On the Fujisaki-Okamoto transform: from Classical CCA Security to Quantum CCA Security

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

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

May 30, 2023

Abstract

The Fujisaki-Okamoto (FO) transformation (CRYPTO 1999 and Journal of Cryptology 2013) and its KEM variants (TCC 2017) are used to construct $\mathsf{IND\text{-}CCA}$-secure PKE or KEM schemes in the random oracle model (ROM).

In the post-quantum setting, the ROM is extended to the quantum random oracle model (QROM), and the $\mathsf{IND\text{-}CCA}$ security of FO transformation and its KEM variants in the QROM has been extensively analyzed. Grubbs et al. (EUROCRYPTO 2021) and Xagawa (EUROCRYPTO 2022) then focused on security properties other than $\mathsf{IND\text{-}CCA}$ security, such as the anonymity aganist chosen-ciphertext attacks ($\mathsf{ANO\text{-}CCA}$) of FO transformation in the QROM.

Beyond the post-quantum setting, Boneh and Zhandry (CRYPTO 2013) considered quantum adversaries that can perform the quantum chosen-ciphertext attacks (qCCA). However, to the best of our knowledge, there are few results on the $\mathsf{IND\text{-}qCCA}$ or $\mathsf{ANO\text{-}qCCA}$ security of FO transformation and its KEM variants in the QROM.

In this paper, we define a class of security games called the oracle-hiding game, and provide a lifting theorem for it. This theorem lifts the security reduction of oracle-hiding games in the ROM to that in the QROM. With this theorem, we prove the $\mathsf{IND\text{-}qCCA}$ and $\mathsf{ANO\text{-}qCCA}$ security of transformation $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\perp}$, $\mathsf{FO}^{\not\perp}_m$ and $\mathsf{FO}^{\perp}_m$, which are KEM variants of FO, in the QROM.

Moreover, we prove the $\mathsf{ANO\text{-}qCCA}$ security of the hybrid PKE schemes built via the KEM-DEM paradigm, where the underlying KEM schemes are obtained by $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\perp}$, $\mathsf{FO}^{\not\perp}_m$ and $\mathsf{FO}^{\perp}_m$. Notably, for those hybrid PKE schemes, our security reduction shows that their anonymity is independent of the security of their underlying DEM schemes. Hence, our result simplifies the anonymity analysis of the hybrid PKE schemes that obtained from the FO transformation.

Keywords: quantum chosen-ciphertext attacks, quantum random oracle model, anonymity, Fujisaki-Okamoto transformation

1 Introduction

1.1 Background

Shor's breakthrough result [Sho99] shows that quantum polynomial-time (QPT) adversary can break cryptosystems based on the factoring problem and the discrete logarithm problem. This motivates researchers to generate post-quantum cryptography and design quantum-resistant cryptosystems. In the post-quantum setting, the adversaries are capable of quantum computing, in contrast to the classical computing power held by the cryptosystem users. Moreover, as introduced in [BDF+11], it is reasonable to assume that the quantum adversary can query random oracles in superposition, and the random oracle model (ROM) should be extended to the quantum random oracle model (QROM) for post-quantum consideration.

The well-known Fujisaki-Okamoto (FO) transform [FO13] is a transformation that combines a public-key encryption (PKE) scheme and a symmetric-key encryption (SKE) scheme to obtain a hybrid PKE scheme that is secure against the indistinguishability under chosen-ciphertext attacks ($\mathsf{IND\text{-}CCA}$) in the ROM. Dent [Den03] then introduced a variant of FO, whose resulting scheme is an $\mathsf{IND\text{-}CCA}$ secure key encapsulation mechanism (KEM). On the other hand, $\mathsf{IND\text{-}CCA}$ secure PKE schemes can be built via the KEM-DEM[1] paradigm with high efficiency and versatility [CS03]. Since then, it has been paid more attention to the constructions of the $\mathsf{IND\text{-}CCA}$-secure KEM.

In what follows, we also denote by $\mathsf{KEM}+\mathsf{DEM}$ the PKE scheme built via the KEM-DEM paradigm with KEM scheme KEM and DEM scheme DEM. Moreover, the scheme is denoted as $\mathcal{T}+\mathsf{DEM}$ if the underlying KEM scheme is obtained by transformation $\mathcal{T}$.

