	2, $b = 1$
$c^* = \mathsf{dEnc}_{pk}(m^*)$	$c^* = \mathsf{dEnc}_{pk}(m^*)$
$K_0^* = G(m^*)$, $K_1^* \xleftarrow{\$} \{0,1\}^n$	$K_0^* = G(m^*)$, $K_1^* \xleftarrow{\$} \{0,1\}^n$
3, $b' \leftarrow \mathcal{A}^G(pk, c^*, K_b^*)$	3, $b' \leftarrow \mathcal{A}^G(pk, c^*, K_b^*)$
4, **return** b'	4, **return** b'
$\underline{\mathbf{NG}_{b=0}}$	$\underline{\mathbf{NG}_{b=1}}$
1, $(G, H, m^*, pk, c^*, K) \leftarrow \mathcal{D}_1$	1, $(G, H, m^*, pk, c^*, K) \leftarrow \mathcal{D}_1$
2, $b' \leftarrow \mathcal{A}^G(pk, c^*, K)$	2, $b' \leftarrow \mathcal{A}^H(pk, c^*, K)$
3, **return** b'	3, **return** b'

Fig. 7. Game $\mathbf{G}_{b=0}$, $\mathbf{G}_{b=1}$, $\mathbf{NG}_{b=0}$ and $\mathbf{NG}_{b=1}$.

Next, we rewrite game $\mathbf{G}_{b=0}$ and $\mathbf{G}_{b=1}$ to new games $\mathbf{NG}_{b=0}$ and $\mathbf{NG}_{b=1}$, respectively, as shown in Fig. 7. The $\mathcal{D}_1$ in games $\mathbf{NG}_{b=0}$ and $\mathbf{NG}_{b=1}$ are joint distributions identical to $\mathcal{D}$, except that two additional values c^* and K are sampled, where $c^* = \mathsf{dEnc}_{pk}(m^*)$ and $K = G(m^*)$. Then we have

$$\Pr[1 \leftarrow \mathbf{G}_{b=0}] = \Pr[1 \leftarrow \mathbf{NG}_{b=0}],\ \Pr[1 \leftarrow \mathbf{G}_{b=1}] = \Pr[1 \leftarrow \mathbf{NG}_{b=1}]. \tag{15}$$

Define $z := (pk, c^*, K)$ and $z' := (G, H, m^*, pk, c^*, K)$, we obtain

$$\begin{aligned} \Pr[1 \leftarrow \mathbf{NG}_{b=0}] &= \Pr[1 \leftarrow \mathcal{A}^G(z) : z' \leftarrow \mathcal{D}_1], \\ \Pr[1 \leftarrow \mathbf{NG}_{b=1}] &= \Pr[1 \leftarrow \mathcal{A}^H(z) : z' \leftarrow \mathcal{D}_1]. \end{aligned} \tag{16}$$

By applying Lemma 2 with $\mathcal{X} = \mathcal{M}$, $\mathcal{Y} = \{0,1\}^n$, $S = \{m^*\}$, and $z = (pk, c^*, K)$, there exists an adversary $\mathcal{B}$ that makes oracle queries to G and H and satisfies

$$\begin{aligned} &|\Pr[1 \leftarrow \mathcal{A}^G(z) : z' \leftarrow \mathcal{D}_1] - \Pr[1 \leftarrow \mathcal{A}^H(z) : z' \leftarrow \mathcal{D}_1]| \\ &\qquad \le 4d_G \cdot \Pr[T \cap S \neq \varnothing : T \leftarrow \mathcal{B}^{G,H}(z), z' \leftarrow \mathcal{D}_1]. \end{aligned} \tag{17}$$

The running time of $\mathcal{B}$ is $\mathsf{Time}(\mathcal{B}) \approx 2 \cdot \mathsf{Time}(\mathcal{A})$, the memory space of $\mathcal{B}$ is $\mathsf{Space}(\mathcal{B}) \approx O(\mathsf{Space}(\mathcal{A}) + \mathsf{Time}(\mathcal{A}))$, and $\mathcal{B}$ makes at most $3q_G$ queries in total to oracles H and G.

Now, we construct an adversary $\tilde{\mathcal{A}}$ that against the $\mathsf{OW\text{-}CPA}$ security of dPKE as follows.

1. $\tilde{\mathcal{A}}$ gets the challenge ciphertext $c^* = \mathsf{dPKE}_{pk}(m^*)$ and public key pk.
2. $\tilde{\mathcal{A}}$ samples K uniformly from $\{0,1\}^n$ and chooses a $3q_G$-wise function f uniformly.
3. $\tilde{\mathcal{A}}$ uses (pk, c^*, K) as input to run adversary $\mathcal{B}$:
   (a) When $\mathcal{B}$ queries H with state $|x, y\rangle_{\mathsf{IO}}$ on input/output register I/O, $\tilde{\mathcal{A}}$ answers by applying unitary operation O_f to registers IO directly, where $O_f|x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$.
   (b) When $\mathcal{B}$ queries G with state $|x, y\rangle_{\mathsf{I}_1\mathsf{O}_1}$ on input/output register $\mathsf{I}_1/\mathsf{O}_1$, $\tilde{\mathcal{A}}$ answers by applying a conditional operation to registers $\mathsf{I}_1\mathsf{O}_1$:
   Apply O_f if $\mathsf{dPKE}_{pk}(x) \neq c^*$, apply U_K if $\mathsf{dPKE}_{pk}(x) = c^*$, where $\mathrm{U}_K|x, y\rangle_{\mathsf{I}_1\mathsf{O}_1} = |x, y \oplus K\rangle_{\mathsf{I}_1\mathsf{O}_1}$.
4. After $\mathcal{B}$ returns its output T, $\tilde{\mathcal{A}}$ searches x that satisfies $\mathsf{dPKE}_{pk}(x) = c^*$ from T and output the minimum one. If such x does not exist, $\tilde{\mathcal{A}}$ output $\perp$.

One can check that the running time of $\tilde{\mathcal{A}}$ is $\mathsf{Time}(\tilde{\mathcal{A}}) \approx \mathsf{Time}(\mathcal{B}) + O(q_G)$, the memory space of $\tilde{\mathcal{A}}$ is $\mathsf{Space}(\tilde{\mathcal{A}}) \approx \mathsf{Space}(\mathcal{B})$.

The adversary $\tilde{\mathcal{A}}$ cannot get m^* to simulate H and G directly. In the above construction, $\tilde{\mathcal{A}}$ tests if x equals m^* by checking if $\mathsf{dPKE}_{pk}(x)$ equals c^*. Therefore, similar to the event $m^* \notin \mathrm{S}_{pk,sk,O}^{collision}$ used in the game $\mathbf{G_6}$ of the proof of Theorem 2, if the following event E_{dPKE} does not occur, the adversary $\tilde{\mathcal{A}}$ simulates the oracle H and G for $\mathcal{B}$ perfectly.

$$E_{\mathsf{dPKE}}\colon m^* \xleftarrow{\$} \mathcal{M},\ \exists m' \neq m^*,\ \mathsf{dEnc}_{pk}(m^*) = \mathsf{dEnc}_{pk}(m').$$

Then, we have

$$\Pr[T \cap S \neq \varnothing : T \leftarrow \mathcal{B}^{G,H}(z), z' \leftarrow \mathcal{D}_1] \leq \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{dPKE},\tilde{\mathcal{A}}} + \Pr[E_{\mathsf{dPKE}}]. \tag{18}$$

Combining (14), (15), (16), (17) and (18), we finally obtain

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{U}^{\perp}_{m}[\mathsf{dPKE},G],\mathcal{A}} \leq 2d_G \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{dPKE},\tilde{\mathcal{A}}} + 2d_G \cdot \Pr[E_{\mathsf{dPKE}}].$$

□

Theorem 4 (IND-CPA$_{\mathsf{P}}$ $\overset{\text{QROM}}{\Rightarrow}$ IND-CPA$_{\mathsf{FO}^{\perp}_{m}[\mathsf{P},H,G]}$). *Let $\mathcal{A}$ be an* IND-CPA *adversary against $\mathsf{FO}^{\perp}_{m}[\mathsf{P},H,G]$ in the QROM that making at most q_H and q_G queries to random oracle H and G, respectively. Let d_H (resp. d_G) be the query depth of $\mathcal{A}$'s random oracle H (resp. G) queries. Then there exists an* IND-CPA *adversary $\mathcal{B}$ against* P *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{FO}^{\perp}_{m}[\mathsf{P},H,G],\mathcal{A}} \leq 2d_G(d_H+2) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{P},\mathcal{B}} + 16d_G(d_H+2)\frac{(q_H+1)}{|\mathcal{M}|} + 4d_G \cdot \delta.$$

The running time and memory space of $\mathcal{B}$ is bounded as $\mathsf{Time}(\mathcal{B}) \approx 2\cdot\mathsf{Time}(\mathcal{A}) + O(q_G)$ and $\mathsf{Space}(\mathcal{B}) \approx O(\mathsf{Space}(\mathcal{A}) + \mathsf{Time}(\mathcal{A}))$, respectively.

Proof. Since $\mathsf{FO}^{\perp}_{m}[\mathsf{P},H,G] = \mathsf{U}^{\perp}_{m}[\mathsf{T}[\mathsf{P},H],G]$, we have

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{FO}^{\perp}_{m}[\mathsf{P},H,G],\mathcal{A}} &= \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{U}^{\perp}_{m}[\mathsf{T}[\mathsf{P},H],G],\mathcal{A}} \\
&\overset{(a)}{\leq} 2d_G \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{T}[\mathsf{P},H],\tilde{\mathcal{A}}} + 2d_G \cdot \Pr[E_{\mathsf{T}[\mathsf{P},H]}] \\
&\overset{(b)}{\leq} 2d_G \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{T}[\mathsf{P},H],\tilde{\mathcal{A}}} + 4d_G \cdot \delta \\
&\overset{(c)}{\leq} 2d_G(d_H+2) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{P},\mathcal{B}} + 16d_G(d_H+2)\frac{(q_H+1)}{|\mathcal{M}|} + 4d_G \cdot \delta.
\end{aligned}$$

Here (a) and (c) uses the Theorem 3 and Lemma 8, respectively. (b) uses the Lemma 9.

