QCCA-Secure Generic Transformations in the Quantum Random Oracle Model

Tianshu Shan[1,2][0000−0002−1918−7464], Jiangxia Ge[1,2][0000−0002−1671−7933], and Rui Xue[1,2(✉)][0000−0001−6024−3635]

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

Abstract. The post-quantum security of cryptographic schemes assumes that the quantum adversary only receives the classical result of computations with the secret key. Further, it is unknown whether the post-quantum secure schemes still remain secure if the adversary can obtain a superposition state of the results.
In this paper, we formalize one class of public-key encryption schemes named oracle-masked schemes. Then we define the plaintext extraction procedure for those schemes and this procedure simulates the quantum-accessible decryption oracle with a certain loss.
The construction of the plaintext extraction procedure does not need to take the secret key as input. Based on this property, we prove the IND-qCCA security of the Fujisaki-Okamoto (FO) transformation in the quantum random oracle model (QROM) and our security proof is tighter than the proof given by Zhandry (Crypto 2019). We also give the first IND-qCCA security proof of the REACT transformation in the QROM. Furthermore, our formalization can be applied to prove the IND-qCCA security of key encapsulation mechanisms with explicit rejection. As an example, we present the IND-qCCA security proof of T_{CH} transformation, proposed by Huguenin-Dumittan and Vaudenay (Eurocrypt 2022), in the QROM.

Keywords: FO transformation · REACT transformation · quantum random oracle model · quantum chosen ciphertext attack.

1 Introduction

There are two criteria for a practical encryption scheme: security and efficiency. Many generic transformations are proposed to enhance the security of public-key encryption schemes (PKEs) to achieve the indistinguishable under chosen ciphertext attacks (IND-CCA) security [2,8,11,23]. As for efficiency, Cramer and Shoup proposed the KEM-DEM hybrid construction that combines an IND-CCA key encapsulation mechanism (KEM) with a one-time chosen ciphertext secure secret-key encryption scheme (SKE) to obtain an IND-CCA PKE [9].

Cryptographic schemes often have efficient constructions in the random oracle model (ROM) [2], in which schemes are proven to be secure assuming the existence of the publicly accessible random oracle. Many generic transforms are relative to random oracles. For instance, the Fujisaki-Okamoto (FO) transformation turns an arbitrary PKE that is one-way under chosen plaintext attacks (OW-CPA) into an IND-CCA PKE in the ROM [11], and the REACT transformation turns an arbitrary PKE that is one-way under plaintext checking attacks (OW-PCA) into an IND-CCA PKE in the ROM [23].

Typically, the random oracle is instantiated with a cryptographic hash function. Thus in the real world attack, a quantum attacker can evaluate the hash function in superposition. To capture this issue, Boneh et al. [4] proposed the quantum random oracle model (QROM) where the quantum adversary can query the random oracle with superposition states. Further, classical schemes may be implemented on quantum computers, which potentially gives quantum attackers more power. For this case, Boneh and Zhandry [5] introduced the indistinguishability under quantum chosen ciphertext attacks (IND-qCCA) for encryption schemes, where the adversary can make quantum queries to the decryption oracle. Following it, Gagliardoni et al. [13] focused on SKE and proposed new notions of indistinguishability and semantic security in the quantum world, e.g. quantum semantic security under chosen plaintext attacks (qSEM-qCPA). On the other hand, Xagawa and Yamakawa [27] presented the IND-qCCA security of KEMs, where the adversary can query the decapsulation oracle in superposition.

Boneh et al. [4] summarized four proof techniques that are commonly used in the ROM but not appropriate to the quantum setting straightforwardly. "Extractability", as one of them, is that the simulator learns the preimages the adversary takes interest in when simulating the random oracle for the adversary.

Extractability is the core to simulate answers to decryption queries in the IND-CCA security proof for both FO and REACT in the ROM. However, in the quantum setting, the non-existence of this technique had been an obstacle to their security proofs in QROM. To circumvent it, Targhi and Unruh [26] and the follow-up work by Ambainis et al. [1] modified the FO transformation by appending an extra hash function to the ciphertext, then applied the One-way to Hiding (O2H) Theorem and its variant to prove the IND-CCA security of the modified FO in the QROM.

Hofheinz et al. [14] divided KEMs into two types: explicit rejection and implicit rejection. The explicit rejection (resp. implicit rejection) type returns a symbol $\perp$ (resp. a pseudorandom value) if the ciphertext is invalid. For both two types, they presented the IND-CCA security proof of transformations with additional hash in the QROM. Later, transformations with implicit rejection had been free from the additional hash and proved to be IND-CCA and even IND-qCCA in the QROM [3,17,19,20,21,24,27]. Nonetheless, for explicit rejection type, the IND-CCA security proofs in the QROM were only given for those transformations either with additional hash [18] or with non-standard security assumptions [19]. It seemed infeasible to give post-quantum security proof of unmodified transformations due to the non-existence of extractability.

In his seminal paper [29], Zhandry proposed the compressed oracle technique, with which the simulator can "record" quantum queries to the random oracle while simulating it efficiently. This enables to use extractability technique in the quantum setting and thus makes it possible to give security proofs of the unmodified FO and those transformations with explicit rejection in QROM.

Indeed in the full version of [29], Zhandry gave a proof that the unmodified FO turns any OW-CPA PKE into an IND-qCCA PKE in the QROM. However, in this proof, as was pointed out by Don et al. [10], the answers to decryption queries in Hybrids 2 to 4 are simulated by applying (purified) measurements on the internal state of the compressed oracle, yet these measurements are hard to be determined explicitly from their respective descriptions. Until now, this is considered as the gap that prevents the analysis of the disturbance caused by those measurements.

As for transformations with explicit rejection, Don et al. [10] presented the first IND-CCA security proof of $\mathsf{FO}_m^{\perp}$, a variant of FO transformation, in the QROM, as well as its concrete security bound. Based on their work, Hövelmanns et al. [15] improved the proof in [10] resulting in a tighter bound. However, as far as we know, there are only a few results on the IND-qCCA security proof of any transformations with explicit rejection [27].

1.1 Our Results

In this paper, we improve the IND-qCCA security proof in [29] and avoid the gap mentioned in [10]. Especially, we simplify that proof with our tool and present a tighter proof. We also give the first IND-qCCA security proof for transformation REACT and T_{CH} in the QROM, where T_{CH} is a KEM variant of REACT with explicit rejection proposed in [16]. The concrete security bounds for these three transformations are shown in Table 1.

Table 1. Concrete security bounds for FO, REACT and T_{CH} in the QROM. The "Underlying security" column omits the one-time security of the underlying SKE for both FO and REACT. ϵ^{asy} is the advantage of the reduced adversary against the security of the underlying PKE. ϵ^{sy} is the advantage against the security of the underlying SKE. d is the number of decryption or decapsulation queries. q is the total number of random oracle queries. γ is from the γ-spreadness of the underlying PKE. n is the length of the hash value being one part of the ciphertext of the achieved PKE or KEM.

Transform	Underlying security	Achieved security	Security bound($\approx$)
FO	OW-CPA	IND-qCCA	$d/\sqrt{2^{\gamma}} + (q+d)\cdot\sqrt{\epsilon^{asy}} + \epsilon^{sy}$
REACT	OW-qPCA	IND-qCCA	$d/\sqrt{2^{n}} + q\cdot d\cdot\sqrt{\epsilon^{asy}} + \epsilon^{sy}$
T_{CH}	OW-qPCA	IND-qCCA	$d/\sqrt{2^{n}} + (q+d)\cdot\sqrt{\epsilon^{asy}}$

Our main tool to prove our results is a unitary $\mathrm{U}_{\mathrm{Ext}}$ named the plaintext extraction procedure for a class of PKE called oracle-masked schemes. Informally, the oracle-masked scheme is defined as follows.

Definition 1 (Oracle-Masked Scheme, informal). *For random oracle $\mathcal{O}$ with codomain $\mathcal{Y}$, we call $\Pi = (\mathrm{Gen}, \mathrm{Enc}^{\mathcal{O}}, \mathrm{Dec}^{\mathcal{O}})$ an oracle-masked scheme if $\mathrm{Enc}^{\mathcal{O}}$ and $\mathrm{Dec}^{\mathcal{O}}$ are constructed as in Fig. 1. Parameter η of Π is defined to be*

$$\eta := \max_{(pk,sk),\, c} \left|\{y \in \mathcal{Y} : c = \mathrm{A}_2\left(pk, \mathrm{A}_3(sk, c), y\right)\}\right| / |\mathcal{Y}|\,,$$

where (pk, sk) is generated by Gen *and $c \in \mathcal{C}$ is such that $\mathrm{A}_3(sk, c) \neq \perp$.*

$\underline{\mathrm{Enc}^{\mathcal{O}}(pk, m; r)}$	$\underline{\mathrm{Dec}^{\mathcal{O}}(sk, c)}$	
$x := \mathrm{A}_1(pk, m, r)$	$x := \mathrm{A}_3(sk, c)$	**if** $c \neq c'$, **return** $\perp$
$y := \mathcal{O}(x)$	**if** $x = \perp$, **return** $\perp$	$m := \mathrm{A}_4(x)$
$c := \mathrm{A}_2(pk, x, y)$	$y := \mathcal{O}(x)$	**return** m
return c	$c' := \mathrm{A}_2(pk, x, y)$	

Fig. 1. Algorithm $\mathrm{Enc}^{\mathcal{O}}$ and $\mathrm{Dec}^{\mathcal{O}}$ of an oracle-masked scheme Π, and the tuple of algorithm A_1, A_2, A_3 and A_4 is called the decomposition of Π.

According to the above definition, oracle-masked schemes contains PKEs obtained by several transformations, including FO transformation, REACT transformation and T in the modular FO toolkit [14]. We then present the plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$ for oracle-masked scheme Π as below.

Definition 2 (Plaintext Extraction Procedure, informal). *Suppose that $\mathcal{O}$ is simulated by the compressed standard oracle CStO with database register D. Then the plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$ of oracle-masked scheme Π applied on register C, Z, D is that $\mathrm{U}_{\mathrm{Ext}}|c, z, D\rangle = |c, z \oplus f(c, D), D\rangle$, where*

$$f(c, D) := \begin{cases} \mathrm{A}_4(x) & \text{if } c \neq c^* \text{ and } \exists\, x \,\text{s.t.}\, \mathrm{A}_2(pk, x, D(x)) = c,\ \mathrm{A}_3(sk, c) = x \\ \perp & \text{otherwise.} \end{cases}$$

Plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$ is to apply extractability technique to simulate the quantum-accessible decryption oracle in the IND-qCCA security proof of Π. When random oracle $\mathcal{O}$ is simulated by CStO, the random oracle queries is recorded on the database register D. Note that the queries is not recorded perfectly, but the simulator can still learn some information from the state on D by quantum measurements or computing functions defined on database [7,10]. Following this fact, $\mathrm{U}_{\mathrm{Ext}}$ extracts plaintext $m(:= \mathrm{A}_4(x))$ for ciphertext c by computing a classical function $f(c, D)$ defined as above. Moreover, $\mathrm{U}_{\mathrm{Ext}}$ is performed efficiently if f can be computed efficiently.

With the notions defined as above, we then prove the IND-qCCA security of transformation FO, REACT and T_{CH}. Our proofs can be outlined as the following three steps.

Firstly, we represent the schemes obtained by transformations as oracle-masked schemes relative to $\mathcal{O}$ and specify their decomposition $(\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{A}_4)$. In the IND-qCCA security games of these schemes, random oracle $\mathcal{O}$ is simulated

by CStO and accordingly, the quantum decryption oracle $\mathrm{Dec}^{\mathcal{O}}$ is simulated by unitary $\mathrm{U}_{\mathrm{Sim}}$.

Next, we replace unitary $\mathrm{U}_{\mathrm{Sim}}$ with the plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$. We also present the detailed construction of $\mathrm{U}_{\mathrm{Ext}}$ without the secret key.

Finally, we apply the semi-classical O2H theorem to reprogram the compressed oracle at some points, which results in a new game. We then connect it to the security game of the underlying schemes.

Here we analyze the security loss introduced by the second and third step.

For the second step, we need to bound the security loss caused by the replacement of the simulation of the decryption oracle $\mathrm{Dec}^{\mathcal{O}}$. Since CStO perfectly simulates the random oracle, $\mathrm{U}_{\mathrm{Sim}}$ and $\mathrm{Dec}^{\mathcal{O}}$ are perfectly indistinguishable for any adversary. Then we analyze the loss introduced by performing unitary $\mathrm{U}_{\mathrm{Ext}}$. For one type of state $|\psi\rangle$, we compute the difference between $\mathrm{U}_{\mathrm{Ext}}|\psi\rangle$ and $\mathrm{U}_{\mathrm{Sim}}|\psi\rangle$ and obtain the following lemma.

Lemma 1 (Informal). *Let $|\psi\rangle$ be a quantum state on register C, Z, D that is orthogonal to $\sum_{c,z,D,x}\alpha_{c,z,D,x}|c,z,D\cup(x,\beta_0)\rangle$. Then $\|(\mathrm{U}_{\mathrm{Sim}}-\mathrm{U}_{\mathrm{Ext}})|\psi\rangle\|\leq 5\sqrt{\eta}$.*

As is argued in [10], there are at least two requirements of refining the proof in [29]: To rigorously specify the quantum measurements in Hybrid 3 and 4, respectively; To analyze the disturbance of the state of CStO caused by quantum measurements.