Modular treatment of FO transformation for KEM variants: Following [Den03], Hofheinz et al. [HHK17] provided a modular toolkit of transformations including T, $\mathsf{U}^{\not\perp}$, $\mathsf{U}^{\perp}$, $\mathsf{U}^{\not\perp}_m$, $\mathsf{U}^{\perp}_m$, $\mathsf{QU}^{\not\perp}_m$ and $\mathsf{QU}^{\perp}_m$. By combining T with $\mathsf{U}^{\not\perp}$, $\mathsf{U}^{\perp}$, $\mathsf{U}^{\not\perp}_m$, $\mathsf{U}^{\perp}_m$, $\mathsf{QU}^{\not\perp}_m$ and $\mathsf{QU}^{\perp}_m$, it is obtained the KEM variants of FO transformation $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\perp}$, $\mathsf{FO}^{\not\perp}_m$, $\mathsf{FO}^{\perp}_m$, $\mathsf{QFO}^{\not\perp}_m$ and $\mathsf{QFO}^{\perp}_m$, respectively. Here, $\perp$ (resp. $\not\perp$) indicates that the transformation is explicit (resp. implicit) rejection type[2] and Q means that the transformation requires an additional "key-confirmation" hash. In what follows, those KEM variants of FO transformation are referred as FO-like transformations.

FO-like transformations are widely used in the submissions to NIST post-quantum cryptography (PQC) standardization process [NIS17] starting from 2016. Among 39 Round-1 KEM submissions to the standardization process, there are 25 submissions following the FO-like transformations to achieve the $\mathsf{IND\text{-}CCA}$ security in the ROM or QROM. In July 2022, NIST announced the first group of winners [NIS22], and CRYSTALS-Kyber, as the only selected KEM scheme for standardization, uses a variant of FO-like transformation $\mathsf{FO}^{\not\perp}$.

Different security guarantees of FO-like transformations under chosen-ciphertext attacks: The classical $\mathsf{IND\text{-}CCA}$ reductions of FO-like transformations in the ROM were provided in [HHK17]. In the post-quantum setting, it has been heavily analysed the $\mathsf{IND\text{-}CCA}$ security of FO-like transformations in the QROM (e.g., [HHK17, XY19, JZC+18, JZM19, BHH+19, HKSU20, KSS+20, DFMS22, HHM22]).

In addition to the standard $\mathsf{IND\text{-}CCA}$ security, researchers have also been studying the important and useful security properties of FO-like transformations under chosen-ciphertext attacks in the post-quantum setting as follows.

- Anonymity: This property in the public-key setting was first introduced by Bellare et al. [BBDP01]. Roughly speaking, a PKE scheme is anonymous if its ciphertexts leak little information of the receiver.

  Grubbs et al. [GMP22] were the first to study anonymity in PKE/KEM for post-quantum considerations. They defined the anonymity against chosen-ciphertext attacks ($\mathsf{ANO\text{-}CCA}$) and provided the $\mathsf{ANO\text{-}CCA}$ security reductions of $\mathsf{HFO}^{\perp'}$ (a variant of $\mathsf{QFO}^{\perp}_m$) and $\mathsf{FO}^{\not\perp}$ in the QROM. Moreover, they proved the $\mathsf{ANO\text{-}CCA}$ security of PKE scheme $\mathsf{HFO}^{\perp'}+\mathsf{DEM}$ and $\mathsf{FO}^{\not\perp}+\mathsf{DEM}$ in the QROM.

  Building on the result of [GMP22], Xagawa [Xag22] investigated the anonymity of NIST PQC Round 3 KEM schemes. The core concept of this work is a new security notion called strong pseudorandomness against chosen-ciphertext attacks ($\mathsf{SPR\text{-}CCA}$).

- Robustness: This property was first introduced in [ABN10], and it means that the receiver can recognize whether a ciphertext is intended for themselves and is difficult to be deceived.

  In the post-quantum setting, Grubbs et al. [GMP22] defined the weak robustness under chosen-ciphertext attacks ($\mathsf{WROB\text{-}CCA}$) and strong robustness under chosen-ciphertext attacks ($\mathsf{SROB\text{-}CCA}$). They also proved the $\mathsf{WROB\text{-}CCA}$ and $\mathsf{SROB\text{-}CCA}$ security of PKE scheme $\mathsf{FO}^{\not\perp}+\mathsf{DEM}$ in the QROM.

[1]DEM is an abbreviation for data encapsulation mechanism. Indeed, a DEM scheme is a SKE scheme, and we will use the terms DEM and SKE interchangeably throughout this paper.

[2]The decapsulation algorithm of an implicit (resp. explicit) rejection type transformation returns a pseudorandom value (resp. an abort symbol $\perp$) when the ciphertext fails to be decrypted.

- Key dependent message (KDM) security: The KDM security was first introduced in [BRS02]. Intuitively speaking, a KDM-secure PKE scheme remains secure even if the adversary can obtain the encryption results of the secret key.

  Kitagawa and Nishimaki [KN22] initialized the study of the KDM security of PKE in the post quantum setting. They proved the key dependent message against chosen-ciphertext attacks ($\mathsf{KDM\text{-}CCA}$) security of PKE scheme $\mathsf{U}_m^{\perp,\mathsf{keyconf}}+\mathsf{DEM}$ in the QROM, where $\mathsf{U}_m^{\perp,\mathsf{keyconf}}$ is a variant of $\mathsf{QU}_m^{\perp}$.