By the result of Theorem 3, the running time of $\tilde{\mathcal{A}}$ is $\mathsf{Time}(\tilde{\mathcal{A}}) \approx 2\cdot\mathsf{Time}(\mathcal{A}) + O(q_G)$. By the result of Lemma 8, the running time of $\mathcal{B}$ is $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\tilde{\mathcal{A}})$. Therefore the running time of $\mathcal{B}$ is $\mathsf{Time}(\mathcal{B}) \approx 2\cdot\mathsf{Time}(\mathcal{A}) + O(q_G)$. The memory space of $\mathcal{B}$ can be obtained in a similar way. □

Combining Theorem 2 and Theorem 4, we obtain following result.

Corollary 1 (IND-CPA$_{\mathsf{P}}$ $\overset{\text{QROM}}{\Rightarrow}$ IND-qCCA$_{\mathsf{FO}^{\perp}_{m}[\mathsf{P},H,G]}$). *Let* P *be a randomized PKE scheme that is δ-correct and weakly γ-spread. Let $\mathcal{A}$ be an* IND-qCCA *adversary against $\mathsf{KEM}^{\perp}_{m} := \mathsf{FO}^{\perp}_{m}[\mathsf{P},H,G]$ in the QROM, making at most q_H, q_G and q_D queries to random oracle H, random oracle G and decapsulation oracle* qDeca^*, *respectively. Let d_H (resp. d_G) be the query depth of $\mathcal{A}$'s random oracle H (resp. G) queries.*

Then there exists an IND-CPA *adversary* $\mathcal{B}$ *against* P *such that*

$$\begin{aligned}\mathrm{Adv}^{\textsf{IND-qCCA}}_{\textsf{KEM}^{\perp}_{m},\mathcal{A}} \le\ & 2(d_G+q_D)(2d_H+2)\cdot \mathrm{Adv}^{\textsf{IND-CPA}}_{\textsf{P},\mathcal{B}} + 40q_D\cdot 2^{-\gamma/2} \\ & + 8\sqrt{q_H(q_H+1)\cdot\delta} + (64q_H+4d_G+4q_D+2)\cdot\delta \\ & + 16(d_G+q_D)(2d_H+2)\frac{(2q_H+1)}{|\mathcal{M}|}.\end{aligned}$$

The running time and memory space of $\mathcal{B}$ *is bounded as* $\mathsf{Time}(\mathcal{B}) \approx 2\cdot\mathsf{Time}(\mathcal{A}) + O(q_Hq_D+q_H^2+q_G)$ *and* $\mathsf{Space}(\mathcal{B}) \approx O(\mathsf{Space}(\mathcal{A})+\mathsf{Time}(\mathcal{A})+q_H)$*, respectively.*

5 Explicit Rejection and Implicit Rejection

In this section, we prove that, in the QROM, $\mathsf{FO}^{\not\perp}_{m}$ is IND-qCCA-secure if $\mathsf{FO}^{\perp}_{m}$ is IND-qCCA-secure and vice versa.

Transformation $\mathsf{FO}^{\not\perp}_{m}$: Let P = (Gen,Enc,Dec) be a randomized PKE scheme with meassage space $\mathcal{M}(=\{0,1\}^m)$ and randomness space $\{0,1\}^n$. Let $H:\mathcal{M}\to\{0,1\}^n$ and $G:\{0,1\}^*\to\{0,1\}^{n'}$ be hash functions. Let F be a pseudorandom function (PRF) with key space $\mathcal{K}^{prf}$. We associate

$$\mathsf{KEM}^{\not\perp}_{m} := \mathsf{FO}^{\not\perp}_{m}[\mathsf{P},H,G] = (\mathsf{Gen}^{\not\perp},\mathsf{Enca}_m,\mathsf{Deca}^{\not\perp}_{m}).$$

The constituting algorithms of $\mathsf{KEM}^{\not\perp}_{m}$ are given in Fig. 8.

$\underline{\mathsf{Gen}^{\not\perp}}$	$\underline{\mathsf{Encap}_m\,(pk)}$	$\underline{\mathsf{Deca}^{\not\perp}_{m}\,(sk'=(sk,s),c)}$
$(pk,sk)\leftarrow\mathsf{Gen}$	$m \xleftarrow{\$} \mathcal{M}$	$m'=\mathsf{Dec}_{sk}\,(c)$
$s\xleftarrow{\$}\mathcal{K}^{prf}$	$c=\mathsf{Enc}_{pk}\,(m;H(m))$	**if** $m'=\perp$
$sk':=(sk,s)$	$K=G(m)$	**return** $\mathsf{F}(s,c)$
return (pk,sk')	**return** (K,c)	**else if** $c\neq\mathsf{Enc}_{pk}\,(m';H(m'))$
		return $\mathsf{F}(s,c)$
		return $K=G(m')$

Fig. 8. KEM scheme $\mathsf{KEM}^{\not\perp}_{m} = (\mathsf{Gen}^{\not\perp},\mathsf{Enca}_m,\mathsf{Deca}^{\not\perp}_{m})$.

Theorem 5 (Explicit → implicit). *Let* P *be a randomized PKE scheme. Let* $\mathcal{A}$ *be an* IND-qCCA *adversary against* $\mathsf{KEM}^{\not\perp}_{m}$ *in the QROM. Then there exists an* IND-qCCA *adversary* $\mathcal{B}$ *against* $\mathsf{KEM}^{\perp}_{m}$ *such that*

$$\mathrm{Adv}^{\textsf{IND-qCCA}}_{\textsf{KEM}^{\not\perp}_{m},\mathcal{A}} = \mathrm{Adv}^{\textsf{IND-qCCA}}_{\textsf{KEM}^{\perp}_{m},\mathcal{B}}.$$

The running time and memory space of $\mathcal{B}$ *is bounded as* $\mathsf{Time}(\mathcal{A})\approx\mathsf{Time}(\mathcal{B})$ *and* $\mathsf{Space}(\mathcal{A})\approx\mathsf{Space}(\mathcal{B})$*, respectively.*

Proof. The only difference between the adversary in the $\mathsf{IND\text{-}qCCA}$ game of $\mathsf{KEM}_m^{\perp}$ and $\mathsf{KEM}_m^{\not\perp}$ is that the former gets $\perp$ from the decapsulation oracle for an input c failed to decapsulate, the latter instead gets pseudorandom value $\mathsf{F}(s,c)$. Indeed, the former adversary can also choose s itself and compute $\mathsf{F}(s,c)$ after it gets $\perp$ from the decapsulation oracle for input c. Following this way, we construct an adversary $\mathcal{B}$ against the $\mathsf{IND\text{-}qCCA}$ security of $\mathsf{KEM}_m^{\perp}$ as follows:

1. $\mathcal{B}$ chooses PRF key $s \xleftarrow{\$} \mathcal{K}^{prf}$ and runs adversary $\mathcal{A}$.
2. $\mathcal{B}$ answers the random oracle H/G queries of $\mathcal{A}$ by querying H/G directly.
3. $\mathcal{B}$ initializes a register K defined over $\{0,1\}^{n'+1}$[18] with state $|0^{n'}\rangle_{\mathsf{K}}$. When $\mathcal{A}$ queries the decapsulation oracle with input state $|c\rangle_{\mathsf{I}}|y\rangle_{\mathsf{O}}$, $\mathcal{B}$ answers by applying following operations sequentially:
   (a) Query the decapsulation oracle with input state $|c\rangle_{\mathsf{I}}|0^{n'}\rangle_{\mathsf{K}}$, suppose the output state is $|c\rangle_{\mathsf{I}}|k\rangle_{\mathsf{K}}$.
   (b) If $k=\perp$, perform unitary operation $\mathrm{U}_s : |c\rangle_{\mathsf{I}}|y\rangle_{\mathsf{O}} \to |c\rangle_{\mathsf{I}}|y \oplus \mathsf{F}(s,c)\rangle_{\mathsf{O}}$. Otherwise, perform unitary operation $\mathrm{U}_{XOR} : |y\rangle_{\mathsf{O}}|k\rangle_{\mathsf{K}} \to |y \oplus k\rangle_{\mathsf{O}}|k\rangle_{\mathsf{K}}$.
   (c) Query the decapsulation oracle with input state $|c\rangle_{\mathsf{I}}|k\rangle_{\mathsf{K}}$, now the register K is guaranteed to contain $0^{n'}$.
4. $\mathcal{B}$ finally outputs $\mathcal{A}$'s output.

Obviously, adversary $\mathcal{B}$ perfectly simulates the $\mathsf{IND\text{-}qCCA}$ game of $\mathsf{KEM}_m^{\not\perp}$ for adversary $\mathcal{A}$ and the running time (resp. memory space) of $\mathcal{B}$ is nearly the same as the running time (resp. memory space) of $\mathcal{A}$. Thus

$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}_m^{\not\perp},\mathcal{A}} = \mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}_m^{\perp},\mathcal{B}}.$$

□

Theorem 6 (Implicit → explicit). *Let* P *be a randomized PKE scheme that is* δ*-correct and weakly* γ*-spread. Let* $\mathcal{A}$ *be an* $\mathsf{IND\text{-}qCCA}$ *adversary against* $\mathsf{KEM}_m^{\perp}$ *that making at most* q_H*,* q_G *and* q_D *queries to random oracle* H*, random oracle* G *and decapsulation oracle* qDeca^**, respectively.*

Then there exists an $\mathsf{IND\text{-}qCCA}$ *adversary* $\mathcal{B}$ *against* $\mathsf{KEM}_m^{\not\perp}$ *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}_m^{\perp},\mathcal{A}} \leq \mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}_m^{\not\perp},\mathcal{B}} + 8\sqrt{q_H(q_H+1)\cdot\delta} + (64q_H+2)\cdot\delta + 40q_D\cdot 2^{-\gamma/2}.$$

The running time and memory space of $\mathcal{B}$ *is bounded as* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + O(q_H q_D + q_H^2)$ *and* $\mathsf{Space}(\mathcal{B}) \approx \mathsf{Space}(\mathcal{A}) + O(q_H)$*, respectively.*

Proof. By using Theorem 2, there exists an $\mathsf{IND\text{-}CPA}$ adversary $\tilde{\mathcal{A}}$ against $\mathsf{KEM}_m^{\perp}$ such that

$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}_m^{\perp},\mathcal{A}} \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}_m^{\perp},\tilde{\mathcal{A}}} + 8\sqrt{q_H(q_H+1)\cdot\delta} + (64q_H+2)\cdot\delta + 40q_D\cdot 2^{-\gamma/2}. \quad (19)$$

[18] Here we embed the set $\{0,1\}^{n'} \cup \perp$ into the set $\{0,1\}^{n'+1}$ as explained in Appendix A.

