A preliminary version of this paper appears in Advances in Cryptology - ASIACRYPT 2024.

Tighter Proofs for PKE-to-KEM Transformation in the Quantum Random Oracle Model

Jinrong Chen [*] jinrongchen@nudt.edu.cn

Yi Wang [†] wangyi14@nudt.edu.cn

Rongmao Chen [(✉)] [‡] chromao@nudt.edu.cn

Xinyi Huang [§] xyhuang81@gmail.com

Wei Peng [¶] wpeng@nudt.edu.cn

October 8, 2024

Abstract

In this work, we provide new, tighter proofs for the T_{RH}-transformation by Jiang et al. (ASIACRYPT 2023), which converts OW-CPA secure PKEs into KEMs with IND-1CCA security, a variant of typical IND-CCA security where only a single decapsulation query is allowed. Such KEMs are efficient and have been shown sufficient for real-world applications by Huguenin-Dumittan and Vaudenay at EUROCRYPT 2022. We reprove Jiang et al.'s T_{RH}-transformation in both the random oracle model (ROM) and the quantum random oracle model (QROM), for the case where the underlying PKE is rigid deterministic. In both ROM and QROM models, our reductions achieve security loss factors of $\mathcal{O}(1)$, significantly improving Jiang et al.'s results which have security loss factors of $\mathcal{O}(q)$ in the ROM and $\mathcal{O}(q^2)$ in the QROM respectively. Notably, central to our tight QROM reduction is a new tool called "reprogram-after-measure", which overcomes the reduction loss posed by oracle reprogramming in QROM proofs. This technique may be of independent interest and useful for achieving tight QROM proofs for other post-quantum cryptographic schemes. We remark that our results also improve the reduction tightness of the T_H-transformation (which also converts PKEs to KEMs) by Huguenin-Dumittan and Vaudenay (EUROCRYPT 2022), as Jiang et al. provided a tight reduction from T_H-transformation to T_{RH}-transformation (ASIACRYPT 2023).

Keywords: QROM · Security proof · Tight reduction · 1CCA security · KEM.

[*]National University of Defense Technology

[†]National University of Defense Technology

[‡]National University of Defense Technology

[§]Jinan University

[¶]National University of Defense Technology

Contents

1 Introduction **2**
1.1 Our Contributions . . . 2
1.2 Results Overview . . . 3
1.3 Discussions . . . 5
1.4 Related Work . . . 6

2 Preliminaries **7**
2.1 Notation . . . 7
2.2 The (Quantum) Random Oracle Model . . . 8
2.3 The One-Way to Hiding Lemma . . . 8
2.4 The Compressed Oracle Technique . . . 8
2.5 Cryptographic Primitives . . . 9

3 The Security of T_{RH} in the ROM **11**

4 The Security Analysis in the QROM **14**
4.1 The Reprogram-after-Measure Technique . . . 14
4.2 The Security of T_{RH} in the QROM . . . 20

1 Introduction

Indistinguishability against Chosen-Ciphertext Attacks (IND-CCA) has been widely considered as the security standard for post-quantum key encapsulation mechanisms (KEMs) [10, 20, 34, 35, 36, 37, 40, 47], which could be achieved by applying the Fujisaki-Okamoto-like (FO-like) transformation [27] to public-key encryption (PKE) with security weaker than IND-CCA. However, in the post-quantum cryptography (PQC) migration, it has been shown that IND-1CCA-secure KEM is sufficient to replace the Diffie-Hellman key exchange in TLS 1.3 [21] and Signal [14] to achieve post-quantum security [31]. Compared to IND-CCA security, IND-1CCA security allows the adversary to make only a single decapsulation query. This restriction enables more efficient transformations [31, 33] than the FO-like approach, as it removes the need for the time-consuming re-encryption operation in the decapsulation algorithm. In particular, Huguenin-Dumittan and Vaudenay [31] pointed out that omitting the re-encryption step could speed up the decapsulation algorithm of Kyber [13] and Frodo-AES [2] by 2.17 times and 6.11 times, respectively. Besides, removing the re-encryption operation might enhance the security of the obtained KEM against side-channel attacks [49].

To design IND-1CCA-secure KEMs, Huguenin-Dumittan and Vaudenay [31] proposed two transformations called T_{CH} and T_H, both of which build KEMs from PKE schemes with One-Wayness against Chosen-Plainxt Attacks (OW-CPA). In particular, T_{CH} is a variant of the REACT transformation [43], and T_H is the same as the $U^{\perp}$ transformation in [27]. Later, Jiang et al. [33] presented an implicit variant of T_H called T_{RH} where the decapsulation algorithm returns a pseudo-random value instead of an explicit abort symbol for an invalid ciphertext. Also, they provided tighter proofs for T_H by reducing its IND-1CCA security to the IND-1CCA security of T_{RH}.

Table 1 lists the reduction tightness of these transformations with deterministic PKE in the random oracle model (ROM) [7] and the quantum random oracle model (QROM) [11]. Hereafter, we will use T_X^{D} to denote T_X with deterministic PKE for $X \in \{CH, H, RH\}$. As shown in Table 1, the ROM proof of T_{CH}^{D} is almost tight, but the QROM proof of T_{CH}^{D} requires an additional hash function for ciphertext verification which increases the size of ciphertext. In contrast, the QROM proofs of T_H^{D} and T_{RH}^{D} in [33] do not need ciphertext expansion.

Jiang et al. [33] not only made improvements on the reduction tightness of T_H, but also proved that the reduction losses $\mathcal{O}(q)$ and $\mathcal{O}\big(q^2\big)$ are unavoidable in the ROM and QROM proofs of T_{RH} respectively. However, in this work we found that these reduction losses could be further reduced to $\mathcal{O}(1)$ when the underlying PKEs are rigid deterministic (See Section 1.2 for detailed explanation). These tight security reductions could improve the practical efficiency of KEMs built via the T_{RH}^{D} due to no need to increase the security parameter to compensate for the loss factor.

1.1 Our Contributions

In this work, we provide new, tighter proofs for the T_{RH}-transformation by Jiang et al. [33] when the underlying PKE is rigid deterministic[1], as shown in Table 1, and our contributions are as follows.

First, we present a tight security proof with loss factor $\mathcal{O}(1)$ for T_{RH}^{D} in the ROM (Theorem 3.1). In this proof, we propose a new strategy to simulate the decapsulation oracle successfully with probability 1/2. This strategy takes full advantage of the rigid deterministic property of PKE, and does not have to guess the random oracle query of adversary.

[1]The property of "rigidity" is studied by Bernstein and Persichetti [8]. Roughly speaking, it means that $\mathsf{Enc}(\mathsf{pk}, m) = c$ for every $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda), c \in \mathcal{C}$, and $m := \mathsf{Dec}(\mathsf{sk}, c)$.

Table 1: The reduction tightness of transformations from OW-CPA-secure deterministic PKE to IND-1CCA-secure KEM in the ROM/QROM. Here ϵ_R represents the advantage of the reduction algorithm R with respect to the OW-CPA security of the underlying PKE scheme, $\epsilon_{\mathcal{A}}$ represents the advantage of the adversary $\mathcal{A}$ with respect to the IND-1CCA security of the obtained KEM scheme, and q is the total number of random oracle queries made by $\mathcal{A}$.

Model	Transformation	Reduction tightness
ROM	T_{CH}^{D} [31]	$\epsilon_R \approx \epsilon_{\mathcal{A}}$ [31]
	T_{H}^{D} [31]	$\epsilon_R \approx \mathcal{O}\big(1/q^2\big)\epsilon_{\mathcal{A}}$ [31] $\epsilon_R \approx \mathcal{O}(1/q)\epsilon_{\mathcal{A}}$ [33]
	T_{RH}^{D} [33]	$\epsilon_R \approx \mathcal{O}(1/q)\epsilon_{\mathcal{A}}$ [33] $\epsilon_R \approx \epsilon_{\mathcal{A}}$ (Our work)
QROM	T_{CH}^{D} [31]	$\epsilon_R \approx \mathcal{O}\big(1/q^3\big)\epsilon_{\mathcal{A}}^2$ [31]
	T_{H}^{D} [31]	$\epsilon_R \approx \mathcal{O}\big(1/q^2\big)\epsilon_{\mathcal{A}}^2$ [33]
	T_{RH}^{D} [33]	$\epsilon_R \approx \mathcal{O}\big(1/q^2\big)\epsilon_{\mathcal{A}}^2$ [33] $\epsilon_R \approx \epsilon_{\mathcal{A}}^2$ (Our work)

$\mathsf{Gen}(1^\lambda)$	$\mathsf{Encaps}(\mathsf{pk})$	$\mathsf{Decaps}(\mathsf{sk}, c)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}'(1^\lambda)$ **return** $(\mathsf{pk}, \mathsf{sk})$	$m \leftarrow\$ \mathcal{M}$ $c \leftarrow \mathsf{Enc}'(\mathsf{pk}, m)$ $k := H(m, c)$ **return** (k, c)	$m' := \mathsf{Dec}'(\mathsf{sk}, c)$ **if** $m' = \perp$ **then** **return** $k' := H(\star, c)$ **return** $k' := H(m', c)$

Figure 1: $\mathsf{KEM}_{RH} := T_{RH}[\mathsf{PKE}', H]$.

Second, we extend the above strategy to the QROM and obtain a tight security proof with loss factor $\mathcal{O}(1)$ for T_{RH}^{D} in the QROM (Theorem 4.2). At the core of our QROM proof for T_{RH}^{D} is a novel technique called *reprogram-after-measure*, which is used to handle the issue of random oracle reprogramming in the QROM.

Compared with existing techniques including one-way to hiding (O2H) [3, 10, 40, 51] and measure-and-reprogram [18, 19], our technique is tailored for this particular case and introduces a reduction loss of $\mathcal{O}(1)$ only. Note that our results also improve the reduction tightness of T_{H}^{D} [31], as Jiang et al. [33] provided a tight reduction from T_H to T_{RH}.

1.2 Results Overview

T_{RH} transformation is shown in Fig. 1, where $\mathcal{M}$ and $\mathcal{C}$ are the message space and the ciphertext space of the underlying PKE scheme $\mathsf{PKE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$, respectively, $\mathcal{K}$ is the key space of KEM_{RH}, $\star$ is a fixed public value, and H is a hash function mapping from $\mathcal{M} \cup \{\star\} \times \mathcal{C}$ to $\mathcal{K}$. For simplicity, we only consider the case of $\star \in \mathcal{M}$, and the case of $\star \notin \mathcal{M}$ can be proved similarly.

On the Reduction Tightness of T_{RH} by Jiang et al. [33]. Theorem 5.1 in [33] says that the reduction loss factors $\mathcal{O}(q)$ and $\mathcal{O}(q^2)$ are unavoidable in the ROM and QROM proofs of T_{RH} when the underlying PKE is malleable. The proof of this theorem describes a ROM/QROM adversary $\mathcal{B}$ against the IND-1CCA security of T_{RH}. In specific, given $c^* \leftarrow \mathsf{Enc}'(\mathsf{pk}, m^*)$ and k^*, $\mathcal{B}$ needs to determine whether $k^* = H(m^*, c^*)$ or k^* is a random value over $\mathcal{K}$. By the malleability of PKE', $\mathcal{B}$ first derives a new c' from c^* where $c' = \mathsf{Enc}'(\mathsf{pk}, f(m^*))$ and f is the function associated to the malleability of PKE'. Then, $\mathcal{B}$ makes the single decapsulation query on c' and receives $\mathsf{tag} = H(f(m^*), c')$. Now $\mathcal{B}$ makes random oracle queries to find $m^* \in \mathcal{M}$ such that $H(f(m^*), c') = \mathsf{tag}$, and computes $H(m^*, c^*)$ to check whether k^* is random or not. Let q be the total number of random oracle queries made by $\mathcal{B}$, Jiang et al. [33] pointed out that these q random oracle queries contribute to unavoidable loss factors of $\mathcal{O}(q)$ and $\mathcal{O}(q^2)$ in the ROM and QROM.

Note that if $\mathsf{Enc}'(\mathsf{pk}, \cdot)$ is rigid deterministic, $\mathcal{B}$ could find correct m^* by computing $\mathsf{Enc}'(\mathsf{pk}, \cdot)$ and comparing with c^* instead of querying random oracle, and the loss factors in the ROM and QROM could be avoided. This fact implies that it might be possible to improve the reduction tightness of T_{RH}^{D} by Jiang et al. [33].

Our Result I: Tight ROM Proof of T_{RH}^{D}. As pointed out by Jiang et al. [33], the core of the ROM proof is simulating decapsulation oracle without sk. The simulation of hash function H relies on a hash list to record all the random oracle queries and corresponding hash values. The ROM proof of T_{RH}^{D} in [33] is based on the fact that the decapsulation oracle always makes a random oracle query to generate k' and one could find the corresponding query from the hash list of H, say the i^*-th entry, where $i^* \in \{0, \ldots, q\}$ and q is the total number of random oracle queries made by $\mathcal{A}$. So, the simulator of decapsulation oracle first randomly selects $i^* \in \{0, \ldots, q\}$. If the i^*-th entry is not empty when $\mathcal{A}$ queries the decapsulation oracle, it returns the hash value of this entry; otherwise, it returns a random $k^* \in \mathcal{K}$ and when $\mathcal{A}$ makes the i^*-th hash query, k^* is returned. Therefore, the probability of a successful simulation is $1/(q+1)$.

To achieve tighter proof, we present a new simulation strategy. That is, determining the way to compute k' in decapsulation oracle based on a correct guess on $m' \neq \perp$ with probability $1/2$. In the case of a correct guess on $m' \neq \perp$, assuming PKE' is perfectly correct, the simulator of decapsulation oracle first checks whether there is a pair (m', c) in the hash list such that $\mathsf{Enc}'(\mathsf{pk}, m') = c$:

- If such a pair exists, then $k' := H(m', c)$. The perfect correctness and the rigidity of the deterministic PKE' guarantee that if $\mathsf{Enc}'(\mathsf{pk}, m') = c$, then $\mathsf{Dec}'(\mathsf{sk}, c) = m'$.
- Otherwise, $\mathcal{A}$ does not have any knowledge of $H(m', c)$. Responding with $k' := k^*$, where $k^* \in \mathcal{K}$ is a random value, implicitly assigns k^* to $H(m', c)$ and would not be noticed by $\mathcal{A}$. After this, if $\mathcal{A}$ makes a random oracle query on this pair, the random oracle should return k^*.