Our proofs meet the first requirement by providing the plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$ of oracle-masked schemes. Indeed, $\mathrm{U}_{\mathrm{Ext}}$ and the scan operation in Hybrid 4 act similarly. They both learns the information from the database. But our $\mathrm{U}_{\mathrm{Ext}}$ is represented in a more specific form and can also be viewed as a formalization of the scan operation. As for the second requirement, we apply Lemma 1 to bound the disturbance caused by performing $\mathrm{U}_{\mathrm{Ext}}$. If the adversary makes at most q decryption queries, then by the hybrid argument, the loss caused by $\mathrm{U}_{\mathrm{Ext}}$ is upper bounded by $5q\sqrt{\eta}$.

For the third step, we stress that we can not reprogram CStO only by applying the semi-classical O2H theorem. As an explanation, suppose that we puncture CStO on point x via the semi-classical oracle $\mathcal{O}^{SC}_{\{x\}}$, which forbids the adversary from querying CStO by x if event Find does not occur. However, the performance of $\mathrm{U}_{\mathrm{Ext}}$ disturbs the database state on register D, which disturbs the simulation of random oracle $\mathcal{O}$. Thus, it can not be concluded that CStO on x is uniformly random even if the adversary never queries CStO on point x (i.e., Find does not occur).

To fix it, before reprogramming the compressed oracle on x, we change $\mathrm{U}_{\mathrm{Ext}}$ into $\mathsf{StdDecomp}_x\circ\mathrm{U}_{\mathrm{Ext}}\circ\mathsf{StdDecomp}_x$, where $\mathsf{StdDecomp}_x$, the local decompression procedure defined in [29], is an involution performed on the database register D. Then by the definition of $\mathrm{U}_{\mathrm{Ext}}$, $\mathsf{StdDecomp}_x\circ\mathrm{U}_{\mathrm{Ext}}\circ\mathsf{StdDecomp}_x$ does not disturb any database state in the form of $|D\cup\mathsf{StdDecomp}_x(x,y)\rangle$, which in contrast to the disturbance made by $\mathrm{U}_{\mathrm{Ext}}$. Then we apply the following lemma to bound the difference between $\mathrm{U}_{\mathrm{Ext}}$ and $\mathsf{StdDecomp}_x\circ\mathrm{U}_{\mathrm{Ext}}\circ\mathsf{StdDecomp}_x$.

Lemma 2 (Informal). *For any x and state $|\psi\rangle$ on register C, Z, D,*

$$\left\|(\mathrm{U}_{\mathrm{Ext}} \circ \mathsf{StdDecomp}_x - \mathsf{StdDecomp}_x \circ \mathrm{U}_{\mathrm{Ext}})|\psi\rangle\right\| \leq 7\sqrt{\eta}\,.$$

Overall, we propose the notion of oracle-masked schemes and define plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$ for these schemes. They can be used to avoid the gap in the FO proof in [29]. And our proof outline can also be applied to the IND-qCCA security proofs of other transformations in the QROM.

1.2 Related work

Abstract frameworks were proposed to simplify the application of the compressed oracle technique in different situations [6,7,10]. They formalized properties that are satisfied in the presence of random oracle, and lifted them to the quantum setting.

Existing proofs from [29] already implicitly were using compressed oracles for some sort of extractability. Don et al. [10] then considered extractability in a general form. Specifically, they define a simulator $\mathcal{S}$ that simulates the random oracle and also allows the extraction query that is replied with a guess of the plaintext of the query. They then prove that this simulation of the random oracle is statistically indistinguishable from the real one if some properties are satisfied. In their security proof, the extraction query is restricted to be classical in the simulation. Therefore, their result seems to be tailored for post-quantum security proofs, yet are not sufficient to prove the IND-qCCA security.

Based on [10], Hövelmanns et al. [15] proposed a variant of semi-classical O2H theorem as the core to prove the post-quantum security of $\mathsf{FO}_m^{\perp}$. Roughly speaking, this theorem states that the probabilities of classical event EXT and $FIND$ can bound the loss caused by the reprogramming of the oracle simulated by $\mathcal{S}$. Different from their work, our argument allows the adversary to make quantum extraction query, which makes event EXT no longer make sense.

2 Preliminaries

2.1 Notation

Denote $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{R}$ as key space, message space and ciphertext space, respectively. A function $f(\lambda)$ is negligible if $f(\lambda) = \lambda^{-\omega(1)}$. Algorithms take as input a security parameter λ, and we omit it for convenience. $\mathrm{Time}(A)$ is denoted as the running time of algorithm A.

For a finite set $\mathcal{X}$, denote $|\mathcal{X}|$ as the number of elements $\mathcal{X}$ contains, and denote $x \xleftarrow{\$} \mathcal{X}$ as uniformly choose a random element x from $\mathcal{X}$. $[b = b']$ is an integer, that is 1 if $b = b'$ and 0 otherwise. $\Pr[P : Q]$ is the probability that predicate P keeps true where all the variables in P are assigned according to the program in Q.

2.2 Quantum Random Oracle Model

We refer to [22] for basics of quantum computation and quantum information.

In the ROM, we assume the existence of the random oracle $\mathcal{O} : \mathcal{X} \to \mathcal{Y}$, and $\mathcal{O}$ is publicly accessible to all parties. For concreteness, let $\mathcal{Y} = \{0,1\}^n$. $\mathcal{O}$ is initialized by choosing $H \xleftarrow{\$} \Omega_H$, where Ω_H is the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. In the QROM, quantum algorithms can query $\mathcal{O}$ with superposition states, and the oracle performs the unitary mapping $|x, y\rangle \mapsto |x, y \oplus H(x)\rangle$ on the query state. Oracle $\mathcal{O}$ also allows making classical queries. To query x, set the input and output state to be $|x, 0\rangle$ and measure it after querying $\mathcal{O}$ to obtain $H(x)$.

Below, we introduce several tools for QROM, that are used in this paper. We begin with two ways for the simulation of the quantum random oracle.

Theorem 1 ([28, Theorem 6.1]). *Let H be a function chosen from the set of $2q$-wise independent functions uniformly at random. Then for any quantum algorithm A with at most q queries,*

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

The Compressed Oracle. Here we briefly introduce the compressed oracle technique, and we only consider the Compressed Standard Oracles(CStO), one version of the compressed oracle, with query number at most q. We refer to the full version of [29] for more details of the compressed oracle.

The core idea of the compressed oracle technique is the purification of the quantum random oracle, and the purified oracle imperfectly records quantum queries to the random oracle. In the QROM, random oracle $\mathcal{O}$ is initialized by uniformly sampling a function H from Ω_H. If $\mathcal{O}$ is queried with a quantum state $|x, y\rangle$, then the replied state is a mixed state and can be represented as $\{p_i, |x, y \oplus H_i(x)\rangle\}$, where $p_i = 1/|\Omega_H|$, $i = 1, \ldots, |\Omega_H|$. This mixed state can be purified to state $1/|\Omega_H| \sum_H |x, y \oplus H(x), H\rangle$, where $|H\rangle$ is the internal state of oracle $\mathcal{O}$ and H of $|H\rangle$ is a truth table of function H.

Instead of a superposition state of H, CStO takes a superposition of database as its internal state and simulates random oracle $\mathcal{O}$. We denote this simulated oracle by CStO directly, and database by D. Here D is an element of set $\mathbf{D}_l := (\mathcal{X} \times \bar{\mathcal{Y}})^l$ where $\bar{\mathcal{Y}} = \mathcal{Y} \cup \{\perp\}$, l is the length of D. For any $x \in \mathcal{X}$, if (x, y) exists as an entry of D, then $(x, y) \in D$ and $D(x) = y$. Otherwise, $D(x) = \perp$. Denote $|D|$ as the total number of $x \in \mathcal{X}$ such that $D(x) \neq \perp$. Then for any $y \in \mathcal{Y}$ and D that $D(x) = \perp$, $|D| < l$, define $D \cup (x, y)$ to be the database that $D \cup (x, y)(x') = D(x')$ for any $x' \neq x$ and $D \cup (x, y)(x) = y$. Moreover, any D is written in the form of $((x_1, y_1), \ldots, (x_s, y_s), (0, \perp), \ldots, (0, \perp))$ such that $|D| = s \leq l$, $x_1 < x_2 < \cdots < x_s$.

For any $x \in \mathcal{X}$, define the local decompression procedure $\mathsf{StdDecomp}_x$ applied on the database state $|D\rangle \in \mathbb{C}[\mathbf{D}_l]$ as below:

- For D that $D(x) = \perp$ and $|D| = l$, $\mathsf{StdDecomp}_x |D\rangle = |D\rangle$.

- For D that $D(x) = \perp$ and $|D| < l$, $\mathsf{StdDecomp}_x|D \cup (x, \beta_r)\rangle = |D \cup (x, \beta_r)\rangle$ for any $r \neq 0$, $\mathsf{StdDecomp}_x|D \cup (x, \beta_0)\rangle = |D\rangle$, $\mathsf{StdDecomp}_x|D\rangle = |D \cup (x, \beta_0)\rangle$, where state $|D \cup (x, \beta_r)\rangle = 1/\sqrt{2^n} \sum_{y \in \mathcal{Y}} (-1)^{y \cdot r} |D \cup (x, y)\rangle$ for any $r \in \mathcal{Y}$.

CStO initializes a database state $|(0, \perp)^q\rangle$ with length q. For any query $|x, y\rangle$ to random oracle $\mathcal{O}$, CStO does three steps: First, perform the unitary $|x, y, D\rangle \mapsto |x, y\rangle \mathsf{StdDecomp}_x|D\rangle$ in superposition. Next, apply the map $|x, y, D\rangle \mapsto |x, y \oplus D(x), D\rangle$. Finally, repeat the first step.

Theorem 2 ([29, Lemma 4]). CStO *and random oracle* $\mathcal{O}$ *are indistinguishable for any quantum algorithm* A*, i.e.,*

$$\Pr[b = 1 : b \leftarrow A^{\mathsf{CStO}}()] = \Pr[b = 1 : b \leftarrow A^{\mathcal{O}}()] .$$

It is also observed that any quantum state on the database register is orthogonal to state $|D \cup (x, \beta_0)\rangle$ in the simulation of CStO. Therefore, the database state should be the superposition state of $|D \cup (x, \beta_r)\rangle$ for $r \neq 0$. This fact will be used later.

Semi-classical Oracle. For set $\mathcal{X}$ and $\mathcal{S}$, define $f_{\mathcal{S}} : \mathcal{X} \to \{0, 1\}$ to be an indicator function such that $f_{\mathcal{S}}(x) = 1$ if $x \in \mathcal{S}$ and 0 otherwise. Then we define the semi-classical oracle $\mathcal{O}_{\mathcal{S}}^{SC} : \mathcal{X} \to \{0, 1\}$. For any quantum query, $\mathcal{O}_{\mathcal{S}}^{SC}$ does the following steps. First, initialize a qubit T to be $|0\rangle$. Then evaluate the mapping $|x, 0\rangle \mapsto |x, f_{\mathcal{S}}(x)\rangle$ in superposition. Finally, measure T in the computational basis and obtain a bit $b \in \{0, 1\}$ as its output.

Theorem 3 (Semi-classical O2H [1, Theorem 1]). *Let* $\mathcal{S}$ *be a random subset of* $\mathcal{X}$*,* $H : \mathcal{X} \to \mathcal{Y}$ *a random function,* z *a random bitstring. And* H*,*$\mathcal{S}$*,*z *may have arbitrary joint distribution. Let* $H \setminus \mathcal{S}$ *be an oracle that first queries* $\mathcal{O}_{\mathcal{S}}^{SC}$ *and then queries* H*. Let* A *be a quantum oracle algorithm with query depth* d*. In the execution of* $A^{H \setminus \mathcal{S}}(z)$*, let Find be the event that* $\mathcal{O}_{\mathcal{S}}^{SC}$ *ever outputs* 1*. Then*

$$\left| \Pr[b = 1 : b \leftarrow A^{H}(z)] - \Pr[b = 1 : b \leftarrow A^{H \setminus \mathcal{S}}(z)] \right| \leq \sqrt{(d + 1) \cdot \Pr[\mathrm{Find}]} .$$

The following theorem gives an upper bound for the probability that Find occurs.

Theorem 4 ([1, Theorem 2]). *Let* $\mathcal{S} \subseteq \mathcal{X}$ *and* $z \in \{0, 1\}^*$*. And* $\mathcal{S}, z$ *may have arbitrary joint distribution. Let* A *be a quantum oracle algorithm making at most* d *queries to* $O_{\mathcal{S}}^{SC}$ *with domain* $\mathcal{X}$*. Let* B *be an algorithm that on input* z*, chooses* $i \xleftarrow{\$} \{1, \ldots, d\}$ *, runs* $A^{\mathcal{O}_{\varnothing}^{SC}}(z)$ *until (just before) the* i*-th query, and then measures all query input registers in the computational basis. Denote by* $\mathcal{T}$ *the set of measurement outcomes. Then*

$$\Pr\left[\mathrm{Find} : A^{\mathcal{O}_{\mathcal{S}}^{SC}}(z)\right] \leq 4d \cdot \Pr[\mathcal{S} \cap \mathcal{T} \neq \varnothing : \mathcal{T} \leftarrow B(z)] .$$

3 Plaintext Extraction of the Oracle-Masked Scheme

In this section, we start by the formalization of the class of PKE Π named the oracle-masked scheme. Then we will introduce plaintext extraction game $\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi}$ for adversary A, and end this section with a theorem that bounds the difference of the output distributions of $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ and $\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi}$. The definition of the IND-qCCA security game $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ is shown in the appendix B.2.