The running time and memory space of $\tilde{\mathcal{A}}$ is bounded as $\mathsf{Time}(\tilde{\mathcal{A}}) \approx \mathsf{Time}(\mathcal{A}) + O(q_H q_D + q_H^2)$ and $\mathsf{Space}(\tilde{\mathcal{A}}) \approx \mathsf{Space}(\mathcal{A}) + O(q_H)$, respectively.

We note that, in the IND-qCCA game of $\mathsf{KEM}_m^{\not\perp}$, the PRF key s chosen as part of the secret key is useless if the adversary never queries the decapsulation oracle. This implies that, even though the IND-qCCA adversary against $\mathsf{KEM}_m^{\not\perp}$ does not know the PRF key s, it can still perfectly simulate the IND-CPA game of $\mathsf{KEM}_m^{\perp}$ for the adversary $\tilde{\mathcal{A}}$. Now, we construct an IND-qCCA adversary $\mathcal{B}$ against $\mathsf{KEM}_m^{\not\perp}$ as follows:

1. $\mathcal{B}$ runs adversary $\tilde{\mathcal{A}}$ and $\mathcal{B}$ never queries the decapsulation oracle.
2. $\mathcal{B}$ answers the random oracle H/G queries of $\tilde{\mathcal{A}}$ by querying H/G directly.
3. $\mathcal{B}$ finally outputs $\tilde{\mathcal{A}}$'s output.

It is straightforward to check that adversary $\mathcal{B}$ perfectly simulates the IND-CPA game of $\mathsf{KEM}_m^{\perp}$ for adversary $\tilde{\mathcal{A}}$, and the running time (resp. memory space) of $\mathcal{B}$ is nearly the same as the running time (resp. memory space) of $\tilde{\mathcal{A}}$. Thus

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}_m^{\perp},\tilde{\mathcal{A}}} = \mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM}_m^{\not\perp},\mathcal{B}}.$$

Combining above equation with (19), we obtain our result. □

Remark 2. In Theorem 6, different from Corollary 1, we note that our reduction only introduces a linear memory space expansion $O(q_H)$. The reason is that the adversary $\tilde{\mathcal{A}}$ in Theorem 2 only invokes adversary $\mathcal{A}$ once in a black-box manner, and it just uses an additional database register D_{q_H} to process the oracle queries of $\mathcal{A}$.

Acknowledgments. We thank the anonymous reviewers of CRYPTO 2023, and Shujiao Cao for their insightful comments and suggestions. This work is supported by National Natural Science Foundation of China (Grants No. 62172405).

References

1. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. In: Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II. pp. 269–295. Springer (2019). https://doi.org/10.1007/978-3-030-26951-7_10
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993. pp. 62–73. ACM (1993). https://doi.org/10.1145/168588.168596
3. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of CCA security in the quantum random oracle model. In: Theory of Cryptography Conference. pp. 61–90. Springer (2019). https://doi.org/10.1007/978-3-030-36033-7_3
4. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. pp. 41–69. Springer (2011). https://doi.org/10.1007/978-3-642-25385-0_3
5. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II. pp. 361–379. Springer (2013). https://doi.org/10.1007/978-3-642-40084-1_21
6. Chung, K., Fehr, S., Huang, Y., Liao, T.: On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In: Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II. pp. 598–629. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_21
7. Czajkowski, J., Majenz, C., Schaffner, C., Zur, S.: Quantum lazy sampling and game-playing proofs for quantum indifferentiability. Cryptology ePrint Archive, Paper 2019/428 (2019), https://eprint.iacr.org/2019/428, https://eprint.iacr.org/2019/428
8. Dent, A.W.: A designer's guide to kems. In: IMA International Conference on Cryptography and Coding. pp. 133–151. Springer (2003). https://doi.org/10.1007/978-3-540-40974-8_12
9. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Online-extractability in the quantum random-oracle model. In: Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III. pp. 677–706. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_24
10. Duman, J., Hartmann, D., Kiltz, E., Kunzweiler, S., Lehmann, J., Riepel, D.: Group action key encapsulation and non-interactive key exchange in the QROM. In: Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13792, pp. 36–66. Springer (2022). https://doi.org/10.1007/978-3-031-22966-4_2

11. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptol. **26**(1), 80–101 (2013). https://doi.org/10.1007/s00145-011-9114-1
12. Ge, J., Shan, T., Xue, R.: On the fujisaki-okamoto transform: from classical cca security to quantum cca security. Cryptology ePrint Archive, Paper 2023/792 (2023), https://eprint.iacr.org/2023/792, https://eprint.iacr.org/2023/792
13. Grubbs, P., Maram, V., Paterson, K.G.: Anonymous, robust post-quantum public key encryption. In: Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III. pp. 402–432. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_15
14. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki-okamoto transformation. In: Theory of Cryptography Conference. pp. 341–371. Springer (2017). https://doi.org/10.1007/978-3-319-70500-2_12
15. Hövelmanns, K., Hülsing, A., Majenz, C.: Failing gracefully: Decryption failures and the fujisaki-okamoto transform. In: Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV. pp. 414–443. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_15
16. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III. pp. 96–125. Springer (2018). https://doi.org/10.1007/978-3-319-96878-0_4
17. Jiang, H., Zhang, Z., Ma, Z.: Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In: Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II. pp. 618–645. Springer (2019). https://doi.org/10.1007/978-3-030-17259-6_21
18. Jiang, H., Zhang, Z., Ma, Z.: Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In: Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers. pp. 227–248. Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_13
19. Jiang, H., Zhang, Z., Ma, Z.: On the non-tightness of measurement-based reductions for key encapsulation mechanism in the quantum random oracle model. In: Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13090, pp. 487–517. Springer (2021). https://doi.org/10.1007/978-3-030-92062-3_17
20. Kuchta, V., Sakzad, A., Stehlé, D., Steinfeld, R., Sun, S.: Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In: Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III. pp. 703–728. Springer (2020). https://doi.org/10.1007/978-3-030-45727-3_24
21. Liu, X., Wang, M.: Qcca-secure generic key encapsulation mechanism with tighter security in the quantum random oracle model. In: Public-Key Cryptography - PKC

2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part I. pp. 3–26. Springer (2021). https://doi.org/10.1007/978-3-030-75245-3_1
22. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information (10th Anniversary edition). Cambridge University Press (2016)
23. NIST: National institute for standards and technology. post quantum crypto project. https://csrc.nist.gov/projects/post-quantum-cryptography (2017)
24. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III. pp. 520–551. Springer (2018). https://doi.org/10.1007/978-3-319-78372-7_17
25. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. IACR Cryptol. ePrint Arch. p. 332 (2004)
26. Unruh, D.: Revocable quantum timed-release encryption. J. ACM **62**(6), 49:1–49:76 (2015). https://doi.org/10.1145/2817206
27. Xagawa, K., Yamakawa, T.: (tightly) qcca-secure key-encapsulation mechanism in the quantum random oracle model. In: Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers. pp. 249–268. Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_14
28. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. pp. 758–775. Springer (2012). https://doi.org/10.1007/978-3-642-32009-5_44
29. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II. pp. 239–268. Springer (2019). https://doi.org/10.1007/978-3-030-26951-7_9

A Quantum Background

A quantum system (register) Q is a complex Hilbert space $\mathcal{H}_Q$ with an inner product $\langle\cdot|\cdot\rangle$, notation like '$|\cdot\rangle$' or '$\langle\cdot|$' is called the Dirac notation. We denote $\mathcal{H}_Q = \mathbb{C}[\mathcal{X}]$ if Q is defined over a finite set $\mathcal{X}$, the orthonormal basis of $\mathbb{C}[\mathcal{X}]$ is $\{|x\rangle\}_{x\in\mathcal{X}}$, where the basis state $|x\rangle$ is labeled by the element x of $\mathcal{X}$. We refer to $\{|x\rangle\}_{x\in\mathcal{X}}$ as the computational basis. The state $|\psi\rangle$ of the quantum system Q is a unit vector, and we also write this state as $|\psi\rangle_Q$.

A qubit in superposition is a linear combination vector $|b\rangle = \alpha|0\rangle + \beta|1\rangle$ of two computational basis states $|0\rangle$ and $|1\rangle$ with $\alpha, \beta \in \mathbb{C}^2$ and $|\alpha|^2+|\beta|^2= 1$, α, β are the probability amplitudes of $|b\rangle$. Given quantum systems Q_1 and Q_2, we call tensor product $Q_1 \otimes Q_2$ is the composite quantum system and the product state is $|\psi_1\rangle \otimes |\psi_2\rangle \in Q_1 \otimes Q_2$ where $|\psi_1\rangle \in Q_1$, $|\psi_2\rangle \in Q_2$. An n-qubit system is $Q^{\otimes n}$ where Q is a single qubit system. We call state $|\psi\rangle \in Q_1 \otimes Q_2$ a product state if $|\psi\rangle$ can be rewritten as $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ and $|\psi_1\rangle \in Q_1$, $|\psi_2\rangle \in Q_2$, if $|\psi\rangle$ is not a product state, we say that the systems Q_0 and Q_1 are entangled, otherwise un-entangled. The norm of a state $|\psi\rangle$ is defined as $\||\psi\rangle\| := \sqrt{\langle\psi|\psi\rangle}$, where $\langle\psi|\psi\rangle$ is the inner product of $|\psi\rangle$.

The evolution of a closed quantum system is described by a unitary operation. That is the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operation U which depends only on the times t_1 and t_2, that $|\psi'\rangle = \mathrm{U}|\psi\rangle$. In our paper, we also write U_Q to emphasize that the unitary operation U acts on the quantum system (register) Q. For any unitary operation U acts on a quantum system, we have $\mathrm{U} \circ \mathrm{U}^\dagger = \mathbf{I}$, where $\mathrm{U}^\dagger$ is the Hermitian transpose of U and $\mathbf{I}$ is the identity operator over the quantum system. The norm of an operator U is defined as $\|\mathrm{U}\| := \max_{\||\Phi\rangle\|=1} \|\mathrm{U}|\Phi\rangle\|$.

Then we introduce a special operation called projector, for state $|\psi\rangle$ of an n-qubit register, a projector $\mathrm{M}_{|y\rangle\langle y|}$ applies the projection $|y\rangle\langle y|$ map to the state $|\psi\rangle$ to get the new state $|y\rangle\langle y|\psi\rangle$. $\mathrm{M}_{|y\rangle\langle y|}$ can also be generalized to a new projector $\mathrm{M}_{y\in S}$ which applies the projection $\sum_{y\in S} |y\rangle\langle y|$. We stress that any projector operator M is Hermitian (i.e., we have $\mathrm{M}^\dagger = \mathrm{M}$) and idempotent (i.e., we have $\mathrm{M}^2 = \mathrm{M}$).