This completes the simulation of the decapsulation oracle without the knowledge of sk. The probability of a successful simulation is $1/2$, and the loss factor of our proof is 2. Note that if PKE' is δ-correct where $\delta \neq 0$, i.e., PKE' is not perfectly correct, then there will be an error term δ in our reduction result.

Our Result II: Tight QROM Proof of T_{RH}^{D}. Note that, in the QROM, since $\mathcal{A}$ can make the random oracle queries in superposition, there is no such a hash list that can copy down $\mathcal{A}$'s queries and their responses, which implies that we cannot implicitly reprogram $H(m', c)$ to the random k^* as above. So, we propose following technique to fix this issue.

A New Tool: Reprogram-after-Measure. We present a simulator that can use a random value to simulate the decapsulation oracle without the knowledge of sk. This simulator simulates the random oracle using Zhandry's compressed oracle technique [55], which can record information about the adversary's quantum queries into a database in superposition without being detected by adversary. Assuming PKE' is perfectly correct and rigid deterministic, we can still use the simulation strategy in the ROM, i.e., guessing whether m' is equal to $\perp$ or not with probability $1/2$. When $m' \neq \perp$ and the guess is correct, we have $\mathsf{Enc}'(\mathsf{pk}, m') = c$, as Enc' is rigid deterministic and perfectly correct. Then we could find the pair (m', c) that satisfies $\mathsf{Enc}'(\mathsf{pk}, m') = c$ in the database, and store the responses in a quantum register in superposition. Now we measure this register in the computational basis to get the classical response to (m', c). This response may be in two cases: $k^* \in \mathcal{K}$, or $\perp \notin \mathcal{K}$. The latter implies that $H(m', c)$ has not been defined, so we use a random $k^* \leftarrow\!\!\$\, \mathcal{K}$ to replace it. Now, we let k^* be the response to the decapsulation oracle query on c. To make the random oracle responses consistent, in the subsequent random oracle query, we respond with k^* if the query is (m', c), or still use the compressed oracle to obtain the responses otherwise. This completes the decapsulation simulation and the proof sketch in the QROM. For generality, we further extend this method into a reprogram-after-measure technique, which can address the oracle reprogramming issue encountered during the single classical query in the QROM, and is proved in Section 4.1.

The Proof in the QROM. Similar to the ROM proof of T_{RH}^{D} (see Theorem 4.2), the QROM proof also can be divided into following two steps:

1. The first step (games G_0 to G_4): We use a random $k \leftarrow\!\!\$\, \mathcal{K}$ to replace the $k := H(m, c)$ in Encaps, and use the double-sided O2H lemma (Lemma 2.1) to convert the advantage of $\mathcal{A}$ detecting this change to the probability of a new adversary $\mathcal{B}$ outputting the corresponding m.

2. The second step (games G_5 to G_8): We use the proposed reprogram-after-measure technique to simulate the decapsulation oracle without sk, and then use the ability of $\mathcal{B}$ to attack the OW-CPA security of PKE'.

The tightness of this QROM proof results from our reprogram-after-measure technique that has a tighter upper bound than the measure-and-reprogram technique used in [33].

1.3 Discussions

Note that Jiang et al. [33] proved that there are unavoidable reduction losses of $\mathcal{O}(q)$ and $\mathcal{O}(q^2)$ in the security proof of T_{RH} in the ROM and the QROM, respectively, where q is the number of random oracle queries made by the adversary. We should stress that instead of indicating any flaw in Jiang et al.'s negative results, our work merely demonstrates that there is a special case which is not captured in their given proofs.

The technique used by Jiang et al. to prove the negative results is a so-called meta-reduction technique [5, 17, 28]. In the case where the underlying PKE' is malleable, the main idea is to use the decapsulation oracle to construct an adversary $\mathcal{B}$ to attack the IND-1CCA security of KEM_{RH} directly.

In the ROM, roughly speaking, given the challenge encapsulation $c^* \leftarrow \mathsf{Enc}'(\mathsf{pk}, m^*)$, $\mathcal{B}$ uses the malleability of PKE' to construct a new encapsulation c' from c^* such that $\mathsf{Dec}(\mathsf{sk}, c') = f(m^*)$, where f is a special function related to the property of the malleability. Then, $\mathcal{B}$ makes a decapsulation query on c' to obtain a $\mathsf{tag} := H(f(m^*), c')$. Finally, through q many H random oracle queries, $\mathcal{B}$ attempts to get $m^* \in \mathcal{M}$ such that $H(f(m^*), c') = \mathsf{tag}$, and then uses $H(m^*, c^*)$ to distinguish k_0 from k_1. Assume that the underlying PKE scheme PKE' is

λ-bit secure, which implies that the probability for any PPT adversary breaking the OW-CPA security of PKE' is no more than $\mathcal{O}(1/2^\lambda)$. Therefore, after q many H random oracle queries, the probability of getting m^* is no more than $\mathcal{O}(q/2^\lambda)$. Therefore, they claim that there is an unavoidable reduction loss of $\mathcal{O}(q)$ in the security proof for T_{RH} in the ROM.

However, we can note that, in the aforementioned attack, the role of the decapsulation oracle is to generate a tag, which is the image of m^* under a deterministic mapping $g(\cdot) := H(f(\cdot), c')$, and the q many H random oracle queries are used to guess the preimage of tag under g. However, in the case where PKE' is deterministic, Enc' itself can provide a deterministic mapping from m^* to c^* that neither relies on the use of the decapsulation oracle nor on queries to the H random oracle. This implies that, at this point, $\mathcal{B}$ does not require access to the decapsulation oracle or the H random oracle; he simply invokes Enc' q times to achieve the effect of invoking the H random oracle q times. Note that $\mathcal{B}$'s advantage in the OW-CPA game of PKE' surpasses the probability of successfully guessing the plaintext m^* corresponding to the ciphertext c^* by invoking Enc' q times. Consequently, the aforementioned conclusion regarding an unavoidable reduction loss of $\mathcal{O}(q)$ in the ROM is inapplicable when PKE' is a deterministic public-key encryption scheme. (Clearly, we also present a security proof with a reduction loss of $\mathcal{O}(1)$ as evidence.) Note that in the case where PKE' is probabilistic, the security proof given by Jiang et al. [33] incurs a reduction loss of $\mathcal{O}(q^2)$, which is not as *tight* as claimed in their negative results. Hence, an intriguing open question is whether this $\mathcal{O}(q^2)$ reduction loss is unavoidable when PKE' is probabilistic, or if the reduction loss in this case can be refined.

In the QROM, similar to the case in the ROM, the core idea is to use the malleability of PKE' to generate a new encapsulation c' from c^*, where c^* is the challenge encapsulation, $\mathsf{Enc}(\mathsf{pk}, m^*) = c^*$, $\mathsf{Dec}(\mathsf{sk}, c') = f(m^*)$, and f is a function related to the malleability of PKE', make the decapsulation oracle query on c' to obtain $k' = H(f(m^*), c')$, use the Grover's algorithm [26] to find m^* from k', and then use $H(m^*, c^*)$ to distinguish k_0 from k_1, where the Grover's algorithm needs q times Grover iterations, and each Grover iteration needs to make a random oracle query. Jiang et al. [33] show that this method to distinguish k_0 from k_1 can succeed with probability at least $(q+1)^2/|\mathcal{M}|$, and can derive the conclusion that reduction loss $\mathcal{O}(q^2)$ in the security proof for T_{RH} in the QROM is unavoidable.

Similar to the analysis in the ROM, the use of the random oracle is to compute the deterministic mapping $g(\cdot) := H(f(\cdot), c')$ in order to recover m^* from k', but the deterministic PKE' itself can provide the deterministic mapping Enc' from m^* to c^*. Therefore, the Grover iteration can use Enc' instead of the random oracle to achieve the same purpose, which implies that the conclusion about the unavoidable reduction loss $\mathcal{O}(q^2)$ for the security proof of T_{RH} in the QROM is inapplicable when PKE' is deterministic.

We should note that in the above analyses, we do not require that the deterministic PKE' should be rigid. Therefore, there is an open problem that when the underlying PKE is deterministic but not rigid, whether the reduction losses of $\mathcal{O}(q)$ and $\mathcal{O}(q^2)$ given by Jiang et al. [33] in the ROM and the QROM, respectively, can be improved or not.

1.4 Related Work

The quantum random oracle model (QROM) [11] has been a popular model to analyze the security of some post-quantum cryptographic schemes, such as encryption [38, 54], signature [1, 9, 24, 46], authenticated key exchange (AKE) [30, 41, 44], classical verification of quantum computations (CVQC) [6, 15], and other cryptographic primities [4, 29, 32]. Many works [53, 56] showed that there exist schemes that are secure in the ROM but insecure in the QROM, which implies that the QROM is stronger than the ROM.

T_{CH}, T_H, and T_{RH} can be seen as the simplified versions of the FO-like transformation, where the FO-like transformation is a variant of the Fujisaki-Okamoto transformation [22, 23] under KEM. Targhi et al. [50] and Hofheinz et al. [27] conducted the first analyses of the security of FO transformation and FO-like transformation in the QROM, respectively. However, these works need to introduce an additional hash function to achieve post-quantum security, and the proofs suffer from the quartic reduction loss. For the case where the KEM is implicit reject, Jiang et al. [34] provided a proof for the FO-like transformation without the additional hash, where the degree the reduction loss is decreased from quartic to quadratic, and the factor of the reduction loss is $\mathcal{O}\big(q^2\big)$. Jiang et al. [37] further pointed out that quadratic loss is unavoidable in the measurement-based black-box reduction, where the adversary is accessed in a black-box manner and is only run once without rewinding, and the reduction algorithm is performed by measuring the state of the adversary. In the following works, Jiang et al. [36] used the semi-classical O2H lemma proposed by Ambainis [3] to improve the security reduction to $\epsilon_R \approx \mathcal{O}(q)\epsilon_{\mathcal{A}}^2$, while Bindel et al. [10] proposed the double-sided O2H lemma to improve the security reduction to $\epsilon_R \approx \epsilon_{\mathcal{A}}^2$. To investigate a tighter transformation, Saito et al. [47] proposed the SXY transformation based on the FO-like transformation, and got a tight security reduction to the newly defined security called disjoint simulatability. This tight result is extended by Jiang et al. [35] to the KEM with explicit reject. Considering stronger quantum adversaries, Xagawa and Yamakawa [52] further proved the IND-QCCA security[2] of these PKE-to-KEM transformations [35, 47] in the QROM. To remove the quadratic loss, Kuchta et al. [40] provided the measure-rewind-measure lemma and obtained a security reduction with tightness $\epsilon_R \approx \mathcal{O}(1/q)\epsilon_{\mathcal{A}}$. As previous works mainly focused on the cases where the underlying PKE has negligible decryption errors, Cini et al. [16] proposed a new transformation that can work for the PKE with non-negligible decryption errors. In addition, Kitagawa and Nishimaki [39] and Pan and Zeng [45] further considered other security notions of the FO-like transformations, named key dependent message (KDM) security and selective opening security (SO) against chosen-ciphertext attacks, respectively.

The compressed oracle technique is a useful tool provided by Zhandry [55]. Based on it, Don et al. [20] proposed an online extractor and provided a proof for the *textbook* FO transformation with tightness $\epsilon_R \approx \mathcal{O}\big(1/q^2\big)\epsilon_{\mathcal{A}}^2$. Using a similar method, Shan et al. [48] and Ge et al. [25] began to analyze the IND-QCCA security of the FO-like transformation. T_{CH} and T_H are proposed by Huguenin-Dumittan and Vaudenay [31], but the QROM proof for T_H is left. Jiang et al. [33] proposed and provided the ROM and QROM proofs for T_{RH}, and related the IND-1CCA security of T_{RH} to that of T_H in the QROM. However, their proofs of T_{RH} can be improved when the underlying PKE is rigid deterministic.

2 Preliminaries

2.1 Notation

We represent the function H with domain $\mathcal{X}$ and codomain $\mathcal{Y}$ as $H : \mathcal{X} \to \mathcal{Y}$. We denote the set of such functions as Ω_H. For any set $\mathcal{S}$, we use $|\mathcal{S}|$ to represent its cardinality and use $s \leftarrow\!\!\$\, \mathcal{S}$ to denote the random choice of an element s from $\mathcal{S}$ with uniform probability. To indicate the output of a probabilistic (or deterministic) algorithm A with input x as y, we use the notation $y \leftarrow A(x)$ (or $y := A(x)$). Additionally, A^H (or $A^{|H\rangle}$) denotes an oracle algorithm that has classical (or quantum) access to the oracle H. We utilize the notation $[x = y]$ to represent an integer value of 1 when $x = y$ and 0 otherwise. The security parameter is denoted by λ, and PPT stands for *probabilistic polynomial time.* The base of logarithm log is 2, unless stated otherwise.

[2] The decapsulation oracle can also be accessed in superposition.

2.2 The (Quantum) Random Oracle Model

For the introduction to the fundamentals of quantum computation, we recommend readers refer to [42]. In brief, the *state space* of a quantum system is a complex vector space with an inner product. The Dirac notation "$|\cdot\rangle$" (and "$\langle\cdot|$") is used to represent unit vectors, known as *state vectors*, in the state space (and their counterparts in the dual space). The state space can be spanned by a set of orthonormal bases called *computational bases*. The *joint state* of $|\psi\rangle$ and $|\phi\rangle$ is $|\psi\rangle \otimes |\phi\rangle$. The *norm* of a state $|\psi\rangle$, denoted as $\||\psi\rangle\|$, is calculated as $\sqrt{\langle\psi|\psi\rangle}$, where "$\langle\psi|\phi\rangle$" signifies the inner product between $|\psi\rangle$ and $|\phi\rangle$.