Definition 3 (Oracle-Masked Scheme). *Let $\Pi = (\mathrm{Gen}, \mathrm{Enc}^{\mathcal{O}}, \mathrm{Dec}^{\mathcal{O}})$ be a PKE relative to random oracle $\mathcal{O}$ with codomain $\mathcal{Y}$. We say that Π is an oracle-masked scheme if there exist deterministic polynomial time algorithm A_1, A_2, A_3, A_4 such that for any (pk, sk) generated by* Gen, $\mathrm{Enc}^{\mathcal{O}}$ *and* $\mathrm{Dec}^{\mathcal{O}}$ *are written as in Fig. 2. Tuple $(\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{A}_4)$ is called the decomposition of Π.*

$\underline{\mathrm{Enc}^{\mathcal{O}}(pk, m; r)}$	$\underline{\mathrm{Dec}^{\mathcal{O}}(sk, c)}$	
$x := \mathrm{A}_1(pk, m, r)$	$x := \mathrm{A}_3(sk, c)$	**if** $c \neq c'$, **return** $\perp$
$y := \mathcal{O}(x)$	**if** $x = \perp$, **return** $\perp$	$m := \mathrm{A}_4(x)$
$c := \mathrm{A}_2(pk, x, y)$	$y := \mathcal{O}(x)$	**return** m
return c	$c' := \mathrm{A}_2(pk, x, y)$	

Fig. 2. Algorithm $\mathrm{Enc}^{\mathcal{O}}$ and $\mathrm{Dec}^{\mathcal{O}}$ of an oracle-masked scheme Π

For an oralce-masked scheme Π, parameter η of Π is defined to be

$$\eta := \max_{(pk,sk),\, c} \left|\{y \in \mathcal{Y} : c = \mathrm{A}_2\left(pk, \mathrm{A}_3(sk, c), y\right)\}\right| / |\mathcal{Y}|\,,$$

where (pk, sk) is generated by Gen *and $c \in \mathcal{C}$ is such that $\mathrm{A}_3(sk, c) \neq \perp$.*

Let Π be an oracle-masked scheme. For quantum adversary A in the security game $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ in the QROM, it can query random oracle $\mathcal{O}$ and decryption oracle $\mathrm{Dec}^{\mathcal{O}}$ both in superposition. Write C and Z to denote the input and output register of the decryption query of A, respectively. The decryption oracle $\mathrm{Dec}^{\mathcal{O}}$ in $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ can be simulated by a unitary operator $\mathrm{U}_{\mathrm{Dec}}$ applied on register C and Z, i.e., for any computational basis state $|c, z\rangle$, $\mathrm{U}_{\mathrm{Dec}}$ acts as follows:

$$\mathrm{U}_{\mathrm{Dec}}|c, z\rangle = \begin{cases} |c, z \oplus \perp\rangle & \text{if } c^* \text{ is defined and } c = c^* \\ |c, z \oplus \mathrm{Dec}^{\mathcal{O}}(c)\rangle & \text{else.} \end{cases}$$

where c^* is the challenge ciphertext in $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$.

Then we introduce a new game $\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi}$, that is identical with $\mathrm{Game}^{\text{IND-qCCA}}_{A\Pi}$ except that random oracle $\mathcal{O}$ is simulated by CStO. In this game, quantum queries to oracle $\mathcal{O}$ are recorded in the database register D imperfectly. The

decryption oracle answers queries in the same process as in Fig. 2 and it can be simulated by a unitary operator on register C, Z, D. We denote this operator by $\mathrm{U}_{\mathrm{Sim}}$. Then by Theorem 2, $\mathrm{U}_{\mathrm{Dec}}$ and $\mathrm{U}_{\mathrm{Sim}}$, these two simulations of the decryption oracle are perfectly indistinguishable for any quantum adversary.

Notice that in the process of the decryption algorithm $\mathrm{Dec}^{\mathcal{O}}$, A_3 is computed first to obtain x and then A_2 is applied to check if $c = \mathrm{A}_2(pk, x, \mathcal{O}(x))$. Then the query x to oracle $\mathcal{O}$ is recorded in the database D imperfectly if the decryption oracle is simulated by $\mathrm{U}_{\mathrm{Sim}}$. With this property, we design a new unitary to reply decryption queries, and it is defined as follows.

Definition 4 (Plaintext Extraction Procedure). *Let Π be an oracle-masked scheme and $(\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{A}_4)$ be its decomposition. For any (pk, sk) of Π, define unitary operation $\mathrm{U}_{\mathrm{Ext}}$, as the plaintext extraction procedure of Π, applied on register C, Z, D as follows.*

$\underline{\mathrm{U}_{\mathrm{Ext}}|c, z, D\rangle}$:

1. *If the challenge ciphertext c^* is defined and $c = c^*$, return $|c, z \oplus \perp, D\rangle$.*
2. *Else if database D contains no pair $(x, D(x))$ such that $\mathrm{A}_2(pk, x, D(x)) = c$, return $|c, z \oplus \perp, D\rangle$.*
3. *Else, for each tuple $(x, D(x))$ that $\mathrm{A}_2(pk, x, D(x)) = c$, check if $\mathrm{A}_3(sk, c) = x$ and do the following procedure:*
   (a) *If a tuple $(x, D(x))$ passes this test,[1] compute $m := \mathrm{A}_4(x)$ and return $|c, z \oplus m, D\rangle$.*
   (b) *Otherwise, return $|c, z \oplus \perp, D\rangle$.*

In addition, the detailed construction of $\mathrm{U}_{\mathrm{Ext}}$ is shown in appendix A.

Compared with $\mathrm{U}_{\mathrm{Sim}}$, $\mathrm{U}_{\mathrm{Ext}}$ does not follow the decryption algorithm to produce the plaintext $m(:= \mathrm{Dec}^{\mathcal{O}}(sk, c))$, but just searches $(x, D(x))$ on D to obtain m. Therefore, we call $\mathrm{U}_{\mathrm{Ext}}$ the plaintext extraction procedure.

By the definition of $\mathrm{U}_{\mathrm{Ext}}$, for any computational basis state $|c, z, D\rangle$, $\mathrm{U}_{\mathrm{Ext}}$ has no effect on $|D\rangle$, and does not need to query oracle $\mathcal{O}$. And for any oracle-masked scheme, such a plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$ exists, and it can be used to answer quantum decryption queries. Then we introduce two properties of $\mathrm{U}_{\mathrm{Ext}}$ by the following two lemmas. Except register C, Z and D, we abbreviate other registers (e.g. other registers of adversary A) into W and the detailed proofs of these lemmas are shown in the full version [25].

Lemma 3. *Let $|\psi\rangle$ be a quantum state on register W, C, Z and D such that $|\psi\rangle$ is orthogonal to any state in the form of $\sum_{w,c,z,D,x} \alpha_{w,c,z,D,x}|w, c, z, D \cup (x, \beta_0)\rangle$. Then*

$$\|(\mathrm{U}_{\mathrm{Sim}} - \mathrm{U}_{\mathrm{Ext}})|\psi\rangle\| \leq 5\sqrt{\eta}\,.$$

Lemma 4. *Given any $x \in \{0,1\}^*$, unitary $\mathsf{StdDecomp}_x$ is performed on register D. For any quantum state $|\psi\rangle$ on register W, C, Z and D,*

$$\left\|(\mathrm{U}_{\mathrm{Ext}} \circ \mathsf{StdDecomp}_x - \mathsf{StdDecomp}_x \circ \mathrm{U}_{\mathrm{Ext}})|\psi\rangle\right\| \leq 7\sqrt{\eta}\,.$$

[1] Such a tuple is unique, since c and sk determines the value of $\mathrm{A}_3(sk, c)$.

Here we define a new game $\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi}$ named plaintext extraction game that differs from $\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi}$ in the way of answering decryption queries: In $\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi}$, the decryption oracle is simulated by unitary $\mathrm{U}_{\mathrm{Ext}}$ while that in $\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi}$ is simulated by unitary $\mathrm{U}_{\mathrm{Sim}}$. With Lemma 3, we obtain Theorem 5 as follows to bound the output difference of $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ and $\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi}$.

Theorem 5. *Let Π be an oracle-masked scheme. For any quantum adversary A against the* IND-qCCA *security of Π in the QROM, if A makes at most q decryption queries, then*

$$\left|\Pr[\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi} \to 1] - \Pr[\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi} \to 1]\right| \leq 5q \cdot \sqrt{\eta}\,.$$

Proof. Given Π and A, recall that $\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi}$ is identical with $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ except that the random oracle is simulated by CStO. By Theorem 2,

$$\Pr[\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi} \to 1] = \Pr[\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi} \to 1]\,.$$

In the following, we prove that

$$\left|\Pr[\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi} \to 1] - \Pr[\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi} \to 1]\right| \leq 5q \cdot \sqrt{\eta}\,.$$

For any fixed (pk, sk), the decryption oracle in $\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi}$ and that in $\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi}$ are simulated by unitary $\mathrm{U}_{\mathrm{Sim}}$ and $\mathrm{U}_{\mathrm{Ext}}$, respectively.

For any $i = 1, \ldots, q$, define G_i to be a game that is the same as $\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi}$ until just before the i-th decryption query of A, then simulates the decryption oracle with unitary $\mathrm{U}_{\mathrm{Ext}}$ instead of $\mathrm{U}_{\mathrm{Sim}}$. Then G_1 is exactly $\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi}$. We also denote $\mathrm{Game}^{\mathrm{Sim}}_{A,\Pi}$ by G_{q+1}.

For $i = 1, \ldots, q+1$, denote by σ_i the final joint state of the registers of G_i including the register of A and the database register. By the triangle inequality of the trace distance,

$$\mathrm{TD}(\sigma_1, \sigma_{q+1}) \leq \mathrm{TD}(\sigma_1, \sigma_2) + \ldots + \mathrm{TD}(\sigma_q, \sigma_{q+1})\,,$$

where $\mathrm{TD}(\rho, \tau)$ is the trace distance of state ρ and τ.

Fix $1 \leq i \leq q$. Since game G_i and G_{i+1} only differ in the i-th decryption query, we denote by ρ the joint state of A and the database register just before the i-th decryption query. All the operations after the i-th decryption query can be represented by a trace-preserving operation, that is denoted by $\mathcal{E}$. Then σ_i and σ_{i+1} can be represented by $\sigma_i = \mathcal{E}(\mathrm{U}_{\mathrm{Sim}}\,\rho\mathrm{U}^{\dagger}_{\mathrm{Sim}})$ and $\sigma_{i+1} = \mathcal{E}(\mathrm{U}_{\mathrm{Ext}}\,\rho\mathrm{U}^{\dagger}_{\mathrm{Ext}})$, respectively. And we have

$$\mathrm{TD}(\sigma_i, \sigma_{i+1}) \leq \mathrm{TD}(\mathrm{U}_{\mathrm{Sim}}\,\rho\mathrm{U}^{\dagger}_{\mathrm{Sim}}, \mathrm{U}_{\mathrm{Ext}}\,\rho\mathrm{U}^{\dagger}_{\mathrm{Ext}})\,.$$

Let $\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j|$ be a spectral decomposition of ρ, where $\sum_j p_j = 1$. Then by the convexity of the trace distance,

$$\begin{aligned}
&\mathrm{TD}(\mathrm{U}_{\mathrm{Sim}}\,\rho\mathrm{U}_{\mathrm{Sim}}^{\dagger}, \mathrm{U}_{\mathrm{Ext}}\,\rho\mathrm{U}_{\mathrm{Ext}}^{\dagger}) \\
&= \mathrm{TD}\big(\sum_j p_j \mathrm{U}_{\mathrm{Sim}}|\psi_j\rangle\langle\psi_j|\mathrm{U}_{\mathrm{Sim}}^{\dagger}, \sum_j p_j \mathrm{U}_{\mathrm{Ext}}|\psi_j\rangle\langle\psi_j|\mathrm{U}_{\mathrm{Ext}}^{\dagger}\big) \\
&\leq \sum_j p_j \mathrm{TD}(\mathrm{U}_{\mathrm{Sim}}|\psi_j\rangle\langle\psi_j|\mathrm{U}_{\mathrm{Sim}}^{\dagger}, \mathrm{U}_{\mathrm{Ext}}|\psi_j\rangle\langle\psi_j|\mathrm{U}_{\mathrm{Ext}}^{\dagger}) \\
&\leq \sum_j p_j \|(\mathrm{U}_{\mathrm{Sim}} - \mathrm{U}_{\mathrm{Ext}})|\psi_j\rangle\| \,.
\end{aligned}$$

Note that before the i-th decryption query, the decryption procedure is $\mathrm{U}_{\mathrm{Sim}}$ and A can be considered as being in $\mathrm{Game}_{A,\Pi}^{\mathrm{Sim}}$. Thus, any state $|\psi_j\rangle$ in the spectral decomposition of ρ is in the form of the superposition state in Lemma 3. By Lemma 3, $\|(\mathrm{U}_{\mathrm{Sim}} - \mathrm{U}_{\mathrm{Ext}})|\psi_j\rangle\| \leq 5\sqrt{\eta}$. Then for every $1 \leq i \leq q$,

$$\mathrm{TD}(\sigma_i, \sigma_{i+1}) \leq \sum_j p_j \cdot \|(\mathrm{U}_{\mathrm{Sim}} - \mathrm{U}_{\mathrm{Ext}})|\psi_j\rangle\| \leq \sum_j p_j \cdot 5\sqrt{\eta} = 5\sqrt{\eta}\,.$$

Thus, $\mathrm{TD}(\sigma_1, \sigma_{q+1}) \leq 5q \cdot \sqrt{\eta}$. Further, the output difference of $\mathrm{Game}_{A,\Pi}^{\mathrm{Sim}}$ and $\mathrm{Game}_{A,\Pi}^{\mathrm{Ext}}$ is upper bounded by the trace distance of σ_1 and σ_{q+1}, the states of these two games. This completes the proof. □

4 Application in the Quantum Security Proof

In this section, we apply Theorem 5 of oracle-masked schemes to provide the IND-qCCA security proof for transformation FO, REACT and T_{CH} in the QROM.