State $|\psi\rangle$ can be measured with respect to a basis, for example, suppose $|\psi\rangle = \sum_x \alpha_x|x\rangle$ with computational basis $\{|x\rangle\}$, if we measure $|\psi\rangle$ in computational basis, the measurement outputs the value x with probability $|\langle x|\psi\rangle|^2 = |\alpha_x|^2$. Note that state $|\psi\rangle$ collapses to state $|x\rangle$ after the measurement, so the state will stay $|x\rangle$, and the subsequent measurements will always output x. Measurements on other basis are defined analogously. In this paper, we will generally only consider measurements on the computational basis. A general projective measurement $\mathbb{M}$ is defined by a set of projection operators $\mathrm{M}_1, \ldots, \mathrm{M}_n$ where M_i are mutually orthogonal and $\sum_{i=1}^n \mathrm{M}_i = \mathbf{I}$. Any general projective measurement can be implemented by composing a unitary operation followed by a measurement in the computational basis.

A quantum oracle algorithm $\mathcal{A}^O(z)$ is an algorithm $\mathcal{A}(z)$ that is given quantum oracle access to oracle O. In this paper, we default that oracle O can be implemented by a unitary operation U_O that operates on the corresponding input/output register. The algorithm $\mathcal{A}(z)$ is allowed to perform parallel queries to O with input/output register I_i/O_i for $i = 1, \ldots, w$, suppose $\mathcal{A}(z)$ can perform parallel queries at most d times, then we call w (resp. d) the query width (resp. query depth) and the total query times of $\mathcal{A}(z)$ is $q := w \cdot d$. Moreover, once parallel query to O with input/output register I_i/O_i for $i = 1, \ldots, w$ can be implemented by unitary operation $(\mathrm{U}_O)^{\otimes w}$

There is a well-known fact that we can construct a unitary variant $\mathcal{A}^O_{\mathrm{U}}(z)$ for any quantum oracle algorithm $\mathcal{A}^O(z)$ with some constant factor computational overhead and these two algorithms have same query width and query depth [1], $\mathcal{A}^O_{\mathrm{U}}(z)$ also called a unitary quantum oracle algorithm. As shown in Definition 8 of [10], the detailed execution of a unitary quantum oracle algorithm can be described as follows:

Unitary quantum oracle algorithm $\mathcal{B}^O$: Suppose $\mathcal{B}$'s query depth is d and query width is p, then $\mathcal{B}$'s execution can be described as

$$\mathrm{U}_d \circ (\mathrm{U}_O)^{\otimes p} \circ \mathrm{U}_{d-1} \circ (\mathrm{U}_O)^{\otimes p} \circ \ldots \circ \mathrm{U}_1 \circ (\mathrm{U}_O)^{\otimes p}|\psi\rangle.$$

Here $\mathrm{U}_1, \ldots, \mathrm{U}_d$ is the fixed unitary operations applied between queries, and $|\psi\rangle$ is the initial pure state. $\mathcal{B}$ perform a projective measurement on its quantum register after applying U_d and output the measure outcome. For multiple oracles case, as explained in the Remark 8 of [10], if $\mathcal{B}$ have quantum access to all oracles, then the execution of $\mathcal{B}$ can be described analogously.

Moreover, in this paper, we sometimes use a special symbol $\perp$ to expand a finite set $\{0,1\}^n$, thus default $\perp \notin \{0,1\}^n$ and then consider a new finite set $\{0,1\}^n \cup \perp$. Roughly speaking, the reason is that, when we define a special unitary operation, we need $\perp$ to denote "not defined (yet)" or "computation failure".

As for the detailed representation of $\{0,1\}^n \cup \perp$, we use the extension method introduced in [6]. That is to say, we use a classical encoding function *enc* that $enc(\perp) = 1||0^n \in \{0,1\}^{n+1}$ and $enc(x) = 0||x \in \{0,1\}^{n+1}$ for any $x \in \{0,1\}^n$, then the set $\{0,1\}^n \cup \perp$ can be embedded into the set $\{0,1\}^{n+1}$. Under this representation, the binary operation $x \oplus y$ for $x, y \in \{0,1\}^n \cup \perp$ that used in this paper actually means $enc(x) \oplus enc(y)$, where operation $\oplus$ denotes bitwise addition modulo 2, a group operation on $\{0,1\}^{n+1}$. Overall, with this representation, the quantum register defined over set $\{0,1\}^n \cup \perp$ is implemented by a quantum register defined over set $\{0,1\}^{n+1}$.

B Cryptographic Primitives and Security Definitions

Definition 3 (Public Key Encryption). *A public key encryption (PKE) scheme consist of a finite message space* $\mathcal{M}$ *and three polynomial algorithm* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *according to security parameter* λ*.*

1. Gen*: a probabilistic algorithm with input* 1^λ *and output a public/secret key pair* (pk, sk)*.*
2. Enc*: a probabilistic algorithm with input a message* $m \in \mathcal{M}$ *and output a ciphertext* $c \in \mathcal{C}$ *(*$\mathcal{C}$ *is the ciphertext space). it choose* $r \leftarrow \mathcal{R}$*(*$\mathcal{R}$ *is the randomness space), computes* $c := \mathsf{Enc}_{pk}(m, r)$ *and output ciphertext* c*. If* Enc *do not use randomness to compute* c*,* Enc *is a deterministic algorithm and output* $c := \mathsf{Enc}_{pk}(m)$*.*
3. Dec*: a deterministic algorithm with input a ciphertext* $c \in \mathcal{C}$ *and secret key* sk*, computes* $m := \mathsf{Dec}_{sk}(c)$ *and output* m *or a rejection symbol* $\perp \notin \mathcal{M}$*.*

Definition 4 (OW-CPA/IND-CPA secure). *We say* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* OW-CPA *(resp.* IND-CPA*) secure if for any quantum polynomial adversary* $\mathcal{A}$*, the* OW-CPA *(resp.* IND-CPA*) advantage of* $\mathcal{A}$ *against* PKE *defined as*

$$\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}} := \Pr[1 \leftarrow \mathrm{Game}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}] \textit{ (resp.}$$
$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}} := |\Pr[1 \leftarrow \mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}] - 1/2|\textit{)}$$

is negligible. The game $\mathrm{Game}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}$ *(resp.* $\mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}$*) is defined in Fig. 9.*

$\underline{\mathrm{Game}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}}$	$\underline{\mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}}$
$(pk, sk) \leftarrow \mathsf{Gen}$	$(pk, sk) \leftarrow \mathsf{Gen}$
$m^* \xleftarrow{\$} \mathcal{M}$	$b \xleftarrow{\$} \{0,1\}$
$c^* = \mathsf{Enc}_{pk}(m^*)$	$(m_0^*, m_1^*) \leftarrow \mathcal{A}(pk)$
$m' \leftarrow \mathcal{A}(pk, c^*)$	$c^* = \mathsf{Enc}_{pk}(m_b^*)$
return $[m' = m^*]$	$b' \leftarrow \mathcal{A}(pk, c^*)$
	return $[b' = b]$

Fig. 9. Game $\mathrm{Game}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}$ and $\mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}$.

Definition 5 (Correctness [14]). *We say that* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct if*

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

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}$*. Define*

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

then we have $\mathbb{E}[\delta(pk, sk)] \leq \delta$*.*

Definition 6 (weakly γ-spread [9]). *We say that* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is weakly γ-spread if*

$$\mathbb{E}\left[\max_{m \in \mathcal{M}, c \in \mathcal{C}} \Pr[c = \mathsf{Enc}_{pk}(m)]\right] \leq 2^{-\gamma},$$

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}$ *and the probability is over the randomness of the encryption. We also define*

$$\gamma(pk, sk) := \max_{m \in \mathcal{M}, c \in \mathcal{C}} \Pr[c = \mathsf{Enc}_{pk}(m)].$$

Definition 7 (Key-encapsulation mechanism). *A key-encapsulation mechanism (KEM) consists of three algorithms* Gen*,* Enca *and* Deca*. The key generation algorithm* Gen *outputs a key pair* (pk, sk)*. The encapsulation algorithm* Enca*, 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* Deca*, on input sk and an encapsulation c, outputs either a key* $K := \mathsf{Deca}_{sk}(c) \in \mathcal{K}$ *or a special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid encapsulation.*

Definition 8 (IND-qCCA/IND-CPA secure). *We say* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enca}, \mathsf{Deca})$ *is* IND-qCCA *(resp.* IND-CPA*) secure if for any quantum polynomial adversary $\mathcal{A}$, the* IND-qCCA *(resp.* IND-CPA*) advantage of $\mathcal{A}$ against* KEM *defined as*

$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{KEM},\mathcal{A}} := |\Pr[1 \leftarrow \mathrm{Game}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}}] - 1/2| \text{ (resp.}$$
$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM},\mathcal{A}} := |\Pr[1 \leftarrow \mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{KEM}}] - 1/2|\text{)}$$

is negligible. The game $\mathrm{Game}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}}$ *(resp.* $\mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{KEM}}$*) is defined in Fig. 10.*

$\mathrm{Game}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}}$	$\mathsf{qDeca}^*(\sum_{c,k} \alpha_{c,k} \lvert c, k\rangle)$	$\mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{KEM}}$
$(pk, sk) \leftarrow \mathsf{Gen}$	**return** $\sum_{c,k} \alpha_{c,k} \lvert c, k \oplus f_{c^*}(c)\rangle$	$(pk, sk) \leftarrow \mathsf{Gen}$
$b \xleftarrow{\$} \{0,1\}$		$b \xleftarrow{\$} \{0,1\}$
$(c^*, k_0^*) \leftarrow \mathsf{Enca}(pk)$	$f_{c^*}(c)$	$(c^*, k_0^*) \leftarrow \mathsf{Enca}(pk)$
$k_1^* \xleftarrow{\$} \mathcal{K}$	**if** $c = c^*$	$k_1^* \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\mathsf{qDeca}^*}(pk, c^*, k_b^*)$	**return** $\perp$	$b' \leftarrow \mathcal{A}(pk, c^*, k_b^*)$
return $[b' = b]$	**else return** $\mathsf{Deca}_{sk}(c)$	**return** $[b' = b]$

Fig. 10. Game $\mathrm{Game}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}}$ and $\mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{KEM}}$.