The random oracle model (ROM), as introduced in [7], is an idealized model where the hash function is modeled as a publicly accessible random oracle. In this model, to get the value of $H(x)$ for a given hash function H, an adversary must make a H random oracle query on x. The quantum analog of this model, known as the quantum random oracle model (QROM) [11], permits adversaries to make the random oracle queries in a superposition state. Here, the H random oracle behaves as a unitary transformation, mapping $|x, y\rangle$ to $|x, y \oplus H(x)\rangle$. It is worth noting that traditional, or "classical", queries are still permissible in the QROM. These can be interpreted as first querying the random oracle on $|x, 0\rangle$ and then measuring the second register to obtain the classical output [20].

2.3 The One-Way to Hiding Lemma

In the ROM, random oracles serve as a crucial tool for learning the adversary's queries. An adversary cannot learn any knowledge about $H(x)$ without querying the H random oracle for x. Furthermore, without querying the random oracle at x, the adversary cannot discover the reprogramming of the oracle at that point. Under certain conditions in the QROM, the simulator can exploit the adversary's behavior to identify the point of random oracle reprogramming by employing the "one-way to hiding (O2H)" lemma. In this work, we adopt the version of the O2H lemma introduced by Bindel et al. [10], which has a tight bound except for a quadratic loss that is impossible to avoid [37].

Lemma 2.1 (Double-Sided One-Way to Hiding [10])**.** *Let $G, H : \mathcal{X} \to \mathcal{Y}$ be random functions such that $\forall x \neq x^* \in \mathcal{X}$, $G(x) = H(x)$, and z be a random value, where (G, H, x^*, z) may have arbitrary joint distribution. Let $A^{|H\rangle}$ be an oracle algorithm that has quantum access to the H random oracle. Then there exists a double-sided oracle algorithm $B^{|G\rangle,|H\rangle}$ that can access both G and H, such that*

$$\left|\Pr[\mathsf{Ev} : A^{|G\rangle}(z)] - \Pr[\mathsf{Ev} : A^{|H\rangle}(z)]\right| \leq 2\sqrt{\Pr[\hat{x} = x^* : \hat{x} \leftarrow B^{|G\rangle,|H\rangle}(z)]}$$

for an arbitrary classical event Ev.

2.4 The Compressed Oracle Technique

The reduction in the ROM is allowed to record the adversaries' queries, but this feature was once considered impossible in the QROM. This is due to the quantum no-cloning principle, which implies that any direct recording of a quantum state would alter the adversary's state. Fortunately, Zhandry [55] overcomes this "recording barrier" by introducing the compressed oracle technique. The basic idea is to purify the quantum random oracle and then record adversaries' queries on the purified quantum random oracle.

Definition 2.1 (Compressed Standard Oracle)**.** Let D represent the database composed of q pairs $(x, y) \in (\mathcal{X} \times \mathcal{Y}) \cup (\perp, 0^n)$ where $n := \log |\mathcal{Y}|$ and q signifies the maximum quantum random oracle queries a quantum adversary could make. The structure of D is as follows:

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

where $0 \le l \le q$, $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ for $i = 1, \cdots, l$, $x_1 < \cdots < x_l$, and D ends with $q - l$ pairs of $(\perp, 0^n)$. We denote the set of such databases as $\mathcal{D}$. For any $x \in \mathcal{X}$, if there exists a y such that $(x, y) \in D$, then we define $D(x) = y$; otherwise, $D(x) = \perp$. Notably, no two pairs in D share the same x. We use $|D|$ to denote the number of (x, y) pairs in D where $x \neq \perp$. When $|D| < q$ and $D(x) = \perp$, we define $D \cup (x, y)$ as the operation of removing one $(\perp, 0^n)$ entry from D and then inserting (x, y) while preserving the ascending order of x values.

Let D be a quantum register with state space $\mathcal{H} = \mathbb{C}[\mathcal{D}]$. On the basis state $|D\rangle$ (where $D \in \mathcal{D}$), we define a unitary decompression procedure F_x as follows:

- If $D(x) = \perp$ and $|D| < q$, we have

$$F_x |D\rangle = 2^{-n/2} \sum_y |D \cup (x, y)\rangle \ ,$$

$$F_x \left(2^{-n/2} \sum_y |D \cup (x, y)\rangle \right) = |D\rangle \ ,$$

$$F_x \left(2^{-n/2} \sum_y (-1)^{z \cdot y} |D \cup (x, y)\rangle \right) = 2^{-n/2} \sum_y (-1)^{z \cdot y} |D \cup (x, y)\rangle \ \text{where } z \neq 0 \ .$$

- If $D(x) = \perp$ but $|D| = q$, we have $F_x |D\rangle = |D\rangle$.

Let X and Y be the input and output registers of the quantum random oracle, respectively. We define a unitary operator O_x that is applied to YD as

$$O_x : |y, D\rangle \to |y \oplus D(x), D\rangle \ .$$

Note that unlike the definition in [55] where $y \oplus \perp = y$, here we define $0^n \oplus \perp = \perp$, $\perp \oplus 0^n = \perp$, $\perp \oplus \perp = 0^n$, and for $y \in \mathcal{Y} \setminus \{0^n\}$, $y \oplus \perp = y$, $\perp \oplus y = \perp$[3]. In the end, the compressed standard oracle applied to XYD can be defined as

$$\mathsf{CStO} := \sum_x |x\rangle \langle x| \otimes F_x O_x F_x \ .$$

The compressed standard oracle is proved to be perfectly indistinguishable from the quantum random oracle by Zhandry [55].

Lemma 2.2 (Lemma 4 in [55])**.** *The compressed oracle as defined in Definition 2.1 with* D *set as* $\bigotimes_{i=1}^{q} (\perp, 0^n)$ *initially is perfectly indistinguishable from a quantum random oracle* $H : \mathcal{X} \to \mathcal{Y}$ *for any quantum adversary making at most* q *random oracle queries.*

2.5 Cryptographic Primitives

Definition 2.2 (Public-Key Encryption)**.** The public-key encryption (PKE) scheme is composed of three PPT algorithms with the security parameter λ, a message space $\mathcal{M}$, and a ciphertext space $\mathcal{C}$: (1) The key generation algorithm Gen is a probabilistic algorithm that takes as input 1^λ and outputs a public/private key pair $(\mathsf{pk}, \mathsf{sk})$. (2) The encryption algorithm

[3] With this definition, we can verify that $O_x O_x = I$, indicating that the adjoint of O_x is itself, and thus O_x is unitary.

Enc is a probabilistic algorithm that takes as input pk and a message $m \in \mathcal{M}$, and outputs a ciphertext $c \in \mathcal{C}$. (3) The decryption algorithm Dec is a deterministic algorithm that takes as input sk and $c \in \mathcal{C}$, and outputs $m \in \mathcal{M}$ or a special $\perp \notin \mathcal{M}$ value.

The correctness requirement of a PKE is that for all possible outputs $(\mathsf{pk}, \mathsf{sk})$ of $\mathsf{Gen}(1^\lambda)$, and all possible outputs c of $\mathsf{Enc}(\mathsf{pk}, m)$, we have $\mathsf{Dec}(\mathsf{sk}, c) = m$. We say a PKE scheme is deterministic if Enc is a deterministic algorithm.

Definition 2.3 (δ-correctness [20]). We say a PKE scheme is δ-correct if

$$\mathbb{E}_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)} \left[\max_{m \in \mathcal{M}} \Pr[\mathsf{Dec}(\mathsf{sk}, c) \neq m : c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)] \right] \leq \delta \ .$$

If $\delta = 0$, then we say the PKE scheme is perfectly correct.

Definition 2.4 (rigidity [8]). We say a deterministic PKE scheme is rigid if $\mathsf{Enc}(\mathsf{pk}, m) = c$ for every $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$, every $c \in \mathcal{C}$, and $m := \mathsf{Dec}(\mathsf{sk}, c)$, when the PKE is correct.

Definition 2.5 (The OW-CPA Security of PKE). We define the OW-CPA security of a PKE scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ in terms of an attack game between a challenger and an adversary $\mathcal{A}$, as follows. The challenger computes

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda), \ m^* \leftarrow\!\!\$\ \mathcal{M}, \ c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*) \ ,$$

and sends (pk, c^*) to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs $\hat{m} \in \mathcal{M}$. We define $\mathcal{A}$'s advantage with respect to PKE as $\mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{A}) := \Pr[m^* = \hat{m}]$, and if this advantage is negligible for all PPT adversaries, we say that PKE is OW-CPA secure. We refer to the m^* and the c^* computed by the challenger as the challenge message and the challenge ciphertext, respectively.

Definition 2.6 (Key Encapsulation Mechanism). Key encapsulation mechanism (KEM) is specified by three PPT algorithms with the security parameter λ, a key space $\mathcal{K}$, and an encapsulation space $\mathcal{C}$: (1) The key generation algorithm Gen is a probabilistic algorithm that takes as input 1^λ and outputs a public/private key pair $(\mathsf{pk}, \mathsf{sk})$. (2) The encapsulation algorithm Encaps is a probabilistic algorithm that takes as input pk and outputs a pair (k, c) where the key $k \in \mathcal{K}$ and the encapsulation $c \in \mathcal{C}$. (3) The decapsulation algorithm Decaps is a deterministic algorithm that takes as input sk and $c \in \mathcal{C}$, and outputs $k \in \mathcal{K} \cup \{\perp\}$.

The correctness requirement of a KEM is that for all possible outputs $(\mathsf{pk}, \mathsf{sk})$ of $\mathsf{Gen}(1^\lambda)$, and all possible outputs (k, c) of $\mathsf{Encaps}(\mathsf{pk})$, we have $\mathsf{Decaps}(\mathsf{sk}, c) = k$. We usually say that a KEM is explicit reject if $\perp \notin \mathcal{K}$, while a KEM is implicit reject if $\perp \in \mathcal{K}$ represents a random value.

Definition 2.7 (The IND-1CCA Security of KEM). We define the IND-1CCA security of a KEM scheme $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ in terms of an attack game between a challenger and an adversary $\mathcal{A}$, as follows. The challenger computes

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda), \ (k_0, c^*) \leftarrow \mathsf{Encaps}(\mathsf{pk}), \ k_1 \leftarrow\!\!\$\ \mathcal{K}, \ b \leftarrow\!\!\$\ \{0, 1\} \ ,$$

and sends (pk, c^*, k_b) to $\mathcal{A}$. In this game, $\mathcal{A}$ can make at most one decapsulation query on any $c \neq c^*$. Finally, $\mathcal{A}$ outputs $\hat{b} \in \{0, 1\}$. We define $\mathcal{A}$'s advantage with respect to KEM as $\mathsf{Adv}^{\text{IND-1CCA}}_{\mathsf{KEM}}(\mathcal{A}) := \left| \Pr[b = \hat{b}] - 1/2 \right|$, and if this advantage is negligible for all PPT adversaries, we say that KEM is IND-1CCA secure. We refer to the c^* and the k_b sent to $\mathcal{A}$ as the challenge encapsulation and the challenge key, respectively.

Theorem 2.3 (Difference Lemma [12]). *Let Z, W_1, W_2 be some events defined over some probability space, and $\bar{Z}$ be the complement of Z. Assume that $W_0 \wedge \bar{Z}$ occurs if and only if $W_1 \wedge \bar{Z}$ occurs, then we have $|\Pr[W_0] - \Pr[W_1]| \leq \Pr[Z]$.*

Games G_0 to G_5:

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}'(1^\lambda),\ m^* \leftarrow\!\!\$\ \mathcal{M}$
$c^* := \mathsf{Enc}'(\mathsf{pk}, m^*),\ k_0 \leftarrow\!\!\$\ \mathcal{K},\ k_1 \leftarrow\!\!\$\ \mathcal{K}$
initialize an empty associative array
$\quad Map : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$
$Map[(m^*, c^*)] := k_0 \quad /\!/\ G_0 - G_1$
$b \leftarrow\!\!\$\ \{0,1\},\ \mathsf{cnt} := 0$
$\mathsf{flag} \leftarrow\!\!\$\ \{0,1\} \quad /\!/\ G_3 - G_5$
$\mathrm{List}^{O_{\mathsf{Dec}}} := \perp,\ k^* \leftarrow\!\!\$\ \mathcal{K} \quad /\!/\ G_4 - G_5$
if COLL_1 **then** $\quad /\!/\ G_1 - G_5$
$\quad$ **return** $\perp \quad /\!/\ G_1 - G_5$
$\hat{b} \leftarrow\!\!\$\ \mathcal{A}^{O_{\mathsf{Dec}}, H}(\mathsf{pk}, c^*, k_b)$
return $[b = \hat{b}]$

Random Oracle $H(m, c)$:

if $(m, c) \notin \mathrm{Domain}(Map)$ **then**
$\quad Map[(m, c)] \leftarrow\!\!\$\ \mathcal{K}$
$\quad$ **if** $c \in \mathrm{List}^{O_{\mathsf{Dec}}}$ **then** $\quad /\!/\ G_4 - G_5$
$\quad\quad$ **if** $(\mathsf{flag} = 0$ **and** $m = \star)$
$\quad\quad$ **or** $(\mathsf{flag} = 1$ **and** $\mathsf{Enc}'(\mathsf{pk}, m) = c)$
$\quad\quad$ **then** $\quad /\!/\ G_4 - G_5$
$\quad\quad\quad Map[(m, c)] := k^* \quad /\!/\ G_4 - G_5$
return $Map[(m, c)]$

Decapsulation Oracle $O_{\mathsf{Dec}}(c \neq c^*)$:

if $\mathsf{cnt} = 0$ **then**
$\quad \mathsf{cnt} := \mathsf{cnt} + 1$
$\quad$ **if** COLL_2 **then** $\quad /\!/\ G_1 - G_5$
$\quad\quad$ **return** $\perp \quad /\!/\ G_1 - G_5$
$\quad \mathrm{List}^{O_{\mathsf{Dec}}}.append(c) \quad /\!/\ G_4 - G_5$
$\quad m' := \mathsf{Dec}'(\mathsf{sk}, c) \quad /\!/\ G_0 - G_4$
$\quad$ **if** $m' = \perp$ **then** $\quad /\!/\ G_0 - G_2$
$\quad$ **if** $\mathsf{flag} = 0$ **then** $\quad /\!/\ G_3 - G_5$
$\quad\quad$ **return** $k' := H(\star, c) \quad /\!/\ G_0 - G_3$
$\quad\quad$ **if** $(\star, c) \in \mathrm{Domain}(Map)$
$\quad\quad$ **then** $\quad /\!/\ G_4 - G_5$
$\quad\quad\quad$ **return** $k' := Map[(\star, c)] \quad /\!/\ G_4 - G_5$
$\quad$ **return** $k' := H(m', c) \quad /\!/\ G_0 - G_3$
$\quad$ **if** $\exists (m', c) \in \mathrm{Domain}(Map)$
$\quad$ $s.t.$ $\mathsf{Enc}'(\mathsf{pk}, m') = c$ **then** $\quad /\!/\ G_4 - G_5$
$\quad\quad$ **return** $k' := Map[(m', c)] \quad /\!/\ G_4 - G_5$
$\quad$ **return** $k' := k^* \quad /\!/\ G_4 - G_5$
return $k' := \perp$

Figure 2: Games G_0 to G_5 for the proof of Theorem 3.1.