4.1 FO: from OW-CPA to IND-qCCA in the QROM

Let $\Pi^{asy} = (\mathrm{Gen}^{asy}, \mathrm{Enc}^{asy}, \mathrm{Dec}^{asy})$ be a PKE with message space $\mathcal{M}^{asy}$, randomness space $\mathcal{R}^{asy}(= \{0,1\}^n)$ and ciphertext space $\mathcal{C}^{asy}$. Let $\Pi^{sy} = (\mathrm{Enc}^{sy}, \mathrm{Dec}^{sy})$ be a SKE with key space $\mathcal{K}^{sy}$, message space $\mathcal{M}^{sy}$ and ciphertext space $\mathcal{C}^{sy}$. Let $H : \{0,1\}^* \to \mathcal{R}^{asy}$ and $G : \{0,1\}^* \to \mathcal{K}^{sy}$ be hash functions. We review the FO transformation in the following definition, and then provide its IND-qCCA security proof in the QROM.

Definition 5. $\mathsf{FO}[\Pi^{asy}, \Pi^{sy}, H, G] = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *obtained from the FO transformation is constructed as shown in Fig. 3.*

Lemma 5. *Assume that H is the random oracle and Π^{asy} is γ-spread, then $\mathsf{FO}[\Pi^{asy}, \Pi^{sy}, H, G]$ is an oracle-masked scheme relative to H, and its parameter η is such that $\eta \leq 1/2^{\gamma}$.*

Proof. We define deterministic polynomial-time algorithm A_1, A_2, A_3 and A_4:

$\underline{\mathrm{Gen}}$	$\underline{\mathrm{Enc}(pk, m; \delta)}$	$\underline{\mathrm{Dec}(sk, (c, d))}$
$(pk, sk) \leftarrow \mathrm{Gen}^{asy}$	$d := \mathrm{Enc}^{sy}(G(\delta), m)$	$\delta' := \mathrm{Dec}^{asy}(sk, c)$
return (pk, sk)	$c := \mathrm{Enc}^{asy}(pk, \delta; H(\delta, d))$	**if** $\delta' = \perp$, **return** $\perp$
	return (c, d)	$c' := \mathrm{Enc}^{asy}(pk, \delta'; H(\delta', d))$
		if $c' \neq c$, **return** $\perp$
		$m := \mathrm{Dec}^{sy}(G(\delta'), d)$
		return m

Fig. 3. PKE $\mathsf{FO}[\Pi^{asy}, \Pi^{sy}, H, G]$ obtained from FO transformation

- A_1 on input δ and m, evaluates $k := G(\delta)$ and $d := \mathrm{Enc}^{sy}(k, m)$, then outputs (δ, d).
- A_2 takes pk, (δ, d) and $y \in \mathcal{R}^{asy}$ as input, computes $c := \mathrm{Enc}^{asy}(pk, \delta; y)$, then outputs (c, d).
- A_3 takes sk and (c, d) as input, evaluates $\delta := \mathrm{Dec}^{asy}(sk, c)$. If $\delta \neq \perp$, output (δ, d). Otherwise, output $\perp$.
- A_4 on input (δ, d), computes $k := G(\delta)$ and $m := \mathrm{Dec}^{sy}(k, d)$, outputs m.

It can be verified that with these four algorithms, algorithm Enc and Dec given in Fig. 3 are written as $\mathrm{Enc}^{\mathcal{O}}$ and $\mathrm{Dec}^{\mathcal{O}}$ in Definition 3 with $\mathcal{O} = H$, respectively. Thus, $\mathsf{FO}[\Pi^{asy}, \Pi^{sy}, H, G]$ is an oracle-masked scheme, and its parameter η is

$$\eta = \max_{(pk,sk),\, c} \left|\{r \in \mathcal{R}^{asy} : c = \mathrm{Enc}^{asy}(pk, \mathrm{Dec}^{asy}(sk, c); r)\}\right| / |\mathcal{R}^{asy}| \,,$$

where (pk, sk) and $c \in \mathcal{C}^{asy}$ are such that $\mathrm{Dec}^{asy}(sk, c) \in \mathcal{M}^{asy}$.

Since Π^{asy} is γ-spread, for any (pk, sk) and $m \in \mathcal{M}^{asy}$,

$$\max_{c \in \mathcal{C}^{asy}} \left|\{r \in \mathcal{R}^{asy} : c = \mathrm{Enc}^{asy}(pk, m; r)\}\right| / |\mathcal{R}^{asy}| \leq 1/2^{\gamma} \,.$$

Therefore, $\eta \leq 1/2^{\gamma}$. □

Note that the above evaluation of function G can be replaced by querying an oracle that computes G. Then algorithm A_1 and A_4 become oracle algorithms denoted by A_1^G and A_4^G, respectively. In this case, the notions in Definition 3 still work, and Theorem 5 holds. Then we apply Theorem 5 to prove the IND-qCCA security of oracle-masked scheme $\mathsf{FO}[\Pi^{asy}, \Pi^{sy}, H, G]$ in the QROM.

Theorem 6. *Let Π^{asy} be γ-spread, for any adversary against the* IND-qCCA *security of scheme $\Pi = \mathsf{FO}[\Pi^{asy}, \Pi^{sy}, H, G]$, making at most q_D queries to the decryption oracle, at most q_H queries to random oracle H and at most q_G queries to random oracle G, there exist an adversary A_{asy} against the* OW-CPA *security of Π^{asy} and an adversary A_{sy} against the* OT *security of Π^{sy} such that*

$$\mathrm{Adv}^{\text{IND-qCCA}}_{A,\Pi} \leq q_D \cdot \frac{12}{\sqrt{2^{\gamma}}} + 2(d+1)\sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{A_{asy},\Pi^{asy}}} + 4d \cdot \mathrm{Adv}^{\text{OW-CPA}}_{A_{asy},\Pi^{asy}} + \mathrm{Adv}^{\text{OT}}_{A_{sy},\Pi^{sy}} \,,$$

where $d = q_D + q_H + 2q_G$, $\mathrm{Time}(A_{sy}) \approx \mathrm{Time}(A) + O\left(d^2 + q_H \cdot q_D \cdot \mathrm{Time}(\mathrm{Enc}^{asy})\right)$ and $\mathrm{Time}(A_{asy}) \approx \mathrm{Time}(A_{sy})$.

Proof. Define **Game 0** to be $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ as in Fig. 4. Then we obtain

$$\left|\Pr[\textbf{Game 0} \to 1] - \frac{1}{2}\right| = \mathrm{Adv}^{\text{IND-qCCA}}_{A,\Pi}. \tag{1}$$

In the following, we will introduce a sequence of games to bound $\mathrm{Adv}^{\text{IND-qCCA}}_{A,\Pi}$.

$\underline{\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}}$
$G \xleftarrow{\$} \Omega_G,\ H \xleftarrow{\$} \Omega_H$
$(pk, sk) \leftarrow \mathrm{Gen}$
$(m_0, m_1) \leftarrow A^{H,G,\mathrm{Dec}_\perp}(pk)$
$b \xleftarrow{\$} \{0,1\},\ \delta^* \xleftarrow{\$} \mathcal{M}^{asy}$
$d^* := \mathrm{Enc}^{sy}(G(\delta^*), m_b)$
$c^* := \mathrm{Enc}^{asy}(pk, \delta^*; H(\delta^*, d^*))$
$b' \leftarrow A^{H,G,\mathrm{Dec}_{(c^*,d^*)}}(pk, (c^*, d^*))$
return $[b = b']$

$\underline{\mathrm{Dec}_a(sk, (c,d))}$
if $(c,d) = a$, **return** $\perp$
$\delta' := \mathrm{Dec}^{asy}(sk, c)$
if $\delta' = \perp$, **return** $\perp$
$c' := \mathrm{Enc}^{asy}(pk, \delta'; H(\delta', d))$
if $c' \neq c$, **return** $\perp$
$m' := \mathrm{Dec}^{sy}(G(\delta'), d)$
return m'

Fig. 4. $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ for FO transformation in the QROM, where oracle H, G and Dec_a are all quantum-accessible.

Starting from **Game 1**, random oracle H is simulated with CStO and its database register is denoted as D. This change is undetectable for A by Theorem 2. Moreover, δ^* is sampled uniformly at the beginning of the game, which is also undetectable for any adversary.

Game 1: In this game, the decryption oracle is simulated by the plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$ of Π. We refer to appendix A for the detailed construction of $\mathrm{U}_{\mathrm{Ext}}$ of Π without sk.

Omitting the $(c,d) = (c^*, d^*)$ case, $\mathrm{U}_{\mathrm{Ext}}$ can also be rephrased as $\mathrm{U}_{\mathrm{Ext}} = \mathrm{U}_{\mathrm{E}}^{\dagger} \circ \mathrm{U}_{\mathrm{C}} \circ \mathrm{U}_{\mathrm{E}}$, based on Lemma 5. Here unitary U_{E} is used to extract (δ', d) corresponding to (c,d) from database and unitary U_{C} is used to compute plaintext m' from (δ', d). And U_{E} acts as follows.

$$\mathrm{U}_{\mathrm{E}}|(c,d), z_1, D\rangle = \begin{cases} |(c,d), z_1 \oplus (1, (\delta', d)), D\rangle & \text{if } \mathrm{Enc}^{asy}(pk, \delta'; D(\delta', d)) = c \\ |(c,d), z_1 \oplus (0, 0^n), D\rangle & \text{otherwise.} \end{cases}$$

It is obvious that **Game 1** is the plaintext extraction game $\mathrm{Game}^{\mathrm{Ext}}_{A,\Pi}$. Then by Theorem 5, we obtain $\left|\Pr[\textbf{Game 0} \to 1] - \Pr[\textbf{Game 1} \to 1]\right| \leq 5q_D \cdot \sqrt{\eta}$ for any fixed $G \in \Omega_G$. Therefore,

$$\left|\Pr[\textbf{Game 0} \to 1] - \Pr[\textbf{Game 1} \to 1]\right| \leq 5q_D \cdot \sqrt{\eta} \leq q_D \cdot \frac{5}{\sqrt{2^\gamma}}, \tag{2}$$

where variable G, both in **Game 0** and **Game 1**, is sampled from Ω_G uniformly.

Game 2: This game is identiacal with **Game 1** except that the decryption oracle is simulated by the following steps after the challenge query.

1. Perform unitary $\mathsf{StdDecomp}_{(\delta^*,d^*)}$ to register D.
2. Apply $\mathrm{U}_{\mathrm{Ext}}$ on register C, Z and D.
3. Perform $\mathsf{StdDecomp}_{(\delta^*,d^*)}$ to register D a second time.

We define unitary $\mathrm{SU}_{\mathrm{Ext}} := \mathsf{StdDecomp}_{(\delta^*,d^*)} \circ \mathrm{U}_{\mathrm{Ext}} \circ \mathsf{StdDecomp}_{(\delta^*,d^*)}$. If we flip the order of the last two steps of $\mathrm{SU}_{\mathrm{Ext}}$, then $\mathsf{StdDecomp}_{(\delta^*,d^*)} \circ \mathsf{StdDecomp}_{(\delta^*,d^*)}$ is an identity operator and in this way, $\mathrm{SU}_{\mathrm{Ext}}$ performs identically as $\mathrm{U}_{\mathrm{Ext}}$. Since Lemma 4 states that $\mathrm{U}_{\mathrm{Ext}}$ commutes with $\mathsf{StdDecomp}_{(\delta^*,d^*)}$ by a loss, we have

$$\mathrm{TD}(\mathrm{U}_{\mathrm{Ext}}\rho\mathrm{U}_{\mathrm{Ext}}^{\dagger}, \mathrm{SU}_{\mathrm{Ext}}\rho\,\mathrm{SU}_{\mathrm{Ext}}^{\dagger}) \leq 7\sqrt{\eta} \leq \frac{7}{\sqrt{2^{\gamma}}}$$

for any joint state ρ on registers in **Game 2**. At most q_D decryption queries are made after the challenge query, and then by the hybrid argument,

$$|\Pr[\textbf{Game 1} \to 1] - \Pr[\textbf{Game 2} \to 1]| \leq q_D \cdot \frac{7}{\sqrt{2^{\gamma}}}\,. \tag{3}$$

Game 3: Differing from **Game 2**, we change the way to answer random oracle queries in some cases: when random oracle H or G is queried by A or G is applied in the decryption process, we query E and then query the random oracle, where E is a constant zero function with quantum access.

Since E is a constant zero function, the random oracle query does not change after querying E, and we have

$$\Pr[\textbf{Game 2} \to 1] = \Pr[\textbf{Game 3} \to 1]\,. \tag{4}$$

Game 4: The only difference btween **Game 3** and **Game 4** is that the semi-classical oracle $O_{\mathcal{S}}^{SC}$ is applied before each query to E, and set $\mathcal{S} := \{\delta^*, \delta^*\|\cdot\}$.