C The Quantum Circuit Implementation of $\mathrm{U}_{\boldsymbol{m}}$

Define function $f : \mathcal{C} \rightarrow \mathcal{M} \cup \perp$ and $g : \mathcal{M} \cup \perp \times \{0,1\}^n \cup \perp \times \mathcal{C} \rightarrow \{0,1\}$ as:

$$f(c) = \mathsf{Dec}_{sk}(c), \quad g(x,y,c) = \begin{cases} 0 \text{ if } \mathsf{Enc}_{pk}(x,y) = c \wedge x,y \neq \perp \\ 1 \text{ otherwise.} \end{cases}$$

Obviously, function f and g can be efficiently computed. Thus, the unitary operation $\mathrm{U}_f : |c,z\rangle \rightarrow |c, z \oplus f(c)\rangle$ and $\mathrm{U}_g : |x,y,c,b\rangle \rightarrow |x,y,c,b \oplus g(x,y,c)\rangle$ can also be efficiently implemented by the basic theory of quantum computation.

By using U_f and U_g above, unitary operation U_m can be implemented by the following procedure:

- Initialize three new registers R_1, R_2 and R_3 to 0, here R_3 is a one qubit register.
- Apply U_f to registers IR_1, here R_1 is the output register. Then apply U_f to registers IM, here M is the output register.
- Query H by registers $\mathsf{R}_1\mathsf{R}_2$, here R_2 is the output register and we default $H(\perp) = \perp$.
- Apply U_g to registers $\mathsf{IR}_1\mathsf{R}_2\mathsf{R}_3$, here R_3 is the output register.
- Apply the following two conditional operations.
  - The controlling bit is R_3, and apply U_f to registers IM if $b = 1$, here M is the output register.
  - The controlling bit is R_3, and apply unitary operation $\mathrm{U}_\perp$ to register M if $b = 1$, where $\mathrm{U}_\perp|0^m\rangle = |\perp\rangle$, $\mathrm{U}_\perp|\perp\rangle = |0^m\rangle$.
- Apply U_g to registers $\mathsf{IR}_1\mathsf{R}_2\mathsf{R}_3$, here R_3 is the output register.
- Query H by registers $\mathsf{R}_1\mathsf{R}_2$, here R_2 is the output register.
- Apply U_f to registers IR_1, here R_1 is the output register. Now the registers R_1, R_2 and R_3 are guaranteed to contain 0, so they can be discarded.

We stress that two queries to H is needed in above procedure.

D Missing Proofs of Section 3

Here we introduce the following corollary, which will be used in the proof of Lemma 5.

Corollary 2. *For any state $|\psi_1\rangle$ to $|\psi_q\rangle$, we have $\| \sum_{i=1}^{q} |\psi_i\rangle\|^2 \leq q \cdot \sum_{i=1}^{q} \||\psi_i\rangle\|^2$.*

Proof. The proof is simple:

$$\left\| \sum_{i=1}^{q} |\psi_i\rangle \right\|^2 \overset{(a)}{\leq} \left(\sum_{i=1}^{q} \||\psi_i\rangle\| \right)^2 \overset{(b)}{\leq} q \cdot \sum_{i=1}^{q} \||\psi_i\rangle\|^2.$$

Here (a) uses the triangle inequality, and (b) uses the AM-QM (or Jensens) inequality.

D.1 Proof of Lemma 5

Proof. Obviously, we can construct an oracle algorithm $\mathcal{B}^{\mathsf{qDeca}^\diamond}(pk, sk, G)$ to execute game $\mathbf{G_2}$. The algorithm generates the challenge ciphertext (c^*, K_b^*) and runs adversary $\mathcal{A}$ to get b'. It finally outputs $[b = b']$. Algorithm $\mathcal{B}^{\mathsf{qDeca}^\diamond}(pk, sk, G)$ prepares database register D_{q_H} and implements the extractable RO-simulator $\mathcal{S}(f_1)$ itself. The queries to $\mathsf{qDeca}^\diamond$ made by algorithm $\mathcal{B}^{\mathsf{qDeca}^\diamond}(pk, sk, G)$ can be answered by applying unitary operation $\mathrm{U}^2_{\mathsf{qD}}$ to registers $\mathsf{IOD}_{q_H}\mathsf{M}$[19]. Then, if we change $\mathsf{qDeca}^\diamond$ into qDeca^* that is answered by applying $\mathrm{U}^1_{\mathsf{qD}}$, we get an oracle algorithm $\mathcal{B}^{\mathsf{qDeca}^*}(pk, sk, G)$ that runs game $\mathbf{G_1}$. Therefore,

$$\begin{aligned}\Pr\left[1 \leftarrow \mathcal{B}^{\mathsf{qDeca}^*}(pk, sk, G)\right] &= \Pr\left[1 \leftarrow \mathbf{G_1} : (pk, sk, G)\right], \\ \Pr\left[1 \leftarrow \mathcal{B}^{\mathsf{qDeca}^\diamond}(pk, sk, G)\right] &= \Pr\left[1 \leftarrow \mathbf{G_2} : (pk, sk, G)\right].\end{aligned} \tag{20}$$

Here $\Pr[1 \leftarrow \mathbf{G_i} : (pk, sk, G)]$ is the probability that game $\mathbf{G_i}$ outputs 1 for fixed (pk, sk) and G.

As explained in Appendix A, for oracle algorithm $\mathcal{B}^O(pk, sk, G)$, we can construct its unitary variant $\mathcal{B}^O_U(pk, sk, G)$ that acts on registers ZIOD_{q_H}. Here register Z contains the adversary $\mathcal{A}$'s register A and the other registers used by $\mathcal{B}^O_U$. Indeed, the corresponding final joint state of $\mathcal{B}^{\mathsf{qDeca}^*}_U(pk, sk, G)$ and $\mathcal{B}^{\mathsf{qDeca}^\diamond}_U(pk, sk, G)$ just before the projective measurement $\mathbb{M} := \left\{\mathrm{M}_{|0\rangle\langle 0|}, \mathrm{M}_{|1\rangle\langle 1|}\right\}$ can be written as:

$$\begin{aligned}\mathcal{B}^{\mathsf{qDeca}^*}_U &: |\Psi_1\rangle|0^m\rangle_{\mathsf{M}} = \mathrm{U}_{q_D} \circ \mathrm{U}^1_{\mathsf{qD}} \cdots \mathrm{U}_2 \circ \mathrm{U}^1_{\mathsf{qD}} \circ \mathrm{U}_1 \circ \mathrm{U}^1_{\mathsf{qD}}|\psi\rangle|0^m\rangle_{\mathsf{M}}, \\ \mathcal{B}^{\mathsf{qDeca}^\diamond}_U &: |\Psi_2\rangle|0^m\rangle_{\mathsf{M}} = \mathrm{U}_{q_D} \circ \mathrm{U}^2_{\mathsf{qD}} \cdots \mathrm{U}_2 \circ \mathrm{U}^2_{\mathsf{qD}} \circ \mathrm{U}_1 \circ \mathrm{U}^2_{\mathsf{qD}}|\psi\rangle|0^m\rangle_{\mathsf{M}}.\end{aligned}$$

[19] This might be confusing because algorithm $\mathcal{B}$ holds the database register itself and it can also perform $\mathrm{U}^2_{\mathsf{qD}}$ efficiently. Indeed, algorithm $\mathcal{B}$ is an artificial algorithm designed only for proof, and there is no ambiguity in its definition.

Here $|\psi\rangle$ is the initial pure state on registers ZIOD_{q_H} and we suppose that (pk, sk, G, c^*, K_b^*) is encoded in this state without loss of generality. $\mathrm{U}_1, \ldots, \mathrm{U}_{q_D}$ are the unitary operations that act on registers ZIOD_{q_H} between oracle queries. Then we have

$$\begin{aligned}\Pr[1 \leftarrow \mathcal{B}^{\mathsf{qDeca}^*}(pk, sk, G)] &= \Pr[1 \leftarrow \mathcal{B}_U^{\mathsf{qDeca}^*}(pk, sk, G)],\\ \Pr[1 \leftarrow \mathcal{B}^{\mathsf{qDeca}^\diamond}(pk, sk, G)] &= \Pr[1 \leftarrow \mathcal{B}_U^{\mathsf{qDeca}^\diamond}(pk, sk, G)].\end{aligned} \tag{21}$$

By the analysis of Appendix D.2, for any unit joint state $|\Phi\rangle$ on registers ZIOD_{q_H} just before the application of $\mathrm{U}^1_{\mathsf{qD}}$ and $\mathrm{U}^2_{\mathsf{qD}}$, we have

$$\left\| \mathrm{U}^1_{\mathsf{qD}}|\Phi\rangle|0^m\rangle_{\mathsf{M}} - \mathrm{U}^2_{\mathsf{qD}}|\Phi\rangle|0^m\rangle_{\mathsf{M}} \right\| \leq 8 \cdot \sqrt{\gamma_{pk,sk}}.$$

By using the hybrid argument, it is straightforward to obtain

$$\| |\Psi_1\rangle|0^m\rangle_{\mathsf{M}} - |\Psi_2\rangle|0^m\rangle_{\mathsf{M}} \| \leq 8q_D \cdot \sqrt{\gamma_{pk,sk}}.$$

Then, by using the Lemma 4 of [1], we have

$$\left|\Pr\left[1 \leftarrow \mathcal{B}_U^{\mathsf{qDeca}^*}(pk, sk, G)\right] - \Pr\left[1 \leftarrow \mathcal{B}_U^{\mathsf{qDeca}^\diamond}(pk, sk, G)\right]\right| \leq 8q_D \cdot \sqrt{\gamma_{pk,sk}}.$$

By (21), we get

$$\left|\Pr\left[1 \leftarrow \mathcal{B}^{\mathsf{qDeca}^*}(pk, sk, G)\right] - \Pr\left[1 \leftarrow \mathcal{B}^{\mathsf{qDeca}^\diamond}(pk, sk, G)\right]\right| \leq 8q_D \cdot \sqrt{\gamma_{pk,sk}}.$$

Finally, combining above equation with (20) and averaging the (pk, sk, G), we obtain

$$|\Pr[1 \leftarrow \mathbf{G_1}] - \Pr[1 \leftarrow \mathbf{G_2}]| \leq 8q_D \cdot \sqrt{\underset{(pk,sk)\leftarrow\mathsf{Gen}}{\mathbb{E}}[\gamma_{pk,sk}]} \overset{(a)}{\leq} 8q_D \cdot 2^{-\gamma/2}.$$