3 The Security of T_{RH} in the ROM

Here, we prove that the IND-1CCA security of $\mathsf{KEM}_{RH} := T_{RH}[\mathsf{PKE}', H]$ can be tightly reduced to the OW-CPA security of PKE' in the ROM, if PKE' is rigid deterministic.

Theorem 3.1 (The security of T_{RH} in the ROM)**.** *Assume $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ is modeled as a random oracle. If PKE' is a rigid deterministic public-key encryption scheme that is δ-correct and* OW-CPA *secure, then KEM_{RH} is* IND-1CCA *secure.*

In particular, for any PPT *adversary $\mathcal{A}$ that attacks the* IND-1CCA *security of KEM_{RH}, there exists a* PPT *adversary $\mathcal{B}$ that attacks the* OW-CPA *security of PKE', such that*

$$\mathsf{Adv}^{\text{IND-1CCA}}_{\mathsf{KEM}_{RH}}(\mathcal{A}) \leq 2\left(\mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}'}(\mathcal{B}) + \delta\right) .$$

Theorem 3.1. Fig. 2 shows the simulation of the challenger for the adversary $\mathcal{A}$ in game G_j for $j = 0, \ldots, 5$. In each game, b is a random bit chosen by the challenger, while $\hat{b}$ is the bit output by $\mathcal{A}$ at the end of the game. We define W_j to be the event that $\hat{b} = b$ in game G_j.

Game G_0. In this game, the challenger explicitly initializes an empty associative array $Map : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ to implement the random oracle. In the initialization step, k_0 is chosen uniformly over $\mathcal{K}$ and then is stored in $Map[(m^*, c^*)]$. This is equivalent to setting the value of the random oracle at (m^*, c^*) to k_0. We can see that, except for the extra records of responses from the random oracle, the behavior of the challenger is clearly consistent with that in the IND-1CCA game of $\mathsf{KEM}_{RH} := T_{RH}[\mathsf{PKE}', H]$. Therefore,

$$|\Pr[W_0] - 1/2| = \mathsf{Adv}^{\text{IND-1CCA}}_{\mathsf{KEM}_{RH}}(\mathcal{A}) \ . \tag{1}$$

Game G_1. This game is the same as game G_0 except that events COLL_1 and COLL_2 do not occur, where COLL_1 (or COLL_2) denotes that decrypting the encapsulation $c^* = \mathsf{Enc}'(\mathsf{pk}, m^*)$ received by $\mathcal{A}$ (or the decapsulation oracle query $c = \mathsf{Enc}'(\mathsf{pk}, m')$ issued by $\mathcal{A}$) with Dec using sk would obtain m such that $m \neq m^*$ (or $m \neq m'$). By the δ-correctness of PKE', the probability of either COLL_1 or COLL_2 occurring is no greater than δ. Therefore,

$$|\Pr[W_1] - \Pr[W_0]| \leq 2\delta \ . \tag{2}$$

Game G_2. This game is the same as game G_1 except that assigning k_0 to $Map[(m^*, c^*)]$ is removed from the initialization step. Let Z_j be the event that $\mathcal{A}$ makes an H random oracle query on (m^*, c^*) in game G_j, then this game and game G_1 proceed identically until Z_1 or Z_2 occurs. By the Difference Lemma (Theorem 2.3), we have

$$|\Pr[W_2] - \Pr[W_1]| \leq \Pr[Z_2] \ . \tag{3}$$

Here, since k_0 and k_1 are both randomly chosen from $\mathcal{K}$, and are both irrelevant to the two oracles, b is independent of $\mathcal{A}$'s view. Therefore,

$$\Pr[W_2] = 1/2 \ . \tag{4}$$

Game G_3. This game modifies the initialization step and the decapsulation oracle in game G_2. In the initialization step, the challenger picks an extra random bit flag. In the decapsulation oracle, the condition $m' = \perp$ is replaced by $\mathsf{flag} = 0$. One can note that if $m' = \perp$ when $\mathsf{flag} = 0$, or if $m' \neq \perp$ when $\mathsf{flag} = 1$, game G_3 is entirely identical to game G_2[4], thereby

$$\Pr[Z_2] = \Pr[Z_3 \wedge m' = \perp | \mathsf{flag} = 0] + \Pr[Z_3 \wedge m' \neq \perp | \mathsf{flag} = 1] \ . \tag{5}$$

Game G_4. Compared with game G_3, we make the following modifications to answer the decapsulation query without using sk. Firstly, in the initialization step, the challenger initializes an extra empty list $\text{List}^{O_{\mathsf{Dec}}}$ to store the c queried to the decapsulation oracle, and chooses a random $k^* \leftarrow\!\$\, \mathcal{K}$ for the decapsulation oracle query. The decapsulation oracle works as follows.

- Case $\mathsf{flag} = 0$: If $(\star, c)$ has been queried in the H random oracle, then return $Map[(\star, c)]$; otherwise, return k^*.
- Case $\mathsf{flag} = 1$: If there exists $(m', c) \in \text{Domain}(Map)$ where $\mathsf{Enc}'(\mathsf{pk}, m') = c$, then return $Map[(m', c)]$; otherwise, return k^*.

In the H random oracle, for the new query (m, c), we introduce the following operations: if c has been queried to the decapsulation oracle, i.e., $c \in \text{List}^{O_{\mathsf{Dec}}}$, then if $m = \star$ when $\mathsf{flag} = 0$, or if $\mathsf{Enc}'(\mathsf{pk}, m) = c$ when $\mathsf{flag} = 1$, we reprogram $Map[(m, c)]$ to k^*.

Recall that we have assumed COLL_2 would not occur since game G_1, and that PKE' is rigid deterministic. This means that for any c where $\mathsf{Dec}(\mathsf{sk}, c) = m' \neq \perp$, m' is the only value in

[4]One may note that there are two additional cases in game G_3, i.e., when $m' = \perp$ but flag $= 1$, and when $m' \neq \perp$ but flag $= 0$, requiring an extension of the domain of H to $\mathcal{M} \times \{\perp\}$ for $H(\perp, c)$ to be defined. Nevertheless, in our analysis of the relationship between G_3 and G_2, these cases are not pertinent and, thus, are omitted for brevity.

$\mathcal{M}$ such that $\mathsf{Enc}'(\mathsf{pk}, m') = c$. Therefore, if $\mathcal{A}$ has performed an H random oracle query on $(\star, c)$ when $\mathsf{flag} = 0$, or on (m', c) when $\mathsf{flag} = 1$, before the decapsulation query of c, then the decapsulation oracle will return the corresponding random oracle value, which is consistent with the behavior in game G_3 in the same case. If $\mathcal{A}$ does not make an H random oracle query on $(\star, c)$ or (m', c), then the decapsulation oracle will return k^*, but in the subsequent H random oracle query on the corresponding $(\star, c)$ or (m', c), it will also respond with the same k^*. Since k^* is chosen randomly, the behavior at this time is consistent with that in game G_3. Therefore,

$$\begin{aligned} &\Pr[Z_4 \wedge m' = \perp | \mathsf{flag} = 0] = \Pr[Z_3 \wedge m' = \perp | \mathsf{flag} = 0] \\ &\Pr[Z_4 \wedge m' \neq \perp | \mathsf{flag} = 1] = \Pr[Z_3 \wedge m' \neq \perp | \mathsf{flag} = 1] \ . \end{aligned} \tag{6}$$

Combining (5) and (6), we obtain

$$\begin{aligned} \Pr[Z_4] &\geq \Pr[Z_4 \wedge m' = \perp \wedge \mathsf{flag} = 0] + \Pr[Z_4 \wedge m' \neq \perp \wedge \mathsf{flag} = 1] \\ &= \Pr[Z_4 \wedge m' = \perp | \mathsf{flag} = 0] \Pr[\mathsf{flag} = 0] \\ &\quad + \Pr[Z_4 \wedge m' \neq \perp | \mathsf{flag} = 1] \Pr[\mathsf{flag} = 1] \\ &= \frac{1}{2} \left(\Pr[Z_4 \wedge m' = \perp | \mathsf{flag} = 0] + \Pr[Z_4 \wedge m' \neq \perp | \mathsf{flag} = 1] \right) \\ &= \frac{1}{2} \left(\Pr[Z_3 \wedge m' = \perp | \mathsf{flag} = 0] + \Pr[Z_3 \wedge m' \neq \perp | \mathsf{flag} = 1] \right) \\ &= \frac{1}{2} \Pr[Z_2] \ . \end{aligned} \tag{7}$$

At this point, it can be observed that the response of the decapsulation oracle in game G_4 no longer depends on $m' := \mathsf{Dec}'(\mathsf{sk}, c)$. Therefore, removing this step has no impact on $\Pr[Z_4]$.

Game 5. This game is the same as game G_4, except for removing the step $m' := \mathsf{Dec}'(\mathsf{sk}, c)$ in the decapsulation oracle. From the above discussion, we have

$$\Pr[Z_5] = \Pr[Z_4] \ . \tag{8}$$

At this point, we can find that all the oracles do not depend on sk and m^*. Therefore, when the event Z_5 occurs, we can construct an adversary $\mathcal{B}$ to attack the OW-CPA security of PKE' as follows: When $\mathcal{B}$ received the public key pk and the challenge ciphertext c^* from the OW-CPA game of PKE', he chooses a random $k \leftarrow\$ \mathcal{K}$, and then sends (pk, c^*, k) to $\mathcal{A}$. After that, he uses the decapsulation oracle and H random oracle described in game G_5 to respond to $\mathcal{A}$'s queries. At the end of the game, $\mathcal{B}$ can search the pair (m^*, c^*) in Map that satisfies $\mathsf{Enc}'(\mathsf{pk}, m^*) = c^*$ and output m^*. Therefore,

$$\Pr[Z_5] \leq \mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}'}(\mathcal{B}) \ . \tag{9}$$

Combining (1)-(4) and (7)-(9), we obtain

$$\mathsf{Adv}^{\text{IND-1CCA}}_{\mathsf{KEM}_{RH}}(\mathcal{A}) \leq 2 \left(\mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}'}(\mathcal{B}) + \delta \right) \ .$$

That completes the proof of the theorem. □

Remark 3.1. The bound given by Jiang et al. [33] in the case of deterministic PKE is

$$\mathsf{Adv}^{\text{IND-1CCA}}_{\mathsf{KEM}_{RH}}(\mathcal{A}) \leq (q_H + 1) \mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}'}(\mathcal{B}) + \delta \ ,$$

where q_H is the number of H random oracle queries. We prove that the reduction from IND-1CCA security of KEM_{RH} to the OW-CPA security of rigid deterministic PKE' is *tight*, with a loss factor of $\mathcal{O}(1)$.

4 The Security Analysis in the QROM

4.1 The Reprogram-after-Measure Technique

During the IND-1CCA game of KEM_{RH} in the ROM, the challenger needs to access the random oracle to calculate the response for the adversary $\mathcal{A}$ in the decapsulation oracle. But in some cases where the challenger does not know the point at which it should query the random oracle, the challenger can directly return a random response instead of accessing the random oracle, and the only requirement is that the random response should be consistent with the response to the corresponding random oracle query made by $\mathcal{A}$ in the subsequent process, e.g., Game 5 in the proof of Theorem 3.1. This process involves two techniques of ROM called *lazy sampling* and *reprogramming*, which are hard to carry over to the quantum setting as Boneh et al. [11] claim.

With the help of the compressed oracle technique introduced by Zhandry [55], we provide a new technique that can simulate the decapsulation oracle in a similar way. We will reprogram the compressed oracle after performing a measurement. Therefore, we refer to the proposed technique as *reprogram-after-measure*. Note that the decapsulation oracle query and the *implicit* random oracle query in it are both classical, and the classical decapsulation oracle is queried at most once. In section 4.2, we can see that we can obtain a *tighter* security proof with this new technique.

Theorem 4.1 (Reprogram-after-Measure)**.** *Let $A^{O,|H\rangle}$ be a quantum oracle algorithm that can make q_H times (quantum) H random oracle queries, but at most one (classical) O oracle query, where $O : \mathcal{C} \to \mathcal{Z}, H : \mathcal{X} \to \mathcal{Y}$. Let $\mathcal{C}^{\perp} \subseteq \mathcal{C}$ be a set on which A is not allowed to make the O oracle query, and for any $c \in \mathcal{C}^{\perp}$ the O oracle always returns $\perp$. For $c \in \mathcal{C} \setminus \mathcal{C}^{\perp}$, the O oracle computes $x := f^{-1}(c)$, (classically) accesses the H random oracle to obtain $y := H(x)$, and returns $g(y)$, where the functions $f : \mathcal{X} \to \mathcal{C}, g : \mathcal{Y} \to \mathcal{Z}$, and there is a unique preimage x for $c \in \mathcal{C} \setminus \mathcal{C}^{\perp}$ under f. Then there exists an algorithm B that does not need to access the O oracle and the H random oracle, and needs to know how to calculate the functions f and g (but does not need to know how to calculate f^{-1}), such that*

$$\Pr[\mathsf{Ev} : A^{O,|H\rangle}] \leq 2 \Pr[\mathsf{Ev} : B] \tag{10}$$

for any classical event Ev.