Let $z := \delta^*$, and $B^E(\delta^*)$ be the algorithm that runs A and simulates **Game 3**. Then we have

$$\begin{aligned}
\Pr[\textbf{Game 3} \to 1] &= \Pr[b = 1 : b \leftarrow B^{E}(\delta^*), \delta^* \xleftarrow{\$} \mathcal{M}^{asy}]\,,\\
\Pr[\textbf{Game 4} \to 1] &= \Pr[b = 1 : b \leftarrow B^{E\setminus\mathcal{S}}(\delta^*), \delta^* \xleftarrow{\$} \mathcal{M}^{asy}]\,,\\
\Pr[\mathrm{Find} : \textbf{Game 4}] &= \Pr[\mathrm{Find} : B^{E\setminus\mathcal{S}}(\delta^*), \delta^* \xleftarrow{\$} \mathcal{M}^{asy}]\,.
\end{aligned}$$

It can be verifid that B makes at most $q_H + q_G + 2q_D$ queries to E. We let $d = q_H + q_G + 2q_D$ and apply Theorem 3 to obtain

$$|\Pr[\textbf{Game 3} \to 1] - \Pr[\textbf{Game 4} \to 1]| \leq \sqrt{(d+1)\Pr[\mathrm{Find} : \textbf{Game 4}]}\,. \tag{5}$$

Notice that by A_4 defined in Lemma 5, G is queried in the process of U_{C} when performing $\mathrm{U}_{\mathrm{Ext}}$. Then oracle $\mathcal{O}_{\mathcal{S}}^{SC}$ should be queried in the process of U_{C} in **Game 4**. We denote by U'_{C} the modified U_{C}. Accordingly, before the

challenge query, the decryption oracle in **Game 4** is simulated by $\mathrm{U_E} \circ \mathrm{U'_C} \circ \mathrm{U_E^\dagger}$, that is denoted by $\mathrm{U'_{Ext}}$. After that, the decryption oracle is simulated by $\mathsf{StdDecomp}_{(\delta^*,d^*)} \circ \mathrm{U'_{Ext}} \circ \mathsf{StdDecomp}_{(\delta^*,d^*)}$, that is denoted by $\mathrm{SU'_{Ext}}$.

We assume that Find does not occur in **Game 4**. In this case, A never queries H by (δ^*, d^*), and the database D is such that $D(\delta^*, d^*) = \perp$ until the challenge query. To produce the challenge ciphertext, $r^* := H(\delta^*, d^*)$ is computed and then the joint state is in a superposition of $\mathsf{StdDecomp}_{(\delta^*,d^*)}|w, D \cup ((\delta^*, d^*), r^*)\rangle$, here w is other registers of this game and $D(\delta^*, d^*) = \perp$. Then by the definition of $\mathrm{U_E}$, we can conclude that for any ciphertext $(c, d) \neq (c^*, d^*)$,

$$\mathrm{U_E}|(c, d), z_1, D \cup ((\delta^*, d^*), r^*)\rangle = |(c, d), z_1 \oplus (b, x), D \cup ((\delta^*, d^*), r^*)\rangle$$

if and only if $\mathrm{U_E}|(c, d), z_1, D\rangle = |(c, d), z_1 \oplus (b, x), D\rangle$.

Furthermore, observe that $\mathsf{StdDecomp}_{(\delta^*,d^*)}$ commutes with $\mathrm{U'_C}$ of $\mathrm{U'_{Ext}}$. Then for any ciphertext $(c, d) \neq (c^*, d^*)$,

$$\begin{aligned}\mathrm{SU'_{Ext}} \circ \mathsf{StdDecomp}_{(\delta^*,d^*)}|(c, d), z, D \cup ((\delta^*, d^*), r^*)\rangle \\ = \mathsf{StdDecomp}_{(\delta^*,d^*)}|c, z \oplus m', D \cup ((\delta^*, d^*), r^*)\rangle\end{aligned}$$

if and only if $\mathrm{U'_{Ext}}|(c, d), z, D\rangle = |(c, d), z \oplus m', D\rangle$. This means that the database state on (δ^*, d^*) is not involved in the decryption process of **Game 4**. Therefore, if Find does not occur, then random oracle H and G are never queried by (δ^*, d) and δ^* by the adversary. Meanwhile, the adversary A can not get information on $H(\delta^*, d^*)$ either by making decryption queries. Therefore, it is undetectable for adversary A to produce the challenge ciphertext with uniformly chosen $k^* \in \mathcal{K}^{sy}$ and $r^* \in \mathcal{R}^{say}$, which is the difference between **Game 4** and **Game 5**.

Game 5: In this game, we pick $k^* \in \mathcal{K}^{sy}$ and $r^* \in \mathcal{R}^{asy}$ uniformly and use them to produce the challenge ciphertext (c^*, d^*). And we replace $\mathrm{SU'_{Ext}}$ with $\mathrm{U'_{Ext}}$.

As analysis in **Game 4**, the view of A in **Game 4** and that in **Game 5** are identical until Find occurs. Therefore,

$$\Pr[\text{Find} : \textbf{Game 4}] = \Pr[\text{Find} : \textbf{Game 5}], \tag{6}$$

$$\Pr[\neg\text{Find} \wedge \textbf{Game 4} \to 1] = \Pr[\neg\text{Find} \wedge \textbf{Game 5} \to 1]. \tag{7}$$

Lemma 6. *There exists a quantum adversary A_{sy} invoking A such that*

$$\left|\Pr[\textbf{Game 5} \to 1] - \frac{1}{2}\right| = \mathrm{Adv}^{\mathrm{OT}}_{A_{sy}, \Pi^{sy}} \tag{8}$$

and $\mathrm{Time}(A_{sy}) \approx \mathrm{Time}(A) + O((q_H + q_G + 2q_D)^2 + q_H \cdot q_D \cdot \mathrm{Time}(\mathrm{Enc}^{asy}))$.

Proof. A quantum algorithm A_{sy} that runs A and breaks the one-time security of Π^{sy} is constructed as follows.

A_{sy} generates $(pk, sk) \leftarrow \mathrm{Gen}$, picks $\delta^* \xleftarrow{\$} \mathcal{M}^{asy}$ and simulates **Game 5** for A. Random oracle G is simulated by a $2(q_G + 2q_D)$-wise independent function, and other oracles used in **Game 5** can be implemented efficiently by A_{sy}. For A's

challenge query (m_0, m_1), A_{sy} sends it to the challenger in $\mathrm{Game}^{\mathrm{OT}}_{A_{sy},\Pi^{sy}}$. After receiving d^*, A_{sy} picks $r \in \mathcal{R}^{asy}$ uniformly, then computes $c^* := \mathrm{Enc}^{asy}(pk, \delta^*; r)$ and sends (c^*, d^*) back to A. After receiving b' from A, A_{sy} output b'.

From the construction of A_{sy}, the output of A_{sy} is correct if and only if A guesses correctly. Moreover, the view of A invoked by A_{sy} is identical with that in **Game 5**. Therefore,

$$\left|\Pr[\textbf{Game 5} \to 1] - \frac{1}{2}\right| = \left|\Pr[\mathrm{Game}^{\mathrm{OT}}_{A_{sy},\Pi^{sy}} \to 1] - \frac{1}{2}\right| = \mathrm{Adv}^{\mathrm{OT}}_{A_{sy},\Pi^{sy}}.$$

Denote by $\mathrm{T}_{\mathcal{O}}$ the time needed to simulate oracle $\mathcal{O}$, then the running time of B is given by $\mathrm{Time}(B) = \mathrm{Time}(A) + \mathrm{T}_G + \mathrm{T}_H + \mathrm{Time}(\mathrm{U}_{\mathrm{Ext}})$, where $\mathrm{T}_G = O\left((q_G + 2q_D)^2\right)$, $\mathrm{T}_H = O(q_H^2)$, $\mathrm{Time}(\mathrm{U}_{\mathrm{Ext}}) = O(q_D \cdot q_H \cdot \mathrm{Time}(\mathrm{Enc}^{asy}))$ by appendix A.1. □

Lemma 7. *There is a quantum adversary A_{asy} invoking A such that*

$$\Pr[\mathrm{Find} : \textbf{Game 5}] \leq 4d \cdot \mathrm{Adv}^{\text{OW-CPA}}_{A_{asy},\Pi^{asy}} \tag{9}$$

and $\mathrm{Time}(A_{asy}) \approx \mathrm{Time}(A) + O((q_H + q_G + 2q_D)^2 + q_H \cdot q_D \cdot \mathrm{Time}\left(\mathrm{Enc}^{asy}\right))$.

Proof. Define $B^{\mathcal{O}^{SC}_{\mathcal{S}}}$ as a quantum oracle algorithm that on input pk, c^*, runs A and simulates **Game 5** for it. Then we have $\Pr[\mathrm{Find} : \textbf{Game 5}] = \Pr[\mathrm{Find} : B^{\mathcal{O}^{SC}_{\mathcal{S}}}(pk, c^*)]$, where $c^* \leftarrow \mathrm{Enc}^{asy}(pk, \delta^*)$, δ^* is sampled uniformly from $\mathcal{M}^{asy}$. As analyzed in **Game 4**, B makes at most $d = q_H + q_G + 2q_D$ queries, then by Theorem 4,

$$\Pr[\mathrm{Find} : B^{\mathcal{O}^{SC}_{\mathcal{S}}}(pk, c^*)] \leq 4d \cdot \Pr[(\delta, d) \in \mathcal{S} : (\delta, d) \leftarrow D(pk, c^*)].$$

Here D is a quantum algorithm invoking B. On input (pk, c^*), D chooses $i \xleftarrow{\$} \{1, \ldots, d\}$, runs $B^{\mathcal{O}^{SC}_{\varnothing}}(pk, c^*)$ until (just before) i-th query of B, and then measures the state on the input register of $\mathcal{O}^{SC}_{\varnothing}$ to obtain (δ, d). Note that the running time of D and that of B are almost the same.

Because $\mathcal{S} = \{\delta^* \| \cdot\}$, $(\delta, d) \in \mathcal{S}$ is equivalent to $\delta = \delta^*$. Then D can be considered as a quantum algorithm A_{asy} that breaks the OW-CPA security of Π^{asy}. Therefore,

$$\Pr[(\delta, d) \in \mathcal{S} : (\delta, d) \leftarrow D(pk, c^*)] = \mathrm{Adv}^{\text{OW-CPA}}_{A_{asy},\Pi^{asy}}.$$

The running time of B is $\mathrm{Time}(B) = \mathrm{Time}(A) + \mathrm{T}_G + \mathrm{T}_H + \mathrm{Time}(\mathrm{U}_{\mathrm{Ext}})$, where $\mathrm{T}_G = O\left((q_G + 2q_D)^2\right)$, $\mathrm{T}_H = O(q_H^2)$, $\mathrm{Time}(\mathrm{U}_{\mathrm{Ext}}) = O(q_D \cdot q_H \cdot \mathrm{Time}(\mathrm{Enc}^{asy}))$. □

Summarizing equation (1) to (9), we have

$$\mathrm{Adv}^{\text{IND-qCCA}}_{A,\Pi} \leq q_D \cdot \frac{12}{\sqrt{2^\gamma}} + 2(d+1)\sqrt{\mathrm{Adv}^{\text{OW-CPA}}_{A_{asy},\Pi^{asy}}} + 4d \cdot \mathrm{Adv}^{\text{OW-CPA}}_{A_{asy},\Pi^{asy}} + \mathrm{Adv}^{\mathrm{OT}}_{A_{sy},\Pi^{sy}}.$$

□

Furthermore, compared with Zhandry's proof for FO transformation, we notice that the plaintext extraction procedure in this proof acts the same as the decryption procedure defined in Hybrid 4 in his proof on input (c, d) such that $c \neq c^*$. With Theorem 5, we can prove that any polynomial time quantum adversary distinguishes Hybrid 1 from Hybrid 4 with a negligible probability. On the other hand, by equation (2), it seems unnecessary to restrict that the decryption oracle outputs $\perp$ directly for query (c, d) such that $c = c^*$.

4.2 REACT: from OW-qPCA to IND-qCCA in the QROM

Let $\Pi^{asy} = (\mathrm{Gen}^{asy}, \mathrm{Enc}^{asy}, \mathrm{Dec}^{asy})$ be a PKE with key space $\mathcal{K}^{asy}$, message space $\mathcal{M}^{asy}$, randomness space $\mathcal{R}^{asy}$ and ciphertext space $\mathcal{C}^{asy}$. Let $\Pi^{sy} = (\mathrm{Enc}^{sy}, \mathrm{Dec}^{sy})$ be a SKE with message space $\mathcal{M}^{sy}$, ciphertext space $\mathcal{C}^{sy}$, key space $\mathcal{K}^{sy}$. Let $H : \{0,1\}^* \to \{0,1\}^n$ and $G : \{0,1\}^* \to \mathcal{R}^{sy}$ be hash functions. We recall the REACT transformation in the following definition, and then provide its IND-qCCA security proof.

Definition 6. $\mathsf{REACT}[\Pi^{asy}, \Pi^{sy}, H, G] = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *obtained from the REACT transformation is constructed as in Fig. 5.*

$\underline{\mathrm{Gen}}$	$\underline{\mathrm{Enc}(pk, m; (R, r))}$	$\underline{\mathrm{Dec}(sk, (c_1, c_2, c_3))}$
$(pk, sk) \leftarrow \mathrm{Gen}^{asy}$	$c_1 := \mathrm{Enc}^{asy}(pk, R; r)$	$R := \mathrm{Dec}^{asy}(sk, c_1)$
return (pk, sk)	$c_2 := \mathrm{Enc}^{sy}(G(R), m)$	$m := \mathrm{Dec}^{sy}(G(R), c_2)$
	$c_3 := H(R, m, c_1, c_2)$	**if** $R = \perp$ **or** $m = \perp$
	return (c_1, c_2, c_3)	**return** $\perp$
		$c_3' := H(R, m, c_1, c_2)$
		if $c_3' \neq c_3$, **return** $\perp$
		return m

Fig. 5. PKE $\mathsf{REACT}[\Pi^{asy}, \Pi^{sy}, H, G]$ obtained from REACT transformation

Lemma 8. *Let H be the random oracle, then $\mathsf{REACT}[\Pi^{asy}, \Pi^{sy}, H, G]$ is an oracle-masked scheme relative to H, and its parameter η is $1/2^n$.*

Proof. We define deterministic polynomial time algorithm A_1, A_2, A_3 and A_4:

- A_1 takes pk, (R, r) and m as input, evaluates $c_1 := \mathrm{Enc}^{asy}(pk, R; r)$, $k := G(R)$, $c_2 := \mathrm{Enc}^{sy}(k, m)$, and then outputs (R, m, c_1, c_2).
- A_2 on input (R, m, c_1, c_2) and $y \in \{0,1\}^n$, lets $c_3 := y$ and outputs (c_1, c_2, c_3).
- A_3 takes sk and (c_1, c_2, c_3) as input, computes $R := \mathrm{Dec}^{asy}(sk, c_1)$. If $R = \perp$, output $\perp$. Else, compute $k := G(R)$ and $m := \mathrm{Dec}^{sy}(k, c_2)$. If $m = \perp$, output $\perp$. Otherwise, output (R, m, c_1, c_2).
- A_4 on input (R, m, c_1, c_2), outputs m directly.