Here (a) uses the fact that the PKE scheme P is weakly γ-spread. $\square$

D.2 Bound on $\left\| \mathbf{U}^{\mathbf{1}}_{\mathsf{qD}} |\boldsymbol{\Phi}\rangle |\mathbf{0}^{m}\rangle_{\mathsf{M}} - \mathbf{U}^{\mathbf{2}}_{\mathsf{qD}} |\boldsymbol{\Phi}\rangle |\mathbf{0}^{m}\rangle_{\mathsf{M}} \right\|$

Define set $\Gamma_{c,x} := \{y \in \{0,1\}^n : f_1(x,y) = c\}$, by the weakly γ-spread property of PKE scheme P, we have

$$\begin{aligned} \max_{c\in\mathcal{C},m\in\mathcal{M}} \frac{|\Gamma_{c,m}|}{2^n} &= \max_{c\in\mathcal{C},m\in\mathcal{M}} \frac{|\{y \in \{0,1\}^n : f_1(m,y) = c\}|}{2^n} \\ &\leq \max_{c\in\mathcal{C},m\in\mathcal{M}} \frac{|\{y \in \{0,1\}^n : \mathsf{Enc}_{pk}(m,y) = c\}|}{2^n} \leq \gamma_{pk,sk}. \end{aligned} \tag{22}$$

We rewrite the unit joint state $|\Phi\rangle$ on registers ZIOD_{q_H} just before the application of $\mathrm{U}^1_{\mathsf{qD}}$ and $\mathrm{U}^2_{\mathsf{qD}}$ as

$$|\Phi\rangle = \sum_{\substack{z\in\{0,1\}^*,c\in\mathcal{C} \\ y\in\{0.1\}^{n'+1},D\in\mathbf{D}_{q_H},n(D)<q_H}} \alpha_{z,c,y,D} |z\rangle_{\mathsf{Z}} |c,y\rangle_{\mathsf{IO}} |D\rangle_{\mathsf{D}_{q_H}}.$$

Here $n(D) < q_H$ because the RO-interface in algorithm $\mathcal{B}_U^{\mathsf{qDeca}^*}$ and $\mathcal{B}_U^{\mathsf{qDeca}^\diamond}$ is implemented at most q_H times. For the sake of convenience, we abbreviate $\substack{z\in\{0,1\}^*,c\in\mathcal{C} \\ y\in\{0.1\}^{n'+1},D\in\mathbf{D}_{q_H},n(D)<q_H}$ into $z,c,y,D,n(D) < q_H$ and $|z\rangle_{\mathsf{Z}}|c,y\rangle_{\mathsf{IO}}|D\rangle_{\mathsf{D}_{q_H}}$ into $|z,c,y,D\rangle$ in the following.

Next, we separate $|\Phi\rangle$ into four mutual orthogonal parts that

$$|\Phi\rangle = |\Phi_1\rangle + |\Phi_2\rangle + |\Phi_3\rangle + |\Phi_4\rangle,$$

where $|\Phi_1\rangle, |\Phi_2\rangle$, $|\Phi_3\rangle$ and $|\Phi_4\rangle$ are the following states:

$$\begin{aligned} |\Phi_1\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H \\ c=c^* \vee \mathsf{Dec}_{sk}(c)=\perp}} \beta_{z,c,y,D} |z,c,y,D\rangle, \\ |\Phi_2\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H \\ m:=\mathsf{Dec}_{sk}(c)\neq\perp \\ c\neq c^*,D(m)=\perp}} \beta_{z,c,y,D} |z,c,y,D\rangle, \\ |\Phi_3\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H-1 \\ m:=\mathsf{Dec}_{sk}(c)\neq\perp \\ c\neq c^*,D(m)=\perp \\ r\in\{0,1\}^n,r\neq 0^n}} \beta_{z,c,y,D,r} \frac{1}{\sqrt{2^n}} \sum_{y_1\in\{0,1\}^n} (-1)^{y_1\cdot r} |z,c,y,D\cup(m,y_1)\rangle, \\ |\Phi_4\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H-1 \\ m:=\mathsf{Dec}_{sk}(c)\neq\perp \\ c\neq c^*,D(m)=\perp}} \beta_{z,c,y,D,0^n} \frac{1}{\sqrt{2^n}} \sum_{y_1\in\{0,1\}^n} |z,c,y,D\cup(m,y_1)\rangle. \end{aligned}$$

For a fixed (z,c,y,D) with $c \neq c^*$, $m := \mathsf{Dec}_{sk}(c) \neq \perp$, $n(D) < q_H$ and $D(m) = \perp$, define states

$$\begin{aligned}
|\Upsilon_1[r,\nu]\rangle_{y,D}^{z,c} &:= \sum_{y_1 \in \Gamma_{c,m}} (-1)^{y_1 \cdot r} |z,c,y \oplus \nu, D \cup (m,y_1)\rangle, \\
|\Upsilon_2[r]\rangle_{y,D}^{z,c} &:= \sum_{y_1 \notin \Gamma_{c,m}} (-1)^{y_1 \cdot r} |z,c,y \oplus \perp, D \cup (m,y_1)\rangle, \\
|\Upsilon_3[r,\nu]\rangle_{y,D}^{z,c} &:= \sum_{y_1 \in \Gamma_{c,m}} (-1)^{y_1 \cdot r} |z,c,y \oplus \nu, D\rangle, \\
|\Upsilon_4[r,\nu]\rangle_{y,D}^{z,c} &:= \sum_{y_1 \in \Gamma_{c,m}} (-1)^{y_1 \cdot r} \sum_{y_2 \in \{0,1\}^n} \frac{1}{\sqrt{2^n}} |z,c,y \oplus \nu, D \cup (m,y_2)\rangle.
\end{aligned} \tag{23}$$

Here $r \in \{0,1\}^n$ and $\nu \in \{G(m), \perp\}$.

By the quantum circuit implementation of unitary operation U_m as shown in Appendix C and the definition of $\mathrm{U}_{\mathsf{qD}}^1$ and $\mathrm{U}_{\mathsf{qD}}^2$, we have[20]

$$\begin{aligned}
\mathrm{U}_{\mathsf{qD}}^1|\Phi_1\rangle|0^m\rangle = \mathrm{U}_{\mathsf{qD}}^2|\Phi_1\rangle|0^m\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H \\ c=c^* \vee \mathsf{Dec}_{sk}(c)=\perp}} \beta_{z,c,y,D}|z,c,y \oplus \perp, D\rangle|0^m\rangle, \\
\mathrm{U}_{\mathsf{qD}}^1|\Phi_2\rangle|0^m\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H \\ m:=\mathsf{Dec}_{sk}(c)\neq\perp \\ c\neq c^*, D(m)=\perp}} \frac{\beta_{z,c,y,D}}{\sqrt{2^n}} \left(\mathsf{S}_m|\Upsilon_1[0^n, G(m)]\rangle_{y,D}^{z,c} + \mathsf{S}_m|\Upsilon_2[0]\rangle_{y,D}^{z,c}\right)|0^m\rangle, \\
\mathrm{U}_{\mathsf{qD}}^2|\Phi_2\rangle|0^m\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H \\ m:=\mathsf{Dec}_{sk}(c)\neq\perp \\ c\neq c^*, D(m)=\perp}} \beta_{z,c,y,D}|z,c,y \oplus \perp, D\rangle|0^m\rangle, \\
\mathrm{U}_{\mathsf{qD}}^1|\Phi_3\rangle|0^m\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H-1 \\ m:=\mathsf{Dec}_{sk}(c)\neq\perp \\ c\neq c^*, D(m)=\perp \\ r\in\{0,1\}^n, r\neq 0^n}} \frac{\beta_{z,c,y,D,r}}{\sqrt{2^n}} \left(\mathsf{S}_m|\Upsilon_1[r, G(m)]\rangle_{y,D}^{z,c} + \mathsf{S}_m|\Upsilon_2[r]\rangle_{y,D}^{z,c}\right)|0^m\rangle, \\
\mathrm{U}_{\mathsf{qD}}^2|\Phi_3\rangle|0^m\rangle &= \sum_{\substack{z,c,y,D,n(D)<q_H-1 \\ m:=\mathsf{Dec}_{sk}(c)\neq\perp \\ c\neq c^*, D(m)=\perp \\ r\in\{0,1\}^n, r\neq 0^n}} \frac{\beta_{z,c,y,D,r}}{\sqrt{2^n}} \left(|\Upsilon_1[r, G(m)]\rangle_{y,D}^{z,c} + |\Upsilon_2[r]\rangle_{y,D}^{z,c}\right)|0^m\rangle.
\end{aligned} \tag{24}$$

As for the $\mathrm{U}_{\mathsf{qD}}^1|\Phi_4\rangle|0^m\rangle$ and $\mathrm{U}_{\mathsf{qD}}^2|\Phi_4\rangle|0^m\rangle$, we note that the state with the form of $\frac{1}{\sqrt{2^n}}\sum_{y_1\in\{0,1\}^n}|z,c,y,D\cup(m,y_1)\rangle$ cannot appear just before the application

[20] Here we omit the detailed computational process since the implementation of U_m is not very simple. Nevertheless, we stress that, following the implementation of U_m, one can get the state shown in (24) by direct computation.

of $\mathrm{U}^1_{\mathsf{qD}}$[21]. Hence we add a complement of the operation of $\mathrm{U}^1_{\mathsf{qD}}$ as

$$\mathrm{U}^1_{\mathsf{qD}}\frac{1}{\sqrt{2^n}}\sum_{y_1\in\{0,1\}^n}|z,c,y,D\cup(m,y_1)\rangle := \frac{1}{\sqrt{2^n}}\sum_{y_1\in\{0,1\}^n}|z,c,y\oplus\perp,D\cup(m,y_1)\rangle,$$

which is easily to implement since the state $\frac{1}{\sqrt{2^n}}\sum_{y_1\in\{0,1\}^n}|z,c,y,D\cup(m,y_1)\rangle$ must be orthogonal with $|\Phi_1\rangle$, $|\Phi_2\rangle$ and $|\Phi_3\rangle$. With this complement, we have

$$\mathrm{U}^1_{\mathsf{qD}}|\Phi_4\rangle|0^m\rangle = \sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp\\ c\neq c^*,D(m)=\perp}}\frac{\beta_{z,c,y,D,0^n}}{\sqrt{2^n}}\sum_{y_1\in\{0,1\}^n}|z,c,y\oplus\perp,D\cup(m,y_1)\rangle|0^m\rangle,$$

$$\mathrm{U}^2_{\mathsf{qD}}|\Phi_4\rangle|0^m\rangle = \sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp\\ c\neq c^*,D(m)=\perp}}\frac{\beta_{z,c,y,D,0^n}}{\sqrt{2^n}}\left(|\Upsilon_1[0^n,G(m)]\rangle^{z,c}_{y,D}+|\Upsilon_2[0^n]\rangle^{z,c}_{y,D}\right)|0^m\rangle.$$