In particular, we can construct B from $A^{O,|H\rangle}$ as follows. Firstly, we use the compressed oracle CStO to replace the H random oracle in $A^{O,|H\rangle}$. Let XY be the input/output registers of CStO, D be the database register used by CStO that is initialized to $\bigotimes_{i=1}^{q_H+1}(\perp, 0^n)$ (note that $A^{O,|H\rangle}$ queries the H random oracle $q_H + 1$ times in total) where $n := \log(|\mathcal{Y}|)$, and CZ be the input/output registers of O oracle. We define a function $e : \mathcal{C} \times \mathcal{D} \to \mathcal{Y} \cup \perp$ as follows, where $\mathcal{D}$ is the database set as defined in Definition 2.1:

$$e(c, D) = \begin{cases} D(x) & \text{if there exists } (x, y) \in D \text{ such that } f(x) = c \text{ and } y \neq \perp , \\ \perp & \text{otherwise} . \end{cases}$$

Since e can be computed efficiently, the unitary operator $U_e : |c, D, y\rangle \to |c, D, y \oplus e(c, D)\rangle$ can also be implemented efficiently based on the quantum computation theory. B first chooses a random $y^* \leftarrow\$ \mathcal{Y}$, and then runs $A^{O,|H\rangle}$ until it makes an O oracle query (with classical input c on register C). If $c \in \mathcal{C}^{\perp}$, B directly sets register Z to $\perp$ and continues running $A^{O,|H\rangle}$ until the end; otherwise, instead of accessing the O oracle, B uses the following O^B oracle as a substitute, as shown in Fig. 3:

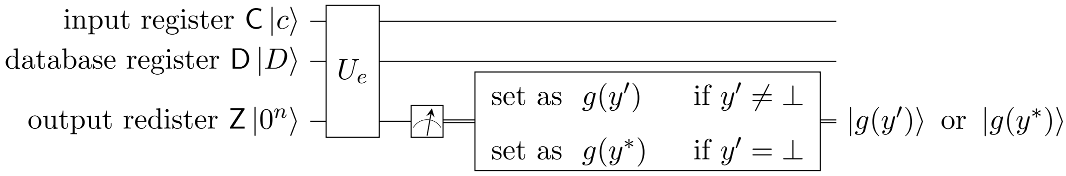

Figure 3: The quantum circuit diagram for O^B.

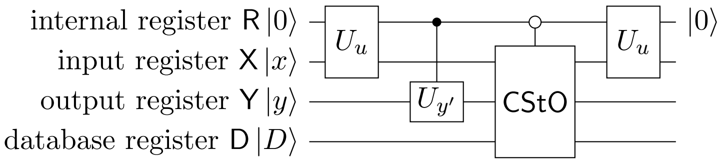

Figure 4: The quantum circuit diagram for CStO^B, where R is an internal register used by CStO^B.

1. *Initialize the register* Z *to* 0^n*.*

2. *Apply* U_e *to registers* CDZ*, where* Z *is the output register.*

3. *Perform the measurement* M_{Z} *on the register* Z *in the computational basis* $\{|y\rangle\}_{y\in\mathcal{Y}\cup\perp}$*, denoting the result as* $|y'\rangle$*.*

4. *If* $y' = \perp$*, let* $y' := y^*$*. Set* Z *to* $g(y')$*.*

After that, define a function $u_c(x) : \mathcal{X} \to \{0,1\}$ *as follows:*

$$u_c(x) = \begin{cases} 1 & \textit{if } f(x) = c \ , \\ 0 & \textit{otherwise} \ , \end{cases}$$

where $c \in \mathcal{C}$ *is the classical input on the register* C *when* $A^{O,|H\rangle}$ *queries the* O *oracle. Construct a unitary operator* $U_{u_c} : |x, b\rangle \to |x, b \oplus u_c(x)\rangle$*. In subsequent* H *random oracle queries,* B *uses the* CStO^B *oracle defined as follows (as shown in Fig. 4) instead of* CStO *to simulate the* H *random oracle:*

1. *Initialize a register* R *to* 0*, where* R *is a one qubit register.*

2. *Apply* U_{u_c} *to registers* XR*, where* R *is the output register.*

3. *Apply the following two conditional operations:*

   (a) *The control bit is* R*, and apply the unitary operator* $U_{y'}$ *to* Y *if* $b = 1$*, where* $U_{y'}|y\rangle = |y \oplus y'\rangle$ *and* y' *is the (classical) value obtained in the* O^B *oracle.*

   (b) *The control bit is* R*, and apply the unitary operator* CStO *to* XYD *if* $b = 0$*.*

4. *Apply* U_{u_c} *on* XR*, where* R *is the output register. Note that* R *is restored to* $|0\rangle$*, so it can be discarded.*

This completes the description of the construction of B.

Theorem 4.1. Here we use the same notation used in Theorem 4.1. Let $A^{O,|H\rangle}$ be the oracle algorithm defined in Theorem 4.1, where XY are the input/output registers of the H random oracle and CZ are the input/output registers of O oracle. We introduce a database register D that is initialized to $\bigotimes_{i=1}^{q_H+1}(\perp, 0^n)$ and use the compressed oracle CStO to implement the H random oracle in $A^{O,|H\rangle}$ to get a new oracle algorithm $\hat{A}^{O,|H\rangle}$. According to Lemma 2.2, we have

$$\Pr[\mathsf{Ev} : A^{O,|H\rangle}] = \Pr[\mathsf{Ev} : \hat{A}^{O,|H\rangle}] \tag{11}$$

for any classical event Ev.

Next, we analyze the relationship between $\hat{A}^{O,|H\rangle}$ and B, where B is defined in Theorem 4.1. Observe that the behavior of B is the same as that of $\hat{A}^{O,|H\rangle}$ until $\hat{A}^{O,|H\rangle}$ makes an O oracle query on $c \in \mathcal{C} \setminus \mathcal{C}^{\perp}$. In other words, if $\hat{A}^{O,|H\rangle}$ does not make an O oracle query, or if $\hat{A}^{O,|H\rangle}$ queries the O oracle on $c \in \mathcal{C}^{\perp}$, the behavior of B is exactly the same as that of $\hat{A}^{O,|H\rangle}$. In these two cases, equation (10) obviously holds. Therefore, in what follows, we only consider the case where $\hat{A}^{O,|H\rangle}$ makes only one (classical) O oracle query on $c \in \mathcal{C} \setminus \mathcal{C}^{\perp}$.

Consider that the H random oracle is invoked $q_H + 1$ times, where q_H times are direct quantum queries made by $\hat{A}^{O,|H\rangle}$, and 1 time is a classical query made through the O oracle. Without loss of generality, let the classical query be the i^*-th H random oracle query ($1 \leq i^* \leq q_H + 1$), and the execution of $\hat{A}^{O,|H\rangle}$ can be described as

$$U_{q_H+2}\left(\prod_{i=i^*+1}^{q_H+1} \mathsf{CStO} \circ U_i\right) O \circ U_{i^*}\left(\prod_{i=1}^{i^*-1} \mathsf{CStO} \circ U_i\right)|\psi_0\rangle \ ,$$

where $|\psi_0\rangle$ is the initial state of $\hat{A}^{O,|H\rangle}$, and for $i = 1, \ldots, q_H + 2$, U_i is a unitary operator[5]. Recall that the i^*-th (classical) H random oracle query is made through the O random oracle. The (non-unitary) O can be described by the following steps, where $\mathcal{C}^{\perp} \subseteq \mathcal{C}$ represents the set of c on which $\hat{A}^{O,|H\rangle}$ is not allowed to make the O oracle query:

1. If $c \in \mathcal{C}^{\perp}$, set Z to $\perp$; otherwise
2. Initialize a register X' to $x := f^{-1}(c)$.
3. Initialize the register Z to 0^n, and apply CStO to registers $\mathsf{X}'\mathsf{ZD}$.
4. Perform the measurement M_{Z} on the register Z in the computational basis $\{|y\rangle\}_{y\in\mathcal{Y}\cup\perp}$, denoting the result as $|y'\rangle$.
5. Compute $g(y')$ on the register Z.

The quantum circuit diagram for steps 2-5 is shown in Fig. 5.

Correspondingly, the execution of $B^{|H\rangle}$ can be described as

$$U_{q_H+2}\left(\prod_{i=i^*+1}^{q_H+1} \mathsf{CStO}^B \circ U_i\right) O^B \circ U_{i^*}\left(\prod_{i=1}^{i^*-1} \mathsf{CStO} \circ U_i\right)|\psi_0\rangle \ ,$$

where CStO^B and O^B are defined in Theorem 4.1.

[5]This follows from the fact that any quantum oracle algorithm can be transformed to a *unitary* quantum oracle with constant factor computational overhead and the same number of oracle queries [3, 25].

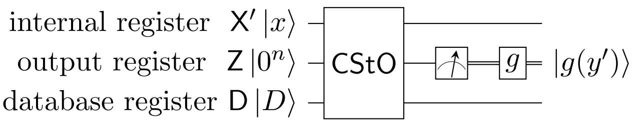

Figure 5: The quantum circuit diagram for steps 2-5 for O, where X' is an internal register used by O.

Since before querying O oracle or O^B oracle, the execution of $\hat{A}^{O,|H\rangle}$ and B are the same, they are in the same state at this time, denoted as $|\Psi\rangle$. Next, we consider the state $|\Psi\rangle$ on the register CZDP, where CZ are the input/output registers of O (or O^B) oracle, D is the database register used by CStO, and P contains all remaining registers of $\hat{A}^{O,|H\rangle}$ (or B).

Next, we divide $|\Psi\rangle$ into three mutually orthogonal parts (note that $c \notin \mathcal{C}^{\perp}$ and it is a certain classical value):

$$|\Psi\rangle = |\Psi_1\rangle + |\Psi_2\rangle + |\Psi_3\rangle \ ,$$

where

$$\begin{aligned}
|\Psi_1\rangle &= \sum_{\substack{z=0^n,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \beta_{z,D,p}\,|c,z,D,p\rangle \\
|\Psi_2\rangle &= \sum_{\substack{z=0^n,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}} \sum_{y_1\in\mathcal{Y}} (-1)^{r\cdot y_1}\,|c,z,D\cup(x,y_1),p\rangle \\
|\Psi_3\rangle &= \sum_{\substack{z=0^n,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y}}} \frac{\beta_{z,D,p,0}}{\sqrt{2^n}} \sum_{y_1\in\mathcal{Y}} |c,z,D\cup(x,y_1),p\rangle \ .
\end{aligned}$$

Recall that the database register D is an internal register of CStO. Thus before querying the O (or O^B) oracle, except for CStO, $\hat{A}^{O,|H\rangle}$ and B did not perform any operation on D. According to [55], $|\Psi\rangle$ does not have the component $|\Psi_3\rangle$. Hence, $|\Psi\rangle$ can be rewritten as

$$|\Psi\rangle = |\Psi_1\rangle + |\Psi_2\rangle \ .$$

Denote the operation of O before performing the measurement M_{Z} as O_1, and the operation of O^B before performing the measurement M_{Z} as O_2, then

$$\begin{aligned}
O_1\,|\Psi_1\rangle &= \sum_{\substack{z=0,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \frac{\beta_{z,D,p}}{\sqrt{2^n}} F_x\left(\sum_{y_1\in\mathcal{Y}} |c,y_1,D\cup(x,y_1),p\rangle\right) \\
O_2\,|\Psi_1\rangle &= \sum_{\substack{z=0,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \beta_{z,D,p}\,|c,\perp,D,p\rangle \\
O_1\,|\Psi_2\rangle &= \sum_{\substack{z=0,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}} F_x\left(\sum_{y_1\in\mathcal{Y}} (-1)^{r\cdot y_1}\,|c,y_1,D\cup(x,y_1),p\rangle\right) \\
O_2\,|\Psi_2\rangle &= \sum_{\substack{z=0,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}} \sum_{y_1\in\mathcal{Y}} (-1)^{r\cdot y_1}\,|c,y_1,D\cup(x,y_1),p\rangle \ ,
\end{aligned}$$

where F_x is the decompression procedure in CStO applying on register D.

Therefore, after $\hat{A}^{O,|H\rangle}$ executes O_1 and performs the measurement M_{Z}, for any $y' \in \mathcal{Y}$, $|\Psi\rangle$ will collapse into the (un-normalized) state

$$|\Psi_{y'}^{\hat{A}}\rangle = \sum_{\substack{z=0,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \frac{\beta_{z,D,p}}{\sqrt{2^n}} F_x \left(|c, y', D \cup (x, y'), p\rangle\right) + \sum_{\substack{z=0,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}} F_x \left((-1)^{r\cdot y'} |c, y', D \cup (x, y'), p\rangle\right)$$

with probability $p_{y'}^{\hat{A}} = \left\| |\Psi_{y'}^{\hat{A}}\rangle \right\|^2$. It implies that for any $y' \in \mathcal{Y}$, the O oracle will respond with $g(y')$ with probability $p_{y'}^{\hat{A}}$.

For B, after executing O_2 and measuring M_{Z}, for any $y' \in \mathcal{Y}$, $|\Psi\rangle$ will collapse into the (un-normalized) state

$$|\Psi_{y'}^{B}\rangle = \sum_{\substack{z=0,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}} (-1)^{r\cdot y'} |c, y', D \cup (x, y'), p\rangle$$

with the *same* probability[6]

$$p_1^B = \left\| |\Psi_{y'}^{B}\rangle \right\|^2 ,$$

and will collapse into the (un-normalized) state

$$|\Psi_{\perp}^{B}\rangle = \sum_{\substack{z=0,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \beta_{z,D,p} |c, \perp, D, p\rangle$$

with the probability

$$p_2^B = \left\| |\Psi_{\perp}^{B}\rangle \right\|^2 .$$

Note that when Z is $\perp$, the result is set to $g(y^*)$, where $y^* \in \mathcal{Y}$ is uniformly and randomly chosen by B in the beginning, so for any $y' \in \mathcal{Y}$, $\Pr[y^* = y'] = 2^{-n}$. It implies that for any y', the probability of O^B returning $g(y')$ is

$$p^B = p_1^B + p_2^B/2^n .$$

Note that after the O oracle query, all the responses of H random oracle query on $x = f^{-1}(c)$ made by $\hat{A}^{O,|H\rangle}$ are

$$\begin{aligned} \mathsf{CStO} F_x |x, y, D \cup (x, y')\rangle &= F_x O_x F_x F_x |x, y, D \cup (x, y')\rangle \\ &= F_x |x, y \oplus y', D \cup (x, y')\rangle \ , \end{aligned}$$

which is equivalent to applying a unitary operator $U_{y'}$ to $|y\rangle$ such that $U_{y'}|y\rangle = |y \oplus y'\rangle$. Therefore, the H random oracle used by $\hat{A}^{O,|H\rangle}$ after the O oracle query is equivalent to being implemented by CStO^B defined in Theorem 4.1. Therefore, the execution of $\hat{A}^{O,|H\rangle}$ can be rewritten as

$$U_{q_H+2} \left(\prod_{i=i^*+1}^{q_H+1} \mathsf{CStO}^B \circ U_i \right) O \circ U_{i^*} \left(\prod_{i=1}^{i^*-1} \mathsf{CStO} \circ U_i \right) |\psi_0\rangle \ .$$

[6]Since the probability $\left\| |\Psi_{y'}^{B}\rangle \right\|^2$ has *same* value for any $y' \in \mathcal{Y}$, we denote this common value as p_1^B.