We can verify that with four algorithms defined as above, algorithm Enc and Dec given in Fig. 5 are written as $\mathrm{Enc}^{\mathcal{O}}$ and $\mathrm{Dec}^{\mathcal{O}}$ in Definition 3 with $\mathcal{O} = H$. And thus Π is an oracle-masked scheme, and its η is

$$\begin{aligned}\eta &= \max_{(pk,sk),(c_1,c_2,c_3)} 1/2^n \left|\{y \in \{0,1\}^n : (c_1,c_2,c_3) = \mathrm{A}_2(pk, \mathrm{A}_3(sk,(c_1,c_2,c_3)), y)\}\right| \\ &= \max_{(pk,sk),(c_1,c_2,c_3)} 1/2^n \left|\{y \in \{0,1\}^n : c_3 = y\}\right| = 1/2^n\,,\end{aligned}$$

where (pk, sk) is generated by Gen, $(c_1, c_2, c_3) \in \mathcal{C}^{asy} \times \mathcal{C}^{sy} \times \{0,1\}^n$ is such that $\mathrm{A}_3(sk,(c_1,c_2,c_3)) \neq \perp$. □

Theorem 7. *For any adversary A against the* IND-qCCA *security of $\Pi = \mathsf{REACT}[\Pi^{asy}, \Pi^{sy}, H, G]$ in the QROM, making at most q_D queries to the decryption oracle, at most q_G queries to random oracle G and at most q_H queries to random oracle H, there exist an adversary A_{asy} against the* OW-qPCA *security of Π^{asy} and an adversary A_{sy} against the* OT *security of Π^{sy} such that*

$$\mathrm{Adv}^{\text{IND-qCCA}}_{A,\Pi} \leq q_D \cdot \frac{12}{\sqrt{2^n}} + 2(d+1)\sqrt{\mathrm{Adv}^{\text{OW-qPCA}}_{A_{asy},\Pi^{asy}}} + 4d \cdot \mathrm{Adv}^{\text{OW-qPCA}}_{A_{asy},\Pi^{asy}} + \mathrm{Adv}^{\text{OT}}_{A_{sy},\Pi^{sy}}\,,$$

where $d = q_H + q_G + 2q_H \cdot q_D$, $\mathrm{Time}(A_{sy}) \approx \mathrm{Time}(A_{asy}) \approx \mathrm{Time}(A) + O(d^2)$.

The IND-qCCA security proof of REACT transformation essentially follows the proof outline for FO transformation, which is presented in the proof of Theorem 6. Thus, we present the proof of Theorem 7 in the full version [25].

4.3 T_{CH}: from OW-qPCA to IND-qCCA in the QROM

Transformation T_{CH} transforms a OW-PCA secure PKE to a q-IND-CCA[2] secure KEM in the quantum random oracle model [16].

Let $\Pi^{asy} = (\mathrm{Gen}^{asy}, \mathrm{Enc}^{asy}, \mathrm{Dec}^{asy})$ be a PKE with message space $\mathcal{M}^{asy}$. Let $H, G : \{0,1\}^* \to \{0,1\}^n$ be hash functions. We then introduce T_{CH} and a new transformation $\widetilde{\mathsf{T}}$ to prove the IND-qCCA security of T_{CH}.

Definition 7. *PKE $\widetilde{\mathsf{T}}[\Pi^{asy}, H] = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ and KEM $\mathsf{T}_{\mathsf{CH}}[\Pi^{asy}, H, G] = (\mathrm{Gen}, \mathrm{Encaps}, \mathrm{Decaps})$ are as shown in Fig. 6, respectively. Particularly, T_{CH} is composited of transformation $\widetilde{\mathsf{T}}$ and modular* FO *transformation $\mathsf{U}^{\perp}_m$, i.e., $\mathsf{T}_{\mathsf{CH}}[\Pi^{asy}, H, G] = \mathsf{U}^{\perp}_m[\widetilde{\mathsf{T}}[\Pi^{asy}, H], G]$.*

Lemma 9. *$\widetilde{\mathsf{T}}[\Pi^{asy}, H]$ is an oracle-masked scheme relative to random oracle H, and its parameter η is $1/2^n$.*

Proof. Tuple $(\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{A}_4)$, as the decomposition of scheme $\widetilde{\mathsf{T}}[\Pi^{asy}, H]$, is defined as follows.

- A_1 takes pk, m and r as input, computes $c_1 := \mathrm{Enc}^{asy}(pk, m; r)$, then outputs (m, c_1).

[2] Here q is a constant and indicates q classical decryption queries.

$\underline{\text{Gen}}$	$\underline{\text{Enc}(pk, m; r)}$	$\underline{\text{Dec}(sk, (c_1, c_2))}$
$(pk, sk) \leftarrow \text{Gen}^{asy}$	$c_1 := \text{Enc}^{asy}(pk, m; r)$	$m' := \text{Dec}^{asy}(sk, c_1)$
return (pk, sk)	$c_2 := H(m, c_1)$	**if** $H(m', c_1) \neq c_2$
	return (c_1, c_2)	**return** $\perp$
		return m'

$\underline{\text{Gen}}$	$\underline{\text{Encaps}(pk)}$	$\underline{\text{Decaps}(sk, (c_1, c_2))}$
$(pk, sk) \leftarrow \text{Gen}^{asy}$	$m \xleftarrow{\$} \mathcal{M}^{asy}$	$m' := \text{Dec}^{asy}(sk, c_1)$
return (pk, sk)	$c_1 \leftarrow \text{Enc}^{asy}(pk, m)$	**if** $H(m', c_1) \neq c_2$
	$c_2 := H(m, c_1)$	**return** $\perp$
	$K := G(m)$	**return** $G(m')$
	return $(K, (c_1, c_2))$	

Fig. 6. PKE $\widetilde{\mathsf{T}}[\Pi^{asy}, H]$ and KEM $\mathsf{T}_{\mathsf{CH}}[\Pi^{asy}, H, G]$

- A_2 takes (m, c_1) and $c_2 \in \{0,1\}^n$ as input, then outputs (c_1, c_2).
- A_3 takes (c_1, c_2) as input, evaluates $m := \text{Dec}^{asy}(sk, c_1)$. If $m = \perp$, output $\perp$. Otherwise, output (m, c_1).
- A_4 on input (m, c_1), outputs m.

Then its parameter η is calculated by

$$\begin{aligned}\eta &= \max_{(pk,sk),(c_1,c_2)} 1/2^n \cdot |\{y \in \{0,1\}^n : (c_1, c_2) = \text{A}_2(pk, \text{A}_3(sk, (c_1, c_2)), y)\}| \\ &= \max_{(pk,sk),(c_1,c_2)} 1/2^n \cdot |\{y \in \{0,1\}^n : c_2 = y\}| = 1/2^n ,\end{aligned}$$

where (pk, sk) and $(c_1, c_2) \in \mathcal{C}^{asy} \times \{0,1\}^n$ are such that $\text{A}_3(sk, (c_1, c_2)) \neq \perp$. □

Theorem 8. *If Π^{asy} is δ-correct, for any adversary A against the* IND-qCCA *security of $\Pi = \mathsf{T}_{\mathsf{CH}}[\Pi^{asy}, H, G]$ in the QROM, making at most q_D queries to decapsulation oracle* Decaps*, at most q_H queries to random oracle H and at most q_G queries to random oracle G, there exists an adversary A_{asy} against the* OW-qPCA *security of Π^{asy} such that*

$$\text{Adv}^{\text{IND-qCCA}}_{A,\Pi} \leq q_D \cdot \frac{24}{\sqrt{2^n}} + 4(d+1)\sqrt{\text{Adv}^{\text{OW-qPCA}}_{A_{asy},\Pi^{asy}}} + 4d \cdot \text{Adv}^{\text{OW-qPCA}}_{A_{asy},\Pi^{asy}} ,$$

where $d = q_D + q_H + q_G$, $\text{Time}(A_{asy}) \approx \text{Time}(A) + O\left(d^2\right)$.

Proof. **Game 0**: This game is exactly $\text{Game}^{\text{IND-qCCA}}_{A,\Pi}$, that is given in Fig. 7. Then we have

$$\left|\Pr[\textbf{Game 0} \to 1] - \frac{1}{2}\right| = \text{Adv}^{\text{IND-qCCA}}_{A,\Pi} .$$

Starting from **Game 1**, random oracle H is simulated with CStO and its database register is denoted by D.

Game 1: In this game, we replace decapsulation oracle Decaps with oracle Decaps_1. Decaps_1 replies quantum query $|(c_1, c_2), z\rangle$ in three steps:

$\underline{\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}}$
$H \xleftarrow{\$} \Omega_H,\ G \xleftarrow{\$} \Omega_G$
$(pk, sk) \leftarrow \mathrm{Gen}$
$b \xleftarrow{\$} \{0,1\},\ m^* \xleftarrow{\$} \mathcal{M}^{asy}$
$c_1^* \leftarrow \mathrm{Enc}^{asy}(pk, m^*),\ c_2^* := H(m^*, c_1^*)$
$K_0^* := G(m^*),\ K_1^* \xleftarrow{\$} \{0,1\}^n$
$b' \leftarrow A^{H,G,\mathrm{Decaps}_{(c_1^*,c_2^*)}}(pk, K_b^*, (c_1^*, c_2^*))$
return $[b = b']$

$\underline{\mathrm{Decaps}_a(sk, (c_1, c_2))}$
if $(c_1, c_2) = a$, **return** $\perp$
$m' := \mathrm{Dec}^{asy}(sk, c_1)$
if $H(m', c_1) \neq c_2$, **return** $\perp$
return $G(m')$

Fig. 7. $\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi}$ for T_{CH} transformation, where oracle H, G and Decaps are all quantum-accessible

1. Perform the plaintext extraction procedure $\mathrm{U}_{\mathrm{Ext}}$ of $\widetilde{\mathsf{T}}[\Pi^{asy}, H]$ to obtain m.
2. If $m = \perp$, return $|(c_1, c_2), z \oplus \perp\rangle$. Otherwise, return $|(c_1, c_2), z \oplus G(m)\rangle$.
3. Perform $\mathrm{U}_{\mathrm{Ext}}$ a second time to uncompute m.

Note that the construction of $\mathrm{U}_{\mathrm{Ext}}$ of $\widetilde{\mathsf{T}}[\Pi^{asy}, H]$ is presented in appendix A. We then can construct Decaps_1 by invoking plaintext checking oracle $\textsc{Pco}$, instead of using sk directly.

That Decaps_1 answers q_D decapsulation queries requires performing plaintext extraction procedure $2q_D$ times. By applying Theorem 5,

$$|\Pr[\textbf{Game 0} \to 1] - \Pr[\textbf{Game 1} \to 1]| \leq 10q_D \cdot \sqrt{\eta} = q_D \cdot \frac{10}{\sqrt{2^n}}\,.$$

Game 2: In this game, we change oracle Decaps_1 by Decaps_2. Decaps_2 differs from Decaps_1 only after the challenge query: Decaps_2 performs $\mathsf{StdDecomp}_{(m^*,c_1^*)}$ on register D before and after applying Decaps_1.

To consider the commutativity of $\mathsf{StdDecomp}_{(m^*,c_1^*)}$ and Decaps_1, note that the second step of Decaps_1 commutes with $\mathsf{StdDecomp}_{(m^*,c_1^*)}$. Then by Lemma 4, the first and last step commute with $\mathsf{StdDecomp}_{(m^*,c_1^*)}$ by a loss. Therefore,

$$|\Pr[\textbf{Game 1} \to 1] - \Pr[\textbf{Game 2} \to 1]| \leq 14q_D \cdot \sqrt{\eta} = q_D \cdot \frac{14}{\sqrt{2^n}}\,.$$

Game 3: In this game, we change the process of replying random oracle queries: When random oracles are queried in the execution of A, we query a constant zero function E and then query these random oracles. Then we have

$$\Pr[\textbf{Game 2} \to 1] = \Pr[\textbf{Game 3} \to 1]\,.$$

Game 4: In this game, the only change is that the semi-classical oracle $\mathcal{O}_{\mathcal{S}}^{SC}$ is applied before querying E, where set $\mathcal{S} = \{m^*, m^* \| \cdot\}$.

E is queried at most $q_D + q_H + q_G$ times. We let $d = q_D + q_H + q_G$, and apply Theorem 3 to obtain

$$|\Pr[\textbf{Game 3} \to 1] - \Pr[\textbf{Game 4} \to 1]| \leq \sqrt{(d+1)\Pr[\mathrm{Find} : \textbf{Game 4}]}\,.$$

Game 5: In this game, we pick $c_2^* \in \{0,1\}^n$ and $K_0^* \in \{0,1\}^n$ uniformly to produce (c_1^*, c_2^*) and K^*. And we replace Decaps_2 with Decaps_1.