Then we can obtain $\mathrm{U}^1_{\mathsf{qD}}|\Phi_1\rangle-\mathrm{U}^2_{\mathsf{qD}}|\Phi_1\rangle=\mathbf{0}$ and

$$(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi_2\rangle|0^m\rangle = \sum_{\substack{z,c,y,D,n(D)<q_H\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp\\ c\neq c^*,D(m)=\perp}}\frac{\beta_{z,c,y,D}}{\sqrt{2^n}}\mathsf{S}_m\left(|\Upsilon_1[0^n,G(m)]\rangle^{z,c}_{y,D}-|\Upsilon_1[0^n,\perp]\rangle^{z,c}_{y,D}\right)|0^m\rangle,$$

$$(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi_3\rangle|0^m\rangle = \sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp\\ c\neq c^*,D(m)=\perp\\ r\in\{0,1\}^n,r\neq 0^n}}\frac{\beta_{z,c,y,D,r}}{2^n}\begin{pmatrix}|\Upsilon_3[r,G(m)]\rangle^{z,c}_{y,D}-|\Upsilon_3[r,\perp]\rangle^{z,c}_{y,D}\\ +|\Upsilon_4[r,\perp]\rangle^{z,c}_{y,D}-|\Upsilon_4[r,G(m)]\rangle^{z,c}_{y,D}\end{pmatrix}|0^m\rangle,$$

$$(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi_4\rangle|0^m\rangle = \sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp\\ c\neq c^*,D(m)=\perp}}\frac{\beta_{z,c,y,D,0^n}}{\sqrt{2^n}}\left(|\Upsilon_1[0^n,\perp]\rangle^{z,c}_{y,D}-|\Upsilon_1[0^n,G(m)]\rangle^{z,c}_{y,D}\right)|0^m\rangle.$$

[21] Roughly speaking, this property can be obtained from the definition of S_m (Section 2.3), thus it always transforms the uniform superposition $|D\cup(x,\hat{0^n})\rangle$ into $|D\rangle$. This property is also used in the proof of Lemma 5 of [29]. However, the state with that form can appear just before the application of $\mathrm{U}^2_{\mathsf{qD}}$, since $\mathrm{U}^2_{\mathsf{qD}}$ uses the extraction-interface $\mathsf{eCO.E}_{f_1}$.

Therefore, we have

$$
\begin{aligned}
&\left\|(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi_2\rangle|0^m\rangle\right\|^2\\
&=\left\|\sum_{\substack{z,c,y,D,n(D)<q_H\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\frac{\beta_{z,c,y,D}}{\sqrt{2^n}}\left(\mathsf{S}_m|\Upsilon_1[0^n,G(m)]\rangle^{z,c}_{y,D}-\mathsf{S}_m|\Upsilon_1[0^n,\perp]\rangle^{z,c}_{y,D}\right)\right\|^2\\
&\overset{(a)}{=}\left\|\sum_{\substack{z,c,y,D,n(D)<q_H\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\frac{\beta_{z,c,y,D}}{\sqrt{2^n}}\left(|\Upsilon_1[0^n,G(m)]\rangle^{z,c}_{y,D}-|\Upsilon_1[0^n,\perp]\rangle^{z,c}_{y,D}\right)\right\|^2\\
&\overset{(b)}{\leq}2\left\|\sum_{\substack{z,c,y,D,n(D)<q_H\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\frac{\beta_{z,c,y,D}}{\sqrt{2^n}}|\Upsilon_1[0^n,G(m)]\rangle^{z,c}_{y,D}\right\|^2+2\left\|\sum_{\substack{z,c,y,D,n(D)<q_H\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\frac{\beta_{z,c,y,D}}{\sqrt{2^n}}|\Upsilon_1[0^n,\perp]\rangle^{z,c}_{y,D}\right\|^2\\
&\overset{(c)}{=}4\sum_{\substack{z,c,y,D,n(D)<q_H\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\frac{|\mathit{\Gamma}_{c,m}|}{2^n}\left|\beta_{z,c,y,D}\right|^2\leq 4\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\mathit{\Gamma}_{c,m}|}{2^n}\sum_{\substack{z,c,y,D,n(D)<q_H\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\left|\beta_{z,c,y,D}\right|^2\\
&=4\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\mathit{\Gamma}_{c,m}|}{2^n}\cdot\|\Phi_2\|^2.
\end{aligned}
\tag{25}
$$

Here (a) uses the fact that S_m is a unitary operation, (b) uses Corollary 2, and (c) uses the definition of state $|\Upsilon_1[r,\nu]\rangle^{z,c}_{y,D}$ in (23).

Similar with the computation of $\|(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi_2\rangle|0^m\rangle\|^2$, we can also obtain

$$
\left\|(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi_4\rangle|0^m\rangle\right\|^2\leq 4\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\mathit{\Gamma}_{c,m}|}{2^n}\cdot\|\Phi_4\|^2. \tag{26}
$$

As for the $\|(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^3_{\mathsf{qD}})|\Phi_3\rangle|0^m\rangle\|^2$, we first compute

$$
\begin{aligned}
\||\Phi_3\rangle\|^2&=\left\|\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{\substack{c\neq c^*,D(m)=\perp\\ r\in\{0,1\}^n,r\neq 0^n}}\beta_{z,c,y,D,r}\frac{1}{\sqrt{2^n}}\sum_{y_1\in\{0,1\}^n}(-1)^{y_1\cdot r}|z,c,y,D\cup(m,y_1)\rangle\right\|^2\\
&=\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\sum_{y_1\in\{0,1\}^n}\left\|\sum_{r\in\{0,1\}^n,r\neq 0^n}\beta_{z,c,y,D,r}\frac{(-1)^{y_1\cdot r}}{\sqrt{2^n}}|z,c,y,D\cup(m,y_1)\rangle\right\|^2\\
&=\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\sum_{y_1\in\{0,1\}^n}\left|\sum_{r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y_1\cdot r}}{\sqrt{2^n}}\beta_{z,c,y,D,r}\right|^2.
\end{aligned}
\tag{27}
$$

Then we have

$$
\begin{aligned}
&\left\|(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi_3\rangle|0^m\rangle\right\|^2 = \left\|\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{\substack{c\neq c^*,D(m)=\perp\\ r\in\{0,1\}^n,r\neq 0^n}} \frac{\beta_{z,c,y,D,r}}{2^n}\begin{pmatrix}|\Upsilon_3[r,G(m)]\rangle^{z,c}_{y,D}-|\Upsilon_3[r,\perp]\rangle^{z,c}_{y,D}\\ +|\Upsilon_4[r,\perp]\rangle^{z,c}_{y,D}-|\Upsilon_4[r,G(m)]\rangle^{z,c}_{y,D}\end{pmatrix}\right\|^2\\
&\overset{(d)}{\leq} 4\left\|\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{\substack{c\neq c^*,D(m)=\perp\\ r\in\{0,1\}^n,r\neq 0^n}} \frac{\beta_{z,c,y,D,r}}{2^n}|\Upsilon_3[r,G(m)]\rangle^{z,c}_{y,D}\right\|^2 + 4\left\|\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{\substack{c\neq c^*,D(m)=\perp\\ r\in\{0,1\}^n,r\neq 0^n}} \frac{\beta_{z,c,y,D,r}}{2^n}|\Upsilon_3[r,\perp]\rangle^{z,c}_{y,D}\right\|^2\\
&\quad + 4\left\|\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{\substack{c\neq c^*,D(m)=\perp\\ r\in\{0,1\}^n,r\neq 0^n}} \frac{\beta_{z,c,y,D,r}}{2^n}|\Upsilon_4[r,\perp]\rangle^{z,c}_{y,D}\right\|^2 + 4\left\|\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{\substack{c\neq c^*,D(m)=\perp\\ r\in\{0,1\}^n,r\neq 0^n}} \frac{\beta_{z,c,y,D,r}}{2^n}|\Upsilon_4[r,G(m)]\rangle^{z,c}_{y,D}\right\|^2\\
&\overset{(e)}{=} 16\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\left|\sum_{y_1\in\Gamma_{c,m}}^{r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y_1\cdot r}}{2^n}\beta_{z,c,y,D,r}\right|^2\\
&\overset{(f)}{\leq} 16\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\sum_{y_1\in\Gamma_{c,m}}\frac{|\Gamma_{c,m}|}{2^n}\left|\sum_{r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y_1\cdot r}}{\sqrt{2^n}}\beta_{z,c,y,D,r}\right|^2\\
&\leq 16\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\Gamma_{c,m}|}{2^n}\sum_{\substack{z,c,y,D,n(D)<q_H-1\\ m:=\mathsf{Dec}_{sk}(c)\neq\perp}}^{c\neq c^*,D(m)=\perp}\sum_{y_1\in\Gamma_{c,m}}\left|\sum_{r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y_1\cdot r}}{\sqrt{2^n}}\beta_{z,c,y,D,r}\right|^2\\
&\overset{(g)}{\leq} 16\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\Gamma_{c,m}|}{2^n}\cdot\||\Phi_3\rangle\|^2.
\end{aligned}
\tag{28}
$$

Here (d) uses Corollary 2 again, (e) uses the definition of state $|\Upsilon_3[r,\nu]\rangle^{z,c}_{y,D}$ and $|\Upsilon_4[r,\nu]\rangle^{z,c}_{y,D}$ in (23), (f) uses the Cauchy-Schwarz inequality, (g) uses (27).

Combining (22), (25), (26) and (28), we finally obtain

$$
\begin{aligned}
&\left\|(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi\rangle|0^m\rangle\right\| \overset{(h)}{\leq} \sum_{i=0}^{4}\left\|(\mathrm{U}^1_{\mathsf{qD}}-\mathrm{U}^2_{\mathsf{qD}})|\Phi_i\rangle|0^m\rangle\right\|\\
&\leq \sqrt{4\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\Gamma_{c,m}|}{2^n}}\cdot\||\Phi_2\rangle\| + \sqrt{16\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\Gamma_{c,m}|}{2^n}}\cdot\||\Phi_3\rangle\| + \sqrt{4\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\Gamma_{c,m}|}{2^n}}\cdot\||\Phi_4\rangle\|\\
&\overset{(i)}{\leq} \sqrt{4\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\Gamma_{c,m}|}{2^n}} + \sqrt{16\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\Gamma_{c,m}|}{2^n}} + \sqrt{4\max_{c\in\mathcal{C},m\in\mathcal{M}}\frac{|\Gamma_{c,m}|}{2^n}} \leq 8\cdot\sqrt{\gamma_{pk,sk}}.
\end{aligned}
$$

Here (h) uses triangle inequality, (i) uses the fact that $\||\Phi_i\rangle\|\leq 1$ $(i=1,\ldots,4)$.