According to [10], when the event Ev is classical and well-defined, the probability of occurrence of the event is equivalent to the measurement of the density operator of the final state of $\hat{A}^{O,|H\rangle}$ or B with M_{Ev}. Recall that the state of $\hat{A}^{O,|H\rangle}$ after the O oracle query is

$$|\Psi^{\hat{A}}_{g(y')}\rangle = \frac{1}{\sqrt{p^{\hat{A}}_{y'}}}\left(\sum_{\substack{z=0,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \frac{\beta_{z,D,p}}{\sqrt{2^n}} F_x\left(|c, g(y'), D\cup(x,y'), p\rangle\right) + \sum_{\substack{z=0,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}} F_x\left((-1)^{r\cdot y'}|c, g(y'), D\cup(x,y'), p\rangle\right)\right)$$

with probability $p^{\hat{A}}_{y'}$, and the state of B after the O^B oracle query is

$$|\Psi^{B}_{g(y'),1}\rangle = \frac{1}{\sqrt{p^B_1}} \sum_{\substack{z=0,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}}(-1)^{r\cdot y'}|c, g(y'), D\cup(x,y'), p\rangle$$

with probability p^{β}_1, or is

$$|\Psi^{B}_{g(y'),2}\rangle = \frac{1}{\sqrt{p^B_2}} \sum_{\substack{z=0,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \beta_{z,D,p}|c, g(y'), D, p\rangle$$

with probability $p^B_2/2^n$. Thus, let Q denote $M_{\mathsf{Ev}}U_{q_H+2}\left(\prod_{i=i^*+1}^{q_H+1}\mathsf{CStO}^B\circ U_i\right)$, then we have

$$\Pr[\mathsf{Ev}:\hat{A}^{O,|H\rangle}] = \sum_{y'} p^{\hat{A}}_{y'}\left\|Q|\Psi^{\hat{A}}_{g(y')}\rangle\right\|^2$$
$$\Pr[\mathsf{Ev}:B] = \sum_{y'}\left(\frac{p^B_2}{2^n}\left\|Q|\Psi^B_{g(y'),2}\rangle\right\|^2 + p^B_1\left\|Q|\Psi^B_{g(y'),1}\rangle\right\|^2\right).$$

Let

$$|\Phi^1_{y'}\rangle = \sum_{\substack{z=0,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \beta_{z,D,p}F_x\left(|c, g(y'), D\cup(x,y'), p\rangle\right)$$
$$|\Phi^2_{y'}\rangle \sum_{\substack{z=0,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}} F_x\left((-1)^{r\cdot y'}|c, g(y'), D\cup(x,y'), p\rangle\right)$$
$$|\Phi^3_{y'}\rangle = \sum_{\substack{z=0,D,p,|D|<q_H+1,\\ x=f^{-1}(c),D(x)=\perp}} \beta_{z,D,p}\left(|c, g(y'), D\cup(x,y'), p\rangle\right)$$
$$|\Phi^4_{y'}\rangle \sum_{\substack{z=0,D,p,|D|<q_H,\\ x=f^{-1}(c),D(x)=\perp,r\in\mathcal{Y},r\neq 0}} \frac{\beta_{z,D,p,r}}{\sqrt{2^n}} \left((-1)^{r\cdot y'}|c, g(y'), D\cup(x,y'), p\rangle\right),$$

then for any $y' \in \mathcal{Y}$, we have

$$\begin{aligned}
\sqrt{p_{y'}^{\hat{A}}}\left\|Q\left|\Psi_{g(y')}^{A}\right\rangle\right\| &= \sqrt{p_{y'}^{\hat{A}}}\left\|Q\frac{1}{\sqrt{p_{y'}^{\hat{A}}}}\left(2^{-n/2}\left|\Phi_{y'}^{1}\right\rangle+\left|\Phi_{y'}^{2}\right\rangle\right)\right\| \\
&= \left\|Q\left(2^{-n/2}\left|\Phi_{y'}^{1}\right\rangle+\left|\Phi_{y'}^{2}\right\rangle\right)\right\| \\
&\leq \left\|Q2^{-n/2}\left|\Phi_{y'}^{1}\right\rangle\right\|+\left\|Q\left|\Phi_{y'}^{2}\right\rangle\right\| \\
&\overset{(*)}{=} 2^{-n/2}\left\|Q\left|\Phi_{y'}^{3}\right\rangle\right\|+\left\|Q\left|\Phi_{y'}^{4}\right\rangle\right\| \\
&= \sqrt{\frac{p_2^B}{2^n}}\left\|Q\frac{1}{\sqrt{p_2^B}}\left|\Phi_{y'}^{3}\right\rangle\right\|+\sqrt{p_1^B}\left\|Q\frac{1}{\sqrt{p_1^B}}\left|\Phi_{y'}^{4}\right\rangle\right\| \\
&= \sqrt{\frac{p_2^B}{2^n}}\left\|Q\left|\Psi_{g(y'),2}^{B}\right\rangle\right\|+\sqrt{p_1^B}\left\|Q\left|\Psi_{g(y'),1}^{B}\right\rangle\right\| ,
\end{aligned}$$

where equation $(*)$ utilizes the fact that unitary operators preserve the norm, and that the compression procedure F_x is unitary. Thus,

$$\begin{aligned}
\Pr[\mathsf{Ev} : \hat{A}^{O,|H\rangle}] &= \sum_{y'} p_{y'}^{\hat{A}}\left\|Q\left|\Psi_{g(y')}^{\hat{A}}\right\rangle\right\|^2 \\
&\leq \sum_{y'}\left(\sqrt{\frac{p_2^B}{2^n}}\left\|Q\left|\Psi_{g(y'),2}^{B}\right\rangle\right\|+\sqrt{p_1^B}\left\|Q\left|\Psi_{g(y'),1}^{B}\right\rangle\right\|\right)^2 \\
&\overset{(*)}{\leq} 2\sum_{y'}\left(\frac{p_2^B}{2^n}\left\|Q\left|\Psi_{g(y'),2}^{B}\right\rangle\right\|^2+p_1^B\left\|Q\left|\Psi_{g(y'),1}^{B}\right\rangle\right\|^2\right) \\
&= 2\Pr[\mathsf{Ev} : B] ,
\end{aligned} \tag{12}$$

where $(*)$ uses the Jensen's inequality. Combining equations (11) and (12), yields (10).

This completes the proof of Theorem 4.1. □

4.2 The Security of T_{RH} in the QROM

The security of T_{RH} in the QROM is captured in the following theorem.

Theorem 4.2 (The security of T_{RH} in the QROM)**.** *Assume $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ is modeled as a quantum-accessible random oracle. If* PKE' *is a rigid deterministic public-key encryption scheme that is δ-correct and* OW-CPA *secure, then* KEM_{RH} *is* IND-1CCA *secure.*

In particular, for any PPT *adversary $\mathcal{A}$ that attacks the* IND-1CCA *security of* KEM_{RH} *and has quantum access to the H random oracle, there exists a* PPT *adversary $\mathcal{B}$ that attacks the* OW-CPA *security of* PKE'*, such that*

$$\mathsf{Adv}_{\mathsf{KEM}_{RH}}^{\text{IND-1CCA}}(\mathcal{A}) \leq 4\sqrt{\mathsf{Adv}_{\mathsf{PKE}'}^{\text{OW-CPA}}(\mathcal{B})} + 2\delta .$$

Theorem 4.2. For $j = 0, \cdots, 3$, we define G_j to be the game played between the adversary $\mathcal{A}$ and the challenger as shown in Fig. 6, where $\mathcal{A}$ can make any number of quantum H random oracle queries, but at most one classical decapsulation oracle query. In each game, b is a random bit chosen by the challenger, while $\hat{b}$ is the bit output by $\mathcal{A}$ at the end of the game. We define W_j to be the event that $\hat{b} = b$ in game G_j.

Games G_0 to G_3:

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}'(1^\lambda),\ m^* \leftarrow\!\!\$\ \mathcal{M}$
$c^* := \mathsf{Enc}'(\mathsf{pk}, m^*),\ \mathsf{H} \leftarrow\!\!\$\ \Omega_H$
$k^*,\ k_1 \leftarrow\!\!\$\ \mathcal{K}$
$k_0 := \mathsf{H}(m^*, c^*)$ // $G_0 - G_1$
$k_0 := k^*$ // $G_2 - G_3$
$b \leftarrow\!\!\$\ \{0,1\},\ \mathsf{cnt} := 0$
if COLL_1 **then** // $G_1 - G_3$
  return $\perp$ // $G_1 - G_3$
$\hat{b} \leftarrow\!\!\$\ \mathcal{A}^{O_{\mathsf{Dec}}, |H\rangle}(\mathsf{pk}, c^*, k_b)$
return $[b = \hat{b}]$

Random Oracle $H(m, c)$:

if $(m, c) = (m^*, c^*)$ **then** // G_3
  return k^* // G_3
return $\mathsf{H}(m, c)$

Decapsulation Oracle $O_{\mathsf{Dec}}(c \neq c^*)$:

if $\mathsf{cnt} = 0$ **then**
  $\mathsf{cnt} := \mathsf{cnt} + 1$
  if COLL_2 **then** // $G_1 - G_3$
    return $\perp$ // $G_1 - G_3$
  $m' := \mathsf{Dec}'(\mathsf{sk}, c)$
  if $m' = \perp$ **then**
    return $k' := H(\star, c)$
  return $k' := H(m', c)$
return $k' := \perp$

Figure 6: Games G_0 to G_3 for the proof of Theorem 4.2.

Game G_0. In this game, the challenger randomly chooses a function H from the set Ω_H of functions $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ to respond to the H random oracle queries made by $\mathcal{A}$. Note that although the challenger chooses a random $k^* \leftarrow\!\!\$\ \mathcal{K}$ in the initialization step, it is not used in subsequent processes. Therefore, the behavior of the challenger is exactly consistent with that in the IND-1CCA game of $\mathsf{KEM}_{RH} := T_{RH}[\mathsf{PKE}', H]$. Thus, we have

$$|\Pr[W_0] - 1/2| = \mathsf{Adv}^{\text{IND-1CCA}}_{\mathsf{KEM}_{RH}}(\mathcal{A}) \ . \tag{13}$$

Game G_1. In this game, similar to game G_1 described in the proof of Theorem 3.1, let COLL_1 (or COLL_2) represent the event of *a collision* occurring in the challenge encapsulation c^* received by $\mathcal{A}$ (or the decapsulation oracle query c issued by $\mathcal{A}$). We assume that neither COLL_1 nor COLL_2 occurs in this game. The probability of either occurring is no greater than δ since PKE' is δ-correct. Thus, we have

$$|\Pr[W_1] - \Pr[W_0]| \leq 2\delta \ . \tag{14}$$

Game G_2. This game is the same as game G_1, except that $k_0 := \mathsf{H}(m^*, c^*)$ in the initialization step is replaced by $k_0 := k^*$. Since k_0 and k_1 are both randomly chosen from $\mathcal{K}^*$, and are not used in any oracles, b is independent of $\mathcal{A}$'s view. Therefore,

$$\Pr[W_2] = 1/2 \ . \tag{15}$$

Game G_3. This game modifies the H random oracle as follows: upon receiving a query where $(m, c) = (m^*, c^*)$, it returns k^*. At this point, the H random oracle is simulated by a new function $\mathsf{G} : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$: for all $(m, c) \neq (m^*, c^*)$, $\mathsf{G}(m, c) = \mathsf{H}(m, c)$; but when $(m, c) = (m^*, c^*)$, $\mathsf{G}(m^*, c^*) = k^*$ is random and independent of $\mathsf{H}(m^*, c^*)$. Since $k_0 = k^* = \mathsf{G}(m^*, c^*)$, the behavior of the challenger is equivalent to that in game G_1. Thus, we have

$$\Pr[W_3] = \Pr[W_1] \ . \tag{16}$$

Recall that $\mathsf{G}(m,c) = \mathsf{H}(m,c)$ for all $(m,c) \neq (m^*,c^*)$ and $\mathcal{A}$ is not allowed to make decapsulation oracle queries on c^*. Therefore, the decapsulation oracle O_{Dec} in game G_2 is identical to that in game G_3. Next, we can construct two oracle algorithms $A^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}(z)$ and $A^{O_{\mathsf{Dec}},|\mathsf{G}\rangle}(z)$ to execute games G_2 and G_3 respectively, where $z = (\mathsf{pk}, c^*, k_0)$, $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}'(\lambda)$, $m^* \leftarrow\$ \mathcal{M}$, $c^* := \mathsf{Enc}'(\mathsf{pk}, m^*), k^* \leftarrow\$ \mathcal{K}, k_0 := k^*$. $A^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}(z)$ (or $A^{O_{\mathsf{Dec}},|\mathsf{G}\rangle}(z)$) first computes $k_1 \leftarrow\$ \mathcal{K}, b \leftarrow\$ \{0,1\}$, then runs the adversary $\mathcal{A}^{O_{\mathsf{Dec}},|H\rangle}(\mathsf{pk}, c^*, k_b)$ in game G_2 (or game G_3) to obtain $\hat{b}$, and finally outputs $[b = \hat{b}]$, where H (or G) is used to simulate the H random oracle in game G_2 (or game G_3). Note that we still assume that the events COLL_1 and COLL_2 defined in game G_1 do not occur. Therefore,

$$
\begin{aligned}
\Pr[1 \leftarrow A^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}(z)] &= \Pr[W_2] \\
\Pr[1 \leftarrow A^{O_{\mathsf{Dec}},|\mathsf{G}\rangle}(z)] &= \Pr[W_3]\ ,
\end{aligned} \tag{17}
$$

where $z = (\mathsf{pk}, c^*, k_0)$, $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}'(\lambda)$, $m^* \leftarrow\$ \mathcal{M}$, $c^* := \mathsf{Enc}'(\mathsf{pk}, m^*)$, $k^* \leftarrow\$ \mathcal{K}$, $k_0 := k^*$. Since H and G only differ at the point (m^*, c^*), according to Lemma 2.1, there exists an oracle algorithm $B^{O_{\mathsf{Dec}},|\mathsf{H}\rangle,|\mathsf{G}\rangle}(z)$ such that[7]

$$
\begin{aligned}
&\left|\Pr[1 \leftarrow A^{O_{\mathsf{Dec}},|\mathsf{G}\rangle}(z)] - \Pr[1 \leftarrow A^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}(z)]\right| \\
&\qquad\qquad\qquad \leq 2\sqrt{\Pr[(m^*,c^*) \leftarrow B^{O_{\mathsf{Dec}},|\mathsf{G}\rangle,|\mathsf{H}\rangle}(z)]}\ .
\end{aligned} \tag{18}
$$