By similar analysis in the proof of Theorem 6, the process of oracle Decaps_2 in **Game 4** does not disturb the database state on (m^*, c_1^*) if Find does not occur. Moreover, **Game 4** and **Game 5** are indistinguishable for adversary A until Find occurs. Thus,

$$\Pr[\text{Find} : \textbf{Game 4}] = \Pr[\text{Find} : \textbf{Game 5}],$$
$$\Pr[\neg\text{Find} \wedge \textbf{Game 4} \to 1] = \Pr[\neg\text{Find} \wedge \textbf{Game 5} \to 1].$$

Furthermore,

$$\Pr[\text{Find} : \textbf{Game 5}] \leq 4d \cdot \text{Adv}^{\text{OW-qPCA}}_{A_{asy}, \Pi^{asy}},$$

where adversary A_{asy} invokes A and breaks the OW-qPCA security of Π^{asy}. The running time of A_{asy} is $\text{Time}(A_{asy}) \approx \text{Time}(A) + O\left(d^2\right)$.

Game 6: In this game, $\mathcal{O}_S^{SC}$ is removed from the process of E.

The output difference of **Game 5** and **Game 6** is bounded by Theorem 3. And in **Game 6**, K_0^* and K_1^* are both chosen from $\{0,1\}^n$ uniformly, which means that **Game 6** outputs 1 with probability 1/2.

Summarizing the above arguments, we obtain

$$\text{Adv}^{\text{IND-qCCA}}_{A,\Pi} \leq q_D \cdot \frac{12}{\sqrt{2^n}} + 4(d+1)\sqrt{\text{Adv}^{\text{OW-qPCA}}_{A_{asy}, \Pi^{asy}}} + 4d \cdot \text{Adv}^{\text{OW-qPCA}}_{A_{asy}, \Pi^{asy}}.$$

□

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

A The Construction of $\mathbf{U_{Ext}}$

To implement U_{Ext}, we first give some notations, then introduce algorithm **Extract**, as a primitive of U_{Ext}, and finally present the construction of U_{Ext}.

As is shown in definition 4, $\mathcal{O}$ is simulated by CStO and we introduce two definitions related to database D: For any $c \in \mathcal{C}$, a completion in D is defined to be a pair $(x, y) \in D$ such that $\text{A}_2(pk, x, y) = c$ and $\text{A}_3(sk, c) = x$. Define D_c to be the subset of D such that $\text{A}_2(pk, x, y) = c$ for any (x, y) in D_c. Then any completion of c in set D is necessarily in set D_c. Note that D contains at most one completion of c, since c determines $\text{A}_3(sk, c)$.

Define relation $\mathcal{R}_1(pk, sk)$ and $\mathcal{R}_2(pk, sk)$ for any (pk, sk) of Π as below.

$$\mathcal{R}_1(pk, sk) := \{(x, c) \in \mathcal{X} \times \mathcal{C} : \exists\, y \in \mathcal{Y} \text{ s.t. } \text{A}_2(pk, x, y) = c\},$$

$$\mathcal{R}_2(pk, sk) := \{(x, c) \in \mathcal{X} \times \mathcal{C} : \text{A}_3(sk, c) = x\},$$

where $\mathcal{X}$ is the output space of algorithm A_1. And we give the definition of the verification oracle $\mathbf{V}(pk, sk, \cdot, \cdot)$ of Π. $\mathbf{V}(pk, sk, \cdot, \cdot)$ takes input $(x, c) \in \mathcal{X} \times \mathcal{C}$ and outputs a bit $b \in \{0, 1\}$. For any $(x, c) \in \mathcal{R}_1(pk, sk)$, $\mathbf{V}(pk, sk, x, c) = 1$ if and only if $(x, c) \in \mathcal{R}_2(pk, sk)$.

Next, we define a classical algorithm **Extract**. **Extract** takes pk, sk, c and D as input. It looks for a completion of c in D. If a completion $(x, y) \in D$ is found, **Extract** outputs $(1, x)$. Otherwise, it outputs $(0, 0)$.

Then we give a construction of **Extract** relative to oracle $\mathbf{V}$. **Extract** on input c and D, finds a completion in two steps: For each pair (x, y) in D, it computes $c' = \mathrm{A}_2(pk, x, y)$ and compares c' with c for equality to check whether $(x, y) \in D_c$. Then to extract a completion from D_c, it invokes $\mathbf{V}$ and computes $\mathbf{V}(pk, sk, x, y)$ for each pair $(x, y) \in D_c$. If $(x, y) \in D$ exists such that $\mathbf{V}(pk, sk, x, y) = 1$, **Extract** outputs $(1, x)$. Otherwise, it outputs $(0, 0)$.

Then we construct $\mathrm{U}_{\mathrm{Ext}}$ with **Extract**, and we start with the case when the challenge query does not happen.

1. Evaluate $(b, x) = \mathbf{Extract}(pk, sk, c, D)$ in superposition and xor the output into a newly created register.
2. Apply the following conditional procedures in superposition:
3. Condition on $b = 0$, evaluate the map $|c, z, D, b, x\rangle \mapsto |c, z \oplus \perp, D, b, x\rangle$.
4. Condition on $b = 1$, evaluate the map $|c, z, D, b, x\rangle \mapsto |c, z \oplus \mathrm{A}_4(x), D, b, x\rangle$.
5. Uncompute (b, x) by evaluating $\mathbf{Extract}(pk, sk, c, D)$ in superposition again. Then discord the new register.

After the challenge query, the challenge ciphertext c^* is produced and $\mathrm{U}_{\mathrm{Ext}}$ is implemented below.

1. Apply the following conditional procedures in superposition:
2. Condition on $c = c^*$, evaluate the map $|c, z, D\rangle \mapsto |c, z \oplus \perp, D\rangle$.
3. Condition on $c \neq c^*$, apply the procedure in the case when c^* is undefined.

In addition, the running time of $\mathrm{U}_{\mathrm{Ext}}$ is upper bounded as follows. Denote the length of database by l. For each database D, $|D| \leq l$ and **Extract** invokes A_2 and $\mathbf{V}$ at most l times during the execution. Thus $O(l \cdot \mathrm{Time}(\mathrm{A}_2) + l \cdot \mathrm{Time}(\mathbf{V}))$ is an upper bound of the running time of $\mathrm{U}_{\mathrm{Ext}}$.

Then we will give respective constructions of $\mathrm{U}_{\mathrm{Ext}}$ for $\mathsf{FO}[\Pi^{asy}, \Pi^{sy}, H, G]$, $\mathsf{REACT}[\Pi^{asy}, \Pi^{sy}, H, G]$ and $\widetilde{\mathsf{T}}[\Pi^{asy}, H]$. Since the implementation of $\mathbf{V}$ is sufficient to determine the construction of $\mathrm{U}_{\mathrm{Ext}}$ for an oracle-masked scheme Π, we only give constructions of the verification oracle $\mathbf{V}$ for these three schemes.

A.1 The Construction of $\mathbf{U}_{\mathbf{Ext}}$ for FO

For scheme $\Pi = \mathsf{FO}[\Pi^{asy}, \Pi^{sy}, H, G]$, we first present relation $\mathcal{R}_1(pk, sk)$ and $\mathcal{R}_2(pk, sk)$ to determine the input form of the verification oracle $\mathbf{V}$, then give an implementation of $\mathbf{V}$.

By Lemma 5, relation $\mathcal{R}_1(pk, sk)$ and $\mathcal{R}_2(pk, sk)$ are subsets of $\mathcal{M}^{asy} \times \mathcal{C}^{sy} \times \mathcal{C}^{asy} \times \mathcal{C}^{sy}$ for any (pk, sk) of Π. Tuple $(\delta, d_1, c, d_2) \in \mathcal{R}_1(pk, sk)$ if $d_1 = d_2$ and

$r \in \mathcal{R}^{asy}$ exists such that $c := \mathrm{Enc}^{asy}(pk, \delta; r)$. Tuple $(\delta, d_1, c, d_2) \in \mathcal{R}_2(pk, sk)$ if $d_1 = d_2$ and $\mathrm{Dec}^{asy}(sk, c) = \delta$.

Further, tuple $(\delta, d_1, c, d_2) \in \mathcal{R}_1(pk, sk)$ also satisfies $\mathrm{Dec}^{asy}(sk, c) = \delta$ by the correctness of Π^{asy}, and thus $(\delta, d_1, c, d_2) \in \mathcal{R}_2(pk, sk)$. Then $\mathcal{R}_1(pk, sk)$ is a subset of $\mathcal{R}_2(pk, sk)$. By similar arguments, we also conclude that $(\delta, d_1, c, d_2) \notin \mathcal{R}_1(pk, sk)$ implies $(\delta, d_1, c, d_2) \notin \mathcal{R}_2(pk, sk)$ for any (pk, sk). Thus for any (pk, sk) of Π, $\mathcal{R}_1(pk, sk) = \mathcal{R}_2(pk, sk)$ and

$$\mathcal{R}_2(pk, sk) = \{(\delta, d, c, d) : c \in \mathcal{C}^{asy}, \delta = \mathrm{Dec}^{asy}(sk, c), d \in \mathcal{C}^{sy}\}.$$

By the definition of the verification oracle, $\mathbf{V}$ for Π can be simply simulated by an algorithm that takes as input tuple (δ, d_1, c, d_2) and trivially outputs 1. Moreover, notice that sk is not used in the construction of $\mathrm{U}_{\mathrm{Ext}}$ except for the verification oracle. Therefore, $\mathrm{U}_{\mathrm{Ext}}$ for Π can be implemented without sk.

Finally, the running time of $\mathrm{U}_{\mathrm{Ext}}$ is given by $O(l \cdot \mathrm{Time}(\mathrm{Enc}^{asy}))$.

A.2 The Construction of $\mathbf{U_{Ext}}$ for REACT

For scheme $\Pi = \mathsf{REACT}[\Pi^{asy}, \Pi^{sy}, H, G]$, we only give an implementation of oracle $\mathbf{V}$ here.

By Lemma 8, $\mathcal{R}_1(pk, sk)$ and $\mathcal{R}_2(pk, sk)$ are subsets of $\mathcal{M}^{asy} \times \mathcal{M}^{sy} \times \mathcal{C}^{asy} \times \mathcal{C}^{sy} \times \mathcal{C}^{asy} \times \mathcal{C}^{sy} \times \{0,1\}^n$ for any (pk, sk). Any tuple $(R, m, c_1, c_2, c_1', c_2', c_3') \in \mathcal{R}_1(pk, sk)$ if $c_1 = c_1'$, $c_2 = c_2'$. And this tuple is an element of $\mathcal{R}_2(pk, sk)$ if $R = \mathrm{Dec}^{asy}(sk, c_1')$, $m = \mathrm{Dec}^{sy}(G(R), c_2')$, $c_1 = c_1'$, $c_2 = c_2'$. Thus, we have $\mathcal{R}_1(pk, sk) = \{(R, m, c_1, c_2, c_1, c_2, c_3) : \mathcal{R} \in \mathcal{M}^{asy}, m \in \mathcal{M}^{sy}, c_1 \in \mathcal{C}^{asy}, c_2 \in \mathcal{C}^{sy}, c_3 \in \{0,1\}^n\}$ and $\mathcal{R}_2(pk, sk) = \{(R, m, c_1, c_2, c_1, c_2, c_3) : c_1 \in \mathcal{C}^{asy}, c_2 \in \mathcal{C}^{sy}, c_3 \in \{0,1\}^n, R = \mathrm{Dec}^{asy}(sk, c_1), m = \mathrm{Dec}^{sy}(G(R), c_2)\}$. Then we assume the input form of $\mathbf{V}$ to be $(R, m, c_1, c_2, c_1, c_2, c_3)$ according to $\mathcal{R}_1(pk, sk)$ of Π.

We present an algorithm $\mathbf{V}_{\mathrm{Sim}}$ relative to plaintext checking oracle $\textsc{Pco}$. $\mathbf{V}_{\mathrm{Sim}}$ takes as input tuple $(R, m, c_1, c_2, c_1, c_2, c_3)$. It first invokes $\textsc{Pco}$ and obtain $b := \textsc{Pco}(R, c_1)$. If $b = 0$, $\mathbf{V}_{\mathrm{Sim}}$ outputs 0. Else, it computes $m' := \mathrm{Dec}^{sy}(G(R), c_2)$. If $m \neq m'$, output 0. Else, output 1. Then by the definition of $\textsc{Pco}$ in appendix B.2, it is easily verified that $\mathbf{V}$ can be simulated by $\mathbf{V}_{\mathrm{Sim}}$. In this way, $\mathrm{U}_{\mathrm{Ext}}$ for Π is implemented by invoking $\textsc{Pco}$ instead of using sk directly. Moreover, the running time of $\mathrm{U}_{\mathrm{Ext}}$ is given by $O(l)$.

A.3 The Construction of $\mathbf{U_{Ext}}$ for $\widetilde{\mathsf{T}}$

For scheme $\widetilde{\mathsf{T}}[\Pi^{asy}, H]$, we give a straightforward way to simulate oracle $\mathbf{V}$ here.

According to Lemma 9, tuple $((m, c_1), (c_1', c_2')) \in \mathcal{R}_1(pk, sk)$ if $c_1 = c_1'$, while tuple $((m, c_1), (c_1', c_2')) \in \mathcal{R}_2(pk, sk)$ if $c_1 = c_1'$ and $m = \mathrm{Dec}^{asy}(sk, c_1)$. Then we can assume the input form of $\mathbf{V}$ to be (m, c_1, c_1, c_2).

We construct an oracle $\mathbf{V}_{\mathrm{Sim}}$ relative to plaintext-checking oracle $\textsc{Pco}$ and use it to simulate $\mathbf{V}$. On input (m, c_1, c_1, c_2), $\mathbf{V}_{\mathrm{Sim}}$ first invokes $\textsc{Pco}$ and obtains $b := \textsc{Pco}(m, c_1)$. If $b = 0$, it outputs 0. Otherwise, it outputs 1. Then $\mathrm{U}_{\mathrm{Ext}}$ can be implemented without sk, and its running time is $O(l)$.