D.3 Proof of Lemma 6

Proof. We first introduce two new games as follows:

Game $\mathbf{G_{2a}}$: This game is identical to game $\mathbf{G_2}$, except that the compressed semi-classical oracle $\mathcal{O}^{CSC}_{R^D_{pk,sk}}$ is queried just after each querying of the RO-interface $\mathsf{eCO.RO}$.

Game $\mathbf{G_{3a}}$: This game is identical to game $\mathbf{G_3}$, except that the compressed semi-classical oracle $\mathcal{O}^{CSC}_{R^D_{pk,sk}}$ is queried just after each querying of the RO-interface $\mathsf{eCO.RO}$.

In game $\mathbf{G_2}$, the RO-interface $\mathsf{eCO.RO}$ of the extractable RO-simulator $\mathsf{S}(f_1)$ is used to simulate the quantum random oracle H. Since the RO-interface $\mathsf{eCO.RO}$ is implemented by the unitary operation CStO, the quantum random oracle H in game $\mathbf{G_2}$ is actually implemented by the compressed standard oracle.

In game $\mathbf{G_2}$, the extraction-interface $\mathsf{eCO.E}_{f_1}$ of the extractable RO-simulator $\mathsf{S}(f_1)$ is used to simulate the decapsulation oracle $\mathsf{qDeca}^{\diamond}$. As explained in Section 2.4, for any fixed function f, the extraction-interface $\mathsf{eCO.E}_f$ is processed by a database read operation Ext_f.

Now, we construct a quantum oracle algorithm $\mathcal{B}^{H,\mathsf{eCO.E}_{f_1}}(pk, sk)$ that executes game $\mathbf{G_2}$, this algorithm makes at most q_H queries to quantum random oracle H. Then,

$$\Pr[1 \leftarrow \mathbf{G_2}] = \Pr[1 \leftarrow \mathcal{B}^{H,\mathsf{eCO.E}_{f_1}}(pk, sk) : (R^D_{pk,sk}, pk, sk) \leftarrow \mathcal{D}].$$

Here $\mathcal{D}$ is a joint distribution that $(pk, sk) \leftarrow \mathsf{Gen}$, and set $R^D_{pk,sk}$ defined in (8) is determined by (pk, sk). Correspondingly, we have

$$\Pr[1 \leftarrow \mathbf{G_{2a}}] = \Pr[1 \leftarrow \mathcal{B}^{H\backslash R^D_{pk,sk},\mathsf{eCO.E}_{f_1}}(pk, sk) : (R^D_{pk,sk}, pk, sk) \leftarrow \mathcal{D}],$$
$$\Pr[1 \leftarrow \mathbf{G_3}] = \Pr[1 \leftarrow \mathcal{B}^{H,\mathsf{eCO.E}_{f_2}}(pk, sk) : (R^D_{pk,sk}, pk, sk) \leftarrow \mathcal{D}],$$
$$\Pr[1 \leftarrow \mathbf{G_{3a}}] = \Pr[1 \leftarrow \mathcal{B}^{H\backslash R^D_{pk,sk},\mathsf{eCO.E}_{f_2}}(pk, sk) : (R^D_{pk,sk}, pk, sk) \leftarrow \mathcal{D}].$$

Thus, by using Theorem 1, we have

$$|\Pr[1 \leftarrow \mathbf{G_2}] - \Pr[1 \leftarrow \mathbf{G_{2a}}]| \leq \sqrt{q_H(q_H+1) \cdot \mathop{\mathbb{E}}_{(R^D_{pk,sk},pk,sk)\leftarrow\mathcal{D}} \left\| \left[\mathrm{J}_{R^D_{pk,sk}}, \mathsf{CStO} \right] \right\|^2}, \tag{29}$$

and

$$|\Pr[1 \leftarrow \mathbf{G_3}] - \Pr[1 \leftarrow \mathbf{G_{3a}}]| \leq \sqrt{q_H(q_H+1) \cdot \mathop{\mathbb{E}}_{(R^D_{pk,sk},pk,sk)\leftarrow\mathcal{D}} \left\| \left[\mathrm{J}_{R^D_{pk,sk}}, \mathsf{CStO} \right] \right\|^2}. \tag{30}$$

By the analysis just before Lemma 6 in the proof of Theorem 2, we know that the extraction-interfaces $\mathsf{eCO.E}_{f_1}$ and $\mathsf{eCO.E}_{f_2}$ proceed identically for any input state $|c,D,m\rangle_{\mathsf{ID}_{q_H}\mathsf{M}}$ if $D\notin R^D_{pk,sk}$. Therefore, algorithms $\mathcal{B}^{H\backslash R^D_{pk,sk},\mathsf{eCO.E}_{f_1}}(pk,sk)$ and $\mathcal{B}^{H\backslash R^D_{pk,sk},\mathsf{eCO.E}_{f_2}}(pk,sk)$ proceed identically if the compressed semi-classical oracle $\mathcal{O}^{CSC}_{R^D_{pk,sk}}$ never returns 1. This implies that

$$\begin{aligned}&\Pr[\mathsf{Find}\text{ occurs in }\mathcal{B}^{H\backslash R^D_{pk,sk},\mathsf{eCO.E}_{f_1}}(pk,sk):(R^D_{pk,sk},pk,sk)\leftarrow\mathcal{D}]\\&\qquad=\Pr[\mathsf{Find}\text{ occurs in }\mathcal{B}^{H\backslash R^D_{pk,sk},\mathsf{eCO.E}_{f_2}}(pk,sk):(R^D_{pk,sk},pk,sk)\leftarrow\mathcal{D}],\end{aligned}$$

then by the difference lemma of [25], we have

$$\begin{aligned}&|\Pr[1\leftarrow\mathbf{G_{2a}}]-\Pr[1\leftarrow\mathbf{G_{3a}}]|\\&\leq\Pr[\mathsf{Find}\text{ occurs in }\mathcal{B}^{H\backslash R^D_{pk,sk},\mathsf{eCO.E}_{f_2}}(pk,sk):(R^D_{pk,sk},pk,sk)\leftarrow\mathcal{D}]\\&\overset{(a)}{\leq}q_H\cdot\mathop{\mathbb{E}}_{(R^D_{pk,sk},pk,sk)\leftarrow\mathcal{D}}\left\|\left[\mathrm{J}_{R^D_{pk,sk}},\mathsf{CStO}\right]\right\|^2.\end{aligned}\tag{31}$$

Here (a) uses Theorem 1 again.

Combining (29), (30) and (31), we obtain

$$\begin{aligned}|\Pr[1\leftarrow\mathbf{G_2^q}]-\Pr[1\leftarrow\mathbf{G_3^q}]|\leq&\sqrt{q_H(q_H+1)\cdot\mathop{\mathbb{E}}_{(R^D_{pk,sk},pk,sk)\leftarrow\mathcal{D}}\left\|\left[\mathrm{J}_{R^D_{pk,sk}},\mathsf{CStO}\right]\right\|^2}\\&+q_H\cdot\mathop{\mathbb{E}}_{(R^D_{pk,sk},pk,sk)\leftarrow\mathcal{D}}\left\|\left[\mathrm{J}_{R^D_{pk,sk}},\mathsf{CStO}\right]\right\|^2.\end{aligned}\tag{32}$$

Define function $g:\{0,1\}^m\times\{0,1\}^n\to\{0,1\}$ as

$$g(x,y)=\begin{cases}1\text{ if }\mathsf{Enc}(pk,x,y)=c\wedge\mathsf{Dec}(sk,c)\neq x\\0\text{ otherwise.}\end{cases}$$

The relation R_1^g and the corresponding parameter $\mathit{\Gamma}_{R_1^g}$ defined in Section 2.4 can be written as

$$R_1^g:=\{(x,y)\in\{0,1\}^m\times\{0,1\}^n|g(x,y)=1\},$$

$$\mathit{\Gamma}_{R_1^g}:=\max_{x\in\{0,1\}^m}|\{y\in\{0,1\}^n|\mathsf{Enc}(pk,x,y)=c\wedge\mathsf{Dec}(sk,c)\neq x\}|\overset{(b)}{\leq}2^n\delta_{pk,sk}.\tag{33}$$

Here (b) uses the fact that the underlying PKE scheme P is δ-correct.

For the relation R_1^g, define following projectors on database register D_{q_H}:

$$\mathit{\Sigma}^x:=\sum_{\substack{D\ s.t.\ (x,D(x))\in R_1^g\\x'<x,(x',D(x'))\notin R_1^g}}|D\rangle\langle D|\ (x\in\{0,1\}^m),\quad\mathit{\Sigma}^\perp:=\mathbf{I}-\sum_{x\in\{0,1\}^m}\mathit{\Sigma}^x.$$

By the definition of set $R^D_{pk,sk}$ defined in (8), it is obvious that $\mathrm{J}_{R^D_{pk,sk}}=\sum_{x\in\{0,1\}^m}\mathit{\Sigma}^x$, thus $\mathit{\Sigma}^\perp=\mathbf{I}-\mathrm{J}_{R^D_{pk,sk}}$. Hence we have

$$\left\|\left[\mathrm{J}_{R^D_{pk,sk}},\mathsf{CStO}\right]\right\|\overset{(c)}{=}\left\|\left[\mathbf{I}-\mathrm{J}_{R^D_{pk,sk}},\mathsf{CStO}\right]\right\|=\|[\mathit{\Sigma}^\perp,\mathsf{CStO}]\|\overset{(d)}{\leq}8\cdot\sqrt{\mathit{\Gamma}_{R_1^g}/2^n}.\tag{34}$$

Here (c) uses the basic property of the commutator, (d) uses Lemma 4.

Combining (32), (33) and (34), we finally obtain

$$|\Pr[1 \leftarrow \mathbf{G_2^q}] - \Pr[1 \leftarrow \mathbf{G_3^q}]| \leq 8 \cdot \sqrt{q_H(q_H+1) \cdot \delta} + 64 q_H \cdot \delta.$$

$\square$