Then for $j = 4, \ldots, 8$, we define G_j played between the oracle algorithm and the challenger as shown in Fig. 7. In each game, m^* is randomly chosen from $\mathcal{M}$, while $\hat{m}$ is output by the oracle

Games G_4 to G_8:

$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}'(1^\lambda),\ m^* \leftarrow\$ \mathcal{M}$
$c^* := \mathsf{Enc}'(\mathsf{pk}, m^*),\ \mathsf{H} \leftarrow\$ \Omega_H$
$k^*,\ k_1 \leftarrow\$ \mathcal{K},\ k_0 := k^*$
$b \leftarrow\$ \{0,1\},\ \mathsf{cnt} := 0$
if COLL_1 **then**
  return $\perp$
$\mathsf{flag} \leftarrow\$ \{0,1\}$ // $G_5 - G_8$
$(\hat{m}, \hat{c}) \leftarrow B^{O_{\mathsf{Dec}},|\mathsf{H}\rangle,|\mathsf{G}\rangle}(\mathsf{pk}, c^*, k_0)$ // $G_4 - G_6$
$(\hat{m}, \hat{c}) \leftarrow \bar{B}^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}(\mathsf{pk}, c^*, k_0)$ // G_7
$(\hat{m}, \hat{c}) \leftarrow \hat{B}(\mathsf{pk}, c^*, k_0, \mathsf{flag})$ // G_8
return $[m^* = \hat{m}]$

Random Oracle $\mathsf{H}(m,c)$:

return $\mathsf{H}(m,c)$

Decapsulation Oracle $O_{\mathsf{Dec}}(c \neq c^*)$:

if $\mathsf{cnt} = 0$ **then**
  $\mathsf{cnt} := \mathsf{cnt} + 1$
  if COLL_2 **then**
    return $\perp$
  $m' := \mathsf{Dec}'(\mathsf{sk}, c)$
  if $m' = \perp$ **then** // G_4
  if $\mathsf{flag} = 0$ **then** // $G_5 - G_8$
    return $k' := H(\star, c)$
  return $k' := H(m', c)$
return $k' := \perp$

Random Oracle $\mathsf{G}(m,c)$: // $G_4 - G_6$

if $(m,c) = (m^*,c^*)$ **then** // $G_4 - G_5$
if $c = c^* \wedge \mathsf{Enc}'(\mathsf{pk}, m) = c^*$ **then** // G_6
  return k^*
return $\mathsf{H}(m,c)$

Figure 7: Games G_4 to G_8 for the proof of Theorem 4.2.

algorithm at the end of the game. We define the event that $\hat{m} = m^*$ as Z_j in game G_j.

[7]Since the decapsulation oracle O_{Dec} in game G_2 is identical to that in game G_3, therefore it can be seen as an internal oracle of the oracle algorithm A.

Game G_4. This game is defined in Fig. 7. It is obvious that

$$\Pr[(m^*, c^*) \leftarrow B^{O_{\mathsf{Dec}},|\mathsf{G}\rangle,|\mathsf{H}\rangle}(z)] \leq \Pr[Z_4] \ , \tag{19}$$

where $z = (\mathsf{pk}, c^*, k_0)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}'(\lambda)$, $m^* \leftarrow\!\!\$\, \mathcal{M}$, $c^* := \mathsf{Enc}'(\mathsf{pk}, m^*)$, $k^* \leftarrow\!\!\$\, \mathcal{K}$, $k_0 := k^*$.

Game G_5. This game is the same as game G_4, except that the challenger chooses an extra random bit $\mathsf{flag} \leftarrow\!\!\$\, \{0,1\}$ in the initialization step, and replaces the condition $m' = \perp$ by $\mathsf{flag} = 0$ in the decapsulation oracle. Similar to the analysis of game G_3 in the proof of Theorem 3.1, we have

$$\Pr[Z_4] = \Pr[Z_5 \wedge m' = \perp | \mathsf{flag} = 0] + \Pr[Z_5 \wedge m' \neq \perp | \mathsf{flag} = 1] \ . \tag{20}$$

Game G_6. This game replaces the condition $(m, c) = (m^*, c^*)$ by $c = c^* \wedge \mathsf{Enc}'(\mathsf{pk}, m) = c^*$ in the G random oracle of game G_5. Recall that since game G_1 we have assumed that the event COLL_1 would not occur, which implies that these two conditions are equivalent. Therefore,

$$\begin{aligned} \Pr[Z_6 \wedge m' = \perp | \mathsf{flag} = 0] &= \Pr[Z_5 \wedge m' = \perp | \mathsf{flag} = 0] \\ \Pr[Z_6 \wedge m' \neq \perp | \mathsf{flag} = 1] &= \Pr[Z_5 \wedge m' \neq \perp | \mathsf{flag} = 1] \ . \end{aligned} \tag{21}$$

Note that at this point, the G random oracle does not depend on the knowledge of m^*, so it can be simulated with only access to the H oracle. Therefore, we can construct a new oracle algorithm $\bar{B}^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}(\mathsf{pk}, c^*, k_b)$, which is the same as $B^{O_{\mathsf{Dec}},|\mathsf{H}\rangle,|\mathsf{G}\rangle}$ except that if it needs to query the G oracle, it accesses H as the same way as the G random oracle in game G_6 and responds with the corresponding result.

Game G_7. This game replaces the oracle algorithm $B^{O_{\mathsf{Dec}},|\mathsf{H}\rangle,|\mathsf{G}\rangle}$ by $\bar{B}^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}$ in game G_6. By the above analysis, we have

$$\begin{aligned} \Pr[Z_7 \wedge m' = \perp | \mathsf{flag} = 0] &= \Pr[Z_6 \wedge m' = \perp | \mathsf{flag} = 0] \\ \Pr[Z_7 \wedge m' \neq \perp | \mathsf{flag} = 1] &= \Pr[Z_6 \wedge m' \neq \perp | \mathsf{flag} = 1] \ . \end{aligned} \tag{22}$$

Denote two functions $f : \mathcal{M} \times \mathcal{C} \to \mathcal{C} \cup \perp$ and $g : \mathcal{K} \to \mathcal{K}$ as follows. If $\mathsf{flag} = 0$, the challenger sets

$$f(m, c) := \begin{cases} c & \text{if } m = \star \\ \perp & \text{otherwise} \ ; \end{cases}$$

while if $\mathsf{flag} = 1$, the challenger sets

$$f(m, c) := \begin{cases} c & \text{if } \mathsf{Enc}'(\mathsf{pk}, m) = c \\ \perp & \text{otherwise} \ . \end{cases}$$

Let g be an identity function, i.e., $g(k) = k$ for all $k \in \mathcal{K}$. It is obvious that for $c \neq c^*$, if $m' = \perp$ when $\mathsf{flag} = 0$, or if $m' \neq \perp$ when $\mathsf{flag} = 1$, the process of O_{Dec} is equivalent to computing $(m, c) := f^{-1}(c)$, (classically) accessing the H random oracle to obtain $k := \mathsf{H}(m, c)$, and returning $g(k)$, where (m, c) is the unique preimage of c under f. Then by Theorem 4.1, there exists a new algorithm $\hat{B}$ that only needs to know how to calculate f and g, such that

$$2 \Pr[\mathsf{Ev} : \hat{B}(z, \mathsf{flag}) | m' = \perp \wedge \mathsf{flag} = 0] \geq \Pr[\mathsf{Ev} : \bar{B}^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}(z) | m' = \perp \wedge \mathsf{flag} = 0]$$
$$2 \Pr[\mathsf{Ev} : \hat{B}(z, \mathsf{flag}) | m' \neq \perp \wedge \mathsf{flag} = 1] \geq \Pr[\mathsf{Ev} : \bar{B}^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}(z) | m' \neq \perp \wedge \mathsf{flag} = 1],$$

for any classical event Ev, where $z = (\mathsf{pk}, c^*, k_0)$[8].

[8] The extra input flag for $\hat{B}$ is used to determine which f should be used.

Game G_8. This game replaces the oracle algorithm $\bar{B}^{O_{\mathsf{Dec}},|\mathsf{H}\rangle}$ by $\hat{B}$ in game G_7. By the above analysis, we have

$$2\Pr[Z_8|m' = \perp \wedge \mathsf{flag} = 0] \geq \Pr[Z_7|m' = \perp \wedge \mathsf{flag} = 0]$$
$$2\Pr[Z_8|m' \neq \perp \wedge \mathsf{flag} = 1] \geq \Pr[Z_7|m' \neq \perp \wedge \mathsf{flag} = 1] \ .$$

Since the event $m' = \perp$ is independent of the event $\mathsf{flag} = 0$, we have

$$\begin{aligned}\Pr[\mathsf{Ev}|m' = \perp \wedge \mathsf{flag} = 0] &= \frac{\Pr[\mathsf{Ev} \wedge m' = \perp \wedge \mathsf{flag} = 0]}{\Pr[m' = \perp \wedge \mathsf{flag} = 0]} \\ = \frac{\Pr[\mathsf{Ev} \wedge m' = \perp \wedge \mathsf{flag} = 0]}{\Pr[m' = \perp]\Pr[\mathsf{flag} = 0]} &= \frac{\Pr[\mathsf{Ev} \wedge m' = \perp|\mathsf{flag} = 0]}{\Pr[m' = \perp]}\end{aligned}$$

for any classic event Ev. Therefore, we obtain

$$\begin{aligned}2\Pr[Z_8 \wedge m' = \perp|\mathsf{flag} = 0] &= 2\Pr[Z_8|m' = \perp \wedge \mathsf{flag} = 0]\Pr[m' = \perp] \\ &\geq \Pr[Z_7|m' = \perp \wedge \mathsf{flag} = 0]\Pr[m' = \perp] \\ &= \Pr[Z_7 \wedge m' = \perp|\mathsf{flag} = 0] \ .\end{aligned} \tag{23}$$

Similarly, we can obtain

$$2\Pr[Z_8 \wedge m' \neq \perp|\mathsf{flag} = 1] \geq \Pr[Z_7 \wedge m' \neq \perp|\mathsf{flag} = 1] \ . \tag{24}$$

Combining (20)-(24), we obtain

$$\begin{aligned}\Pr[Z_8] &\geq \Pr[Z_8 \wedge m' = \perp \wedge \mathsf{flag} = 0] + \Pr[Z_8 \wedge m' \neq \perp \wedge \mathsf{flag} = 1] \\ &= \frac{1}{2}\left(\Pr[Z_8 \wedge m' = \perp|\mathsf{flag} = 0] + \Pr[Z_8 \wedge m' \neq \perp|\mathsf{flag} = 1]\right) \\ &\geq \frac{1}{4}\left(\Pr[Z_7 \wedge m' = \perp|\mathsf{flag} = 0] + \Pr[Z_7 \wedge m' \neq \perp|\mathsf{flag} = 1]\right) \\ &= \frac{1}{4}\left(\Pr[Z_6 \wedge m' = \perp|\mathsf{flag} = 0] + \Pr[Z_6 \wedge m' \neq \perp|\mathsf{flag} = 1]\right) \\ &= \frac{1}{4}\left(\Pr[Z_5 \wedge m' = \perp|\mathsf{flag} = 0] + \Pr[Z_5 \wedge m' \neq \perp|\mathsf{flag} = 1]\right) \\ &= \frac{1}{4}\Pr[Z_4] \ .\end{aligned} \tag{25}$$

At this point, we can find that sk is useless in game G_8. Therefore, if the event Z_8 occurs, we can construct an adversary $\mathcal{B}$ to attack the OW-CPA security of PKE' as follows: Upon receiving the public key pk and the challenge ciphertext c^* from the OW-CPA game of PKE', $\mathcal{B}$ randomly chooses $k_0 \leftarrow\!\!\$\ \mathcal{K}$ and $\mathsf{flag} \leftarrow\!\!\$\ \{0,1\}$, and uses $(\mathsf{pk}, c^*, k_0, \mathsf{flag})$ as input to run $\hat{B}$. When the game ends, $\mathcal{B}$ outputs $\hat{m}$ that outputed by $\hat{B}$. Therefore,

$$\Pr[Z_8] \leq \mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}'}(\mathcal{B}) \ . \tag{26}$$

Combining (13)-(19) and (25)-(26), we obtain

$$\mathsf{Adv}^{\text{IND-1CCA}}_{\mathsf{KEM}_{RH}}(\mathcal{A}) \leq 4\sqrt{\mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}'}(\mathcal{B}')} + 2\delta \ .$$

That completes the proof of the theorem. □

Remark 4.1. The bound given by Jiang et al. [33] in this case is

$$\mathsf{Adv}^{\text{IND-1CCA}}_{\mathsf{KEM}_{RH}}(\mathcal{A}) \leq 6(q_H + 1)\sqrt{\mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}'}(\mathcal{B}) + 1/|\mathcal{K}| + \delta}\ ,$$

where q_H is the number of H random oracle queries made by $\mathcal{A}$. Despite the unavoidable quadratic reduction loss [37], our reduction is also *tight* in the QROM, with a loss factor of $\mathcal{O}(1)$.