B Cryptographic Primitives

Here we introduce secret-key encryption schemes (SKE), public-key encryption schemes (PKE), key encapsulation mechanisms (KEM) and their security notions.

B.1 Secret-Key Encryption

Definition 8. *A SKE Π^{sy} consists of a pair of polynomial-time algorithms* (E, D) *as follows.*

1. E*, the encryption algorithm, takes as input a message m and a key k, and outputs a ciphertext c.*
2. D*, the decryption algorithm, on input a ciphertext c and a key k outputs either a message m or a special symbol $\perp$ if c is invalid.*

Let $\Pi^{sy} = (\mathrm{E}, \mathrm{D})$ be a SKE and define one-time (OT) security for it.

Definition 9 (OT). *Define the advantage of adversary A against the* OT *security of Π^{sy} as* $\mathrm{Adv}^{\mathrm{OT}}_{A,\Pi^{sy}} := \left|\Pr[\mathrm{Game}^{\mathrm{OT}}_{A,\Pi^{sy}} \to 1] - 1/2\right|$ *and* $\Pr[\mathrm{Game}^{\mathrm{OT}}_{A,\Pi^{sy}} \to 1]$ *is written by* $\Pr[b' = b : (m_0, m_1) \leftarrow A, b \xleftarrow{\$} \{0,1\}, c^* \leftarrow \mathrm{E}(k, m_b), b' \leftarrow A(c^*)]$. *Then Π^{sy} is* OT *secure if* $\mathrm{Adv}^{\mathrm{OT}}_{A,\Pi^{sy}}$ *is negligible for any polynomial-time adversary A.*

B.2 Public-Key Encryption

Definition 10. *A PKE Π^{asy} consists of a triple of polynomial-time algorithms* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *as follows.*

1. Gen*, the key generation algorithm, on input 1^λ outputs a public/secret keypair (pk, sk).*
2. Enc*, the encryption algorithm, on input a public key pk and a message m outputs a ciphertext c.*
3. Dec*, the decryption algorithm, on input a secret key sk and a ciphertext c outputs either a message m or a special symbol $\perp$ if c is invalid.*

Let $\Pi^{asy} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ be a PKE with message space $\mathcal{M}$. Then we introduce γ-spread and δ-correct property for it.

Definition 11 (γ-spread [12]). *Π^{asy} is γ-spread if for any pk produced by* $\mathrm{Gen}(1^\lambda)$ *and any message $m \in \mathcal{M}$,*

$$\max_{c \in \{0,1\}^*} \Pr[c' = c : c' \leftarrow \mathrm{Enc}(pk, m)] \leq 1/2^\gamma.$$

And Π^{asy} is called well-spread in λ if $\gamma = \omega(\log(\lambda))$.

Definition 12 (δ-correct [14]). *Π^{asy} is δ-correct if*

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

And Π^{asy} is called perfectly correct if $\delta = 0$.

In the following, we define one-wayness under chosen plaintext attacks (OW-CPA), one-wayness under quantum plaintext checking attacks (OW-qPCA) and indistinguishability under quantum chosen ciphertext attacks (IND-qCCA) these three security notions for Π^{asy}.

Definition 13 (OW-CPA). *The* OW-CPA *game for Π^{asy} is defined in Fig. 8. The advantage of an adversary A against the* OW-CPA *security of Π is defined to be $\mathrm{Adv}^{\text{OW-CPA}}_{A,\Pi^{asy}} := \Pr[\mathrm{Game}^{\text{OW-CPA}}_{A,\Pi^{asy}} \to 1]$. Then Π^{asy} is* OW-CPA *secure if $\mathrm{Adv}^{\text{OW-CPA}}_{A,\Pi^{asy}}$ is negligible for any polynomial-time adversary A.*

Definition 14 (OW-qPCA [17]). *The* OW-qPCA *game for Π^{asy} is defined in Fig. 8. The advantage of an adversary A against the* OW-qPCA *security of Π^{asy} is defined as $\mathrm{Adv}^{\text{OW-qPCA}}_{A,\Pi^{asy}} := \Pr[\mathrm{Game}^{\text{OW-qPCA}}_{A,\Pi^{asy}} \to 1]$. Π^{asy} is* OW-qPCA *secure if $\mathrm{Adv}^{\text{OW-qPCA}}_{A,\Pi^{asy}}$ is negligible for any polynomial-time adversary A.*

$\underline{\mathrm{Game}^{\text{OW-ATK}}_{A,\Pi^{asy}}}$
$(pk, sk) \leftarrow \mathrm{Gen}$
$m^* \xleftarrow{\$} \mathcal{M}$
$c^* \leftarrow \mathrm{Enc}(pk, m^*)$
$m' \leftarrow A^{\mathcal{O}_{\mathrm{ATK}}}(pk, c^*)$
return $[m = m']$

$\underline{\mathrm{Pco}(m, c)}$
$m' := \mathrm{Dec}(sk, c)$
return $[m = m']$

ATK	CPA	qPCA
$\mathcal{O}_{\mathrm{ATK}}$	$\perp$	Pco

Fig. 8. Game OW-ATK for Π^{asy} (ATK $\in$ {CPA, qPCA}), where oracle $\mathcal{O}_{\mathrm{ATK}}$ is quantum-accessible.

Definition 15 (IND-qCCA [5]). *The* IND-qCCA *game for Π^{asy} is defined in Fig. 9. The advantage of an adversary A against the* IND-qCCA *security of Π^{asy} is defined as $\mathrm{Adv}^{\text{IND-qCCA}}_{A,\Pi^{asy}} := |\Pr[\mathrm{Game}^{\text{IND-qCCA}}_{A,\Pi^{asy}} \to 1] - 1/2|$. Then Π^{asy} is* IND-qCCA *secure if $\mathrm{Adv}^{\text{IND-qCCA}}_{A,\Pi^{asy}}$ is negligible for any polynomial-time adversary A.*

B.3 Key Encapsulation

Definition 16. *A KEM Π^{kem} consists of a triple of polynomial-time algorithms* (Gen, Encaps, Decaps) *as follows.*

$\underline{\text{Game}^{\text{IND-qCCA}}_{A,\Pi^{asy}}}$	$\underline{\text{Game}^{\text{IND-qCCA}}_{A,\Pi^{kem}}}$	$\underline{\text{Dec}_a(sk,c)}$
$(pk,sk) \leftarrow \text{Gen}$	$(pk,sk) \leftarrow \text{Gen}$	**if** $c = a$, **return** $\perp$
$(m_0,m_1) \leftarrow A^{\text{Dec}_\perp}(pk)$	$b \xleftarrow{\$} \{0,1\}$	$m' := \text{Dec}(sk,c)$
$b \xleftarrow{\$} \{0,1\}$	$(K_0^*,c^*) \leftarrow \text{Encaps}(pk)$	**return** m'
$c^* \leftarrow \text{Enc}(pk,m_b)$	$K_1 \xleftarrow{\$} \mathcal{K}$	$\underline{\text{Decaps}_a(sk,c)}$
$b' \leftarrow A^{\text{Dec}_{c^*}}(pk,c^*)$	$b' \leftarrow A^{\text{Decaps}_{c^*}}(pk,K_b^*,c^*)$	**if** $c = a$, **return** $\perp$
return $[m = m']$	**return** $[b = b']$	$K := \text{Decaps}(sk,c)$
		return K

Fig. 9. Game IND-qCCA for Π^{asy} and Π^{kem}, where oracle Dec_a and Decaps_a are both quantum-accessible.

1. Gen, *the key generation algorithm,on input* 1^λ *outputs a public/secret key-pair* (pk,sk).
2. Encaps, *the encapsulation algorithm, takes as input a public key* pk *and outputs a ciphertext* c *and a key* k.
3. Decaps, *the decapsulation algorithm, on input a secret key* sk *and a ciphertext* c *outputs either a key* k *or a special symbol* $\perp$ *if* c *is invalid.*

Let $\Pi^{kem} = (\text{Gen}, \text{Encaps}, \text{Decaps})$ be a KEM and define IND-qCCA security for it.

Definition 17 (IND-qCCA [27]). *The* IND-qCCA *game for* Π^{kem} *is defined in Fig. 9. The advantage of an adversary* A *against the* IND-qCCA *security of* Π^{kem} *is defined as* $\text{Adv}^{\text{IND-qCCA}}_{A,\Pi^{kem}} := |\Pr[\text{Game}^{\text{IND-qCCA}}_{A,\Pi^{kem}} \rightarrow 1] - 1/2|$. *Then* Π^{kem} *is* IND-qCCA *secure if* $\text{Adv}^{\text{IND-qCCA}}_{A,\Pi^{kem}}$ *is negligible for any polynomial-time adversary* A.

References

1. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. In: Annual International Cryptology Conference. pp. 269–295. Springer (2019). https://doi.org/10.1007/978-3-030-26951-7_10
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. pp. 62–73 (1993). https://doi.org/10.1145/168588.168596
3. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of cca security in the quantum random oracle model. In: Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II. pp. 61–90. Springer (2019). https://doi.org/10.1007/978-3-030-36033-7_3
4. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: International conference on the theory and application of cryptology and information security. pp. 41–69. Springer (2011). https://doi.org/10.1007/978-3-642-25385-0_3

5. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Annual cryptology conference. pp. 361–379. Springer (2013). https://doi.org/10.1007/978-3-642-40084-1_21
6. Chiesa, A., Manohar, P., Spooner, N.: Succinct arguments in the quantum random oracle model. In: Theory of Cryptography Conference. pp. 1–29. Springer (2019). https://doi.org/10.1007/978-3-030-36033-7_1
7. Chung, K.M., Fehr, S., Huang, Y.H., Liao, T.N.: On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 598–629. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_21
8. Coron, J.S., Handschuh, H., Joye, M., Paillier, P., Pointcheval, D., Tymen, C.: Gem: A generic chosen-ciphertext secure encryption method. In: CT-RSA. vol. 2271, pp. 263–276. Springer (2002). https://doi.org/10.1007/3-540-45760-7_18
9. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing **33**(1), 167–226 (2003). https://doi.org/10.1137/S0097539702403773
10. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Online-extractability in the quantum random-oracle model. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 677–706. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_24
11. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Annual international cryptology conference. pp. 537–554. Springer (1999). https://doi.org/10.1007/3-540-48405-1_34
12. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of cryptology **26**(1), 80–101 (2013). https://doi.org/10.1007/s00145-011-9114-1
13. Gagliardoni, T., Hülsing, A., Schaffner, C.: Semantic security and indistinguishability in the quantum world. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III. pp. 60–89. Springer (2016). https://doi.org/10.1007/978-3-662-53015-3_3
14. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki-okamoto transformation. In: Theory of Cryptography Conference. pp. 341–371. Springer (2017). https://doi.org/10.1007/978-3-319-70500-2_12
15. Hövelmanns, K., Hülsing, A., Majenz, C.: Failing gracefully: Decryption failures and the fujisaki-okamoto transform. In: Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13794, pp. 414–443. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_15
16. Huguenin-Dumittan, L., Vaudenay, S.: On ind-qcca security in the ROM and its applications - CPA security is sufficient for TLS 1.3. In: Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III. pp. 613–642. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_22
17. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Annual International Cryptology Conference. pp. 96–125. Springer (2018). https://doi.org/10.1007/978-3-319-96878-0_4

18. Jiang, H., Zhang, Z., Ma, Z.: Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In: IACR International Workshop on Public Key Cryptography. pp. 618–645. Springer (2019). https://doi.org/10.1007/978-3-030-17259-6_21
19. Jiang, H., Zhang, Z., Ma, Z.: Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In: International Conference on Post-Quantum Cryptography. pp. 227–248. Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_13
20. Kuchta, V., Sakzad, A., Stehlé, D., Steinfeld, R., Sun, S.: Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and cca security. In: Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III. pp. 703–728. Springer (2020). https://doi.org/10.1007/978-3-030-45727-3_24
21. Liu, X., Wang, M.: Qcca-secure generic key encapsulation mechanism with tighter security in the quantum random oracle model. In: IACR International Conference on Public-Key Cryptography. pp. 3–26. Springer (2021). https://doi.org/10.1007/978-3-030-75245-3_1
22. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
23. Okamoto, T., Pointcheval, D.: React: Rapid enhanced-security asymmetric cryptosystem transform. In: CT-RSA 2001. pp. 159–174. Springer (2001). https://doi.org/10.1007/3-540-45353-9_13
24. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 520–551. Springer (2018). https://doi.org/10.1007/978-3-319-78372-7_17
25. Shan, T., Ge, J., Xue, R.: Qcca-secure generic transformations in the quantum random oracle model. IACR Cryptol. ePrint Arch. p. 1235 (2022), https://eprint.iacr.org/2022/1235
26. Targhi, E.E., Unruh, D.: Post-quantum security of the fujisaki-okamoto and oaep transforms. In: Theory of Cryptography Conference. pp. 192–216. Springer (2016). https://doi.org/10.1007/978-3-662-53644-5_8
27. Xagawa, K., Yamakawa, T.: (tightly) qcca-secure key-encapsulation mechanism in the quantum random oracle model. In: International Conference on Post-Quantum Cryptography. pp. 249–268. Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_14
28. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7417, pp. 758–775. Springer (2012). https://doi.org/10.1007/978-3-642-32009-5_44
29. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Annual International Cryptology Conference. pp. 239–268. Springer (2019). https://doi.org/10.1007/978-3-030-26951-7_9, Full Version: https://eprint.iacr.org/2018/276