Remark 4.2. The security proof technique used by Jiang et al. [33] is called (single-classical-query) measure-and-reprogram lemma, which is first proposed by Don et al. [18, 19] and then is extended by Jiang et al. [33]. In this technique, to simulate the decapsulation oracle without sk, the basic strategy adopted by the challenger is to randomly choose one of the q random oracle queries made by $\mathcal{A}$, measure its input register, consider it as the point that needs reprogramming, and use the reprogrammed random oracle to respond to subsequent random oracle queries. This analysis method needs to consider the impact of different measurements at different times on the final state of $\mathcal{A}$, and ultimately derives an upper bound for the norm of the final state of $\mathcal{A}$, which is a sum of approximately q terms. When considering probability, it is necessary to square this upper bound, and when using Jensen's inequality to relate the probability in a specific case, a coefficient of $\mathcal{O}(q^2)$ will be generated. Therefore, using this technique in security proofs can introduce a loss factor related to q. However, the strategy adopted here is similar to that in the proof in the ROM, where the random oracle and decapsulation oracle are modified during the execution of $\mathcal{A}$, resulting in a loss factor of only $\mathcal{O}(1)$ for the derived bound. Thus, using our proposed new technique for security proofs will not introduce an additional loss factor exceeding $\mathcal{O}(1)$.

Acknowledgments We would like to thank all anonymous reviewers for their valuable comments. This work is supported in part by the National Natural Science Foundation of China (Grant No.62122092, No.62202485, No.62032005, No.62425205).

References

[1] Alagic, G., Majenz, C., Russell, A., Song, F.: Quantum-access-secure message authentication via blind-unforgeability. In: EUROCRYPT (3). Lecture Notes in Computer Science, vol. 12107, pp. 788–817. Springer (2020), https://doi.org/10.1007/978-3-030-45727-3_27

[2] Alkim, E., Bos, J.W., Ducas, L., Longa, P., Mironov, I., Naehrig, M., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: Frodokem: Learning with errors key encapsulation (2021), https://frodokem.org/files/FrodoKEM-specification-20210604.pdf

[3] Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 11693, pp. 269–295. Springer (2019), https://doi.org/10.1007/978-3-030-26951-7_10

[4] Ananth, P., Kaleoglu, F., Li, X., Liu, Q., Zhandry, M.: On the feasibility of unclonable encryption, and more. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 13508, pp. 212–241. Springer (2022), https://doi.org/10.1007/978-3-031-15979-4_8

[5] Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: EUROCRYPT (2). Lecture Notes in Computer Science, vol. 9666, pp. 273–304. Springer (2016), https://doi.org/10.1007/978-3-662-49896-5_10

[6] Bartusek, J., Malavolta, G.: Indistinguishability obfuscation of null quantum circuits and applications. In: ITCS. LIPIcs, vol. 215, pp. 15:1–15:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022), https://doi.org/10.4230/LIPICS.ITCS.2022.15

[7] Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS. pp. 62–73. ACM (1993), https://doi.org/10.1145/168588.168596

[8] Bernstein, D.J., Persichetti, E.: Towards KEM unification. IACR Cryptol. ePrint Arch. p. 526 (2018), https://eprint.iacr.org/2018/526

[9] Beullens, W., Faugère, J., Koussa, E., Macario-Rat, G., Patarin, J., Perret, L.: Pkp-based signature scheme. In: INDOCRYPT. Lecture Notes in Computer Science, vol. 11898, pp. 3–22. Springer (2019), https://doi.org/10.1007/978-3-030-35423-7_1

[10] Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of CCA security in the quantum random oracle model. In: TCC (2). Lecture Notes in Computer Science, vol. 11892, pp. 61–90. Springer (2019), https://doi.org/10.1007/978-3-030-36033-7_3

[11] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer (2011), https://doi.org/10.1007/978-3-642-25385-0_3

[12] Boneh, D., Shoup, V.: A Graduate Course in Applied Cryptography (2023), http://toc.cryptobook.us/

[13] Bos, J.W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In: EuroS&P. pp. 353–367. IEEE (2018), https://doi.org/10.1109/EUROSP.2018.00032

[14] Brendel, J., Fiedler, R., Günther, F., Janson, C., Stebila, D.: Post-quantum asynchronous deniable key exchange and the signal handshake. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 13178, pp. 3–34. Springer (2022), https://doi.org/10.1007/978-3-030-97131-1_1

[15] Chia, N., Chung, K., Yamakawa, T.: Classical verification of quantum computations with efficient verifier. In: TCC (3). Lecture Notes in Computer Science, vol. 12552, pp. 181–206. Springer (2020), https://doi.org/10.1007/978-3-030-64381-2_7

[16] Cini, V., Ramacher, S., Slamanig, D., Striecks, C.: Cca-secure (puncturable) kems from encryption with non-negligible decryption errors. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 12491, pp. 159–190. Springer (2020), https://doi.org/10.1007/978-3-030-64837-4_6

[17] Coron, J.: Optimal security proofs for PSS and other signature schemes. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 2332, pp. 272–287. Springer (2002), https://doi.org/10.1007/3-540-46035-7_18

[18] Don, J., Fehr, S., Majenz, C.: The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 12172, pp. 602–631. Springer (2020), https://doi.org/10.1007/978-3-030-56877-1_21

[19] Don, J., Fehr, S., Majenz, C., Schaffner, C.: Security of the fiat-shamir transformation in the quantum random-oracle model. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 11693, pp. 356–383. Springer (2019), https://doi.org/10.1007/978-3-030-26951-7_13

[20] Don, J., Fehr, S., Majenz, C., Schaffner, C.: Online-extractability in the quantum random-oracle model. In: EUROCRYPT (3). Lecture Notes in Computer Science, vol. 13277, pp. 677–706. Springer (2022), https://doi.org/10.1007/978-3-031-07082-2_24

[21] Dowling, B., Fischlin, M., Günther, F., Stebila, D.: A cryptographic analysis of the TLS 1.3 handshake protocol. J. Cryptol. **34**(4), 37 (2021), https://doi.org/10.1007/S00145-021-09384-1

[22] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: CRYPTO. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer (1999), https://doi.org/10.1007/3-540-48405-1_34

[23] 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

[24] Galbraith, S.D., Petit, C., Silva, J.: Identification protocols and signature schemes based on supersingular isogeny problems. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 10624, pp. 3–33. Springer (2017), https://doi.org/10.1007/978-3-319-70694-8_1

[25] Ge, J., Shan, T., Xue, R.: Tighter qcca-secure key encapsulation mechanism with explicit rejection in the quantum random oracle model. In: CRYPTO (5). Lecture Notes in Computer Science, vol. 14085, pp. 292–324. Springer (2023), https://doi.org/10.1007/978-3-031-38554-4_10

[26] Grover, L.K.: Quantum mechanics helps in searching for a needle in a haystack. Phys. Rev. Lett. **79**, 325–328 (Jul 1997), https://doi.org/10.1103/PhysRevLett.79.325

[27] Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki-okamoto transformation. In: TCC (1). Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer (2017), https://doi.org/10.1007/978-3-319-70500-2_12

[28] Hofheinz, D., Jager, T., Knapp, E.: Waters signatures with optimal security reduction. In: Public Key Cryptography. Lecture Notes in Computer Science, vol. 7293, pp. 66–83. Springer (2012), https://doi.org/10.1007/978-3-642-30057-8_5

[29] Hosoyamada, A., Iwata, T.: On tight quantum security of HMAC and NMAC in the quantum random oracle model. In: CRYPTO (1). Lecture Notes in Computer Science, vol. 12825, pp. 585–615. Springer (2021), https://doi.org/10.1007/978-3-030-84242-0_21

[30] Hövelmanns, K., Kiltz, E., Schäge, S., Unruh, D.: Generic authenticated key exchange in the quantum random oracle model. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 12111, pp. 389–422. Springer (2020), https://doi.org/10.1007/978-3-030-45388-6_14

[31] Huguenin-Dumittan, L., Vaudenay, S.: On ind-qcca security in the ROM and its applications - CPA security is sufficient for TLS 1.3. In: EUROCRYPT (3). Lecture Notes in Computer Science, vol. 13277, pp. 613–642. Springer (2022), https://doi.org/10.1007/978-3-031-07082-2_22

[32] Jaeger, J., Song, F., Tessaro, S.: Quantum key-length extension. In: TCC (1). Lecture Notes in Computer Science, vol. 13042, pp. 209–239. Springer (2021), https://doi.org/10.1007/978-3-030-90459-3_8

[33] Jiang, H., Ma, Z., Zhang, Z.: Post-quantum security of key encapsulation mechanism against CCA attacks with a single decapsulation query. In: ASIACRYPT (4). Lecture Notes in Computer Science, vol. 14441, pp. 434–468. Springer (2023), https://doi.org/10.1007/978-981-99-8730-6_14

[34] Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 10993, pp. 96–125. Springer (2018), https://doi.org/10.1007/978-3-319-96878-0_4

[35] Jiang, H., Zhang, Z., Ma, Z.: Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 11443, pp. 618–645. Springer (2019), https://doi.org/10.1007/978-3-030-17259-6_21

[36] Jiang, H., Zhang, Z., Ma, Z.: Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In: PQCrypto. Lecture Notes in Computer Science, vol. 11505, pp. 227–248. Springer (2019), https://doi.org/10.1007/978-3-030-25510-7_13

[37] 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: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 13090, pp. 487–517. Springer (2021), https://doi.org/10.1007/978-3-030-92062-3_17

[38] Katsumata, S., Yamada, S., Yamakawa, T.: Tighter security proofs for GPV-IBE in the quantum random oracle model. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 11273, pp. 253–282. Springer (2018), https://doi.org/10.1007/978-3-030-03329-3_9

[39] Kitagawa, F., Nishimaki, R.: KDM security for the fujisaki-okamoto transformations in the QROM. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 13178, pp. 286–315. Springer (2022), https://doi.org/10.1007/978-3-030-97131-1_10

[40] 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: EUROCRYPT (3). Lecture Notes in Computer Science, vol. 12107, pp. 703–728. Springer (2020), https://doi.org/10.1007/978-3-030-45727-3_24

[41] Lyu, Y., Liu, S.: Two-message authenticated key exchange from public-key encryption. In: ESORICS (1). Lecture Notes in Computer Science, vol. 14344, pp. 414–434. Springer (2023), https://doi.org/10.1007/978-3-031-50594-2_21

[42] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information (10th Anniversary edition). Cambridge University Press (2016), https://www.cambridge.org/de/academic/subjects/physics/quantum-physics-quantum-information-and-quantum-computation/quantum-computation-and-quantum-information-10th-anniversary-edition?format=HB

[43] Okamoto, T., Pointcheval, D.: REACT: rapid enhanced-security asymmetric cryptosystem transform. In: CT-RSA. Lecture Notes in Computer Science, vol. 2020, pp. 159–175. Springer (2001), https://doi.org/10.1007/3-540-45353-9_13

[44] Pan, J., Wagner, B., Zeng, R.: Tighter security for generic authenticated key exchange in the QROM. In: ASIACRYPT (4). Lecture Notes in Computer Science, vol. 14441, pp. 401–433. Springer (2023), https://doi.org/10.1007/978-981-99-8730-6_13

[45] Pan, J., Zeng, R.: Selective opening security in the quantum random oracle model, revisited. In: Public Key Cryptography (3). Lecture Notes in Computer Science, vol. 14603, pp. 92–122. Springer (2024), https://doi.org/10.1007/978-3-031-57725-3_4

[46] Sageloli, É., Pébereau, P., Méaux, P., Chevalier, C.: Shorter and faster identity-based signatures with tight security in the (Q)ROM from lattices. In: ACNS (1). Lecture Notes in Computer Science, vol. 13905, pp. 634–663. Springer (2023), https://doi.org/10.1007/978-3-031-33488-7_24

[47] Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: EUROCRYPT (3). Lecture Notes in Computer Science, vol. 10822, pp. 520–551. Springer (2018), https://doi.org/10.1007/978-3-319-78372-7_17

[48] Shan, T., Ge, J., Xue, R.: Qcca-secure generic transformations in the quantum random oracle model. In: Public Key Cryptography (1). Lecture Notes in Computer Science, vol. 13940, pp. 36–64. Springer (2023), https://doi.org/10.1007/978-3-031-31368-4_2

[49] Tanaka, Y., Ueno, R., Xagawa, K., Ito, A., Takahashi, J., Homma, N.: Multiple-valued plaintext-checking side-channel attacks on post-quantum kems. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2023**(3), 473–503 (2023), https://doi.org/10.46586/TCHES.V2023.I3.473-503

[50] Targhi, E.E., Unruh, D.: Post-quantum security of the fujisaki-okamoto and OAEP transforms. In: TCC (B2). Lecture Notes in Computer Science, vol. 9986, pp. 192–216 (2016), https://doi.org/10.1007/978-3-662-53644-5_8

[51] Unruh, D.: Quantum position verification in the random oracle model. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 8617, pp. 1–18. Springer (2014), https://doi.org/10.1007/978-3-662-44381-1_1

[52] Xagawa, K., Yamakawa, T.: (tightly) qcca-secure key-encapsulation mechanism in the quantum random oracle model. In: PQCrypto. Lecture Notes in Computer Science, vol. 11505, pp. 249–268. Springer (2019), https://doi.org/10.1007/978-3-030-25510-7_14

[53] Yamakawa, T., Zhandry, M.: Classical vs quantum random oracles. In: EUROCRYPT (2). Lecture Notes in Computer Science, vol. 12697, pp. 568–597. Springer (2021), https://doi.org/10.1007/978-3-030-77886-6_20

[54] Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: CRYPTO. Lecture Notes in Computer Science, vol. 7417, pp. 758–775. Springer (2012), https://doi.org/10.1007/978-3-642-32009-5_44

[55] Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 11693, pp. 239–268. Springer (2019), https://doi.org/10.1007/978-3-030-26951-7_9

[56] Zhang, J., Yu, Y., Feng, D., Fan, S., Zhang, Z.: On the (quantum) random oracle methodology: New separations and more. IACR Cryptol. ePrint Arch. p. 1101 (2019), https://eprint.iacr.org/2019/1101