The extension to the post-quantum security arguments: It was further assumed that quantum adversary has quantum access to secretive primitives. Especially for PKE, Boneh and Zhandry [BZ13] defined a new security notion named indistinguishability against quantum chosen-ciphertext attacks ($\mathsf{IND\text{-}qCCA}$), in which the quantum adversary is able to query the decryption oracle in superposition. They also presented the first $\mathsf{IND\text{-}qCCA}$-secure PKE scheme by the transformation defined in [BCHK07].

Following [BZ13], Xagawa and Yamakawa [XY19] introduced the $\mathsf{IND\text{-}qCCA}$ security for KEM scheme, where the adversary can make quantum queries to the decapsulation oracle. They also provided the $\mathsf{IND\text{-}qCCA}$ security reductions of transformation SXY ($\mathsf{U}_m^{\not\perp}$) and HU (an adapted version of $\mathsf{QU}_m^{\not\perp}$) in the QROM. Later, Liu and Wang [LW21] gave a tighter $\mathsf{IND\text{-}qCCA}$ security reduction of SXY from the standard security in the QROM.

Apart from the standard security, anonymity, robustness and key dependent message security, these security properties under chosen-ciphertext attacks can be extended into ones under quantum chosen-ciphertext attacks (e.g. $\mathsf{ANO\text{-}qCCA}$, $\mathsf{WROB\text{-}qCCA}$, $\mathsf{SROB\text{-}qCCA}$, $\mathsf{KDM\text{-}qCCA}$).

To the best of our knowledge, for $\mathsf{GOAL} \in \{\mathsf{ANO,WROB,SROB,KDM}\}$, the $\mathsf{GOAL\text{-}qCCA}$ security of any PKE scheme $\mathsf{KEM}+\mathsf{DEM}$ in the QROM, whose underlying KEM scheme KEM is obtained by FO-like transformations, have not yet been studied. A natural question arises.

Can we prove security properties, such as anonymity, of those PKE schemes $\mathsf{KEM}+\mathsf{DEM}$ even under quantum chosen-ciphertext attacks?

Lift classical CCA reductions to qCCA reductions: In his seminal paper [Zha19], Zhandry proposed the compressed oracle technique, which can be used to perfectly simulate quantum random oracles and "record" quantum queries on the database register without detecting. This technique can be considered the quantum counterpart of *on-the-fly* simulation, and thus makes it possible to mimic the classical security reduction in the ROM when proving security under quantum chosen ciphertext attacks (qCCA). With this technique, Zhandry proved the $\mathsf{IND\text{-}qCCA}$ security of the FO transformation in the QROM.

Based on the same technique, Don et al. [DFMS22] took the extracting action on the database register as a whole part, and provided the generic extractability result (i.e. the extractable RO-simulator and Theorem 4.3 of [DFMS22]), which can be applied to bound the loss caused by the simulation of the decryption oracle in the QROM reductions. Moreover, it was proved that $\mathsf{FO}_m^{\perp}$ is $\mathsf{IND\text{-}CCA}$-secure in the QROM.

In contrast to [DFMS22], Shan et al. [SGX23] investigated a more specific setting. Their study focused on PKE schemes that contain re-encryption computation in the decryption algorithms. In their paper, *plaintext extractor* is developed to simulate quantum decryption oracle for this type of schemes, and an upper bound of this simulation in the QROM reductions is also presented. Furthermore, several transformations, including FO and REACT, were proved to be $\mathsf{IND\text{-}qCCA}$-secure in the QROM, with concrete security bounds.

The $\mathsf{IND\text{-}qCCA}$ and $\mathsf{IND\text{-}CCA}$ reductions in the aforementioned works can be regarded as the quantum counterparts of the classical $\mathsf{IND\text{-}CCA}$ reductions of schemes, respectively. We adopt this view to prove the $\mathsf{IND\text{-}qCCA}$ security of FO-like transformations, and furthermore, to explore the $\mathsf{GOAL\text{-}qCCA}$ reductions of them for $\mathsf{GOAL} \in \{\mathsf{ANO}, \mathsf{WROB}, \mathsf{SROB}, \mathsf{KDM}\}$. This promotes the following question.

Is there a lifting theorem that straightforwardly extends the classical CCA reduction of FO-like transformations to the qCCA ones?

1.2 Our Contribution

A lifting theorem for oracle-hiding games: To answer the second question, a lifting theorem is proposed in this paper. This theorem is established on one type of games called the oracle-hiding games, as shown in Definition 1.

Definition 1 (Oracle-hiding Game, informal). *For random oracle H, G and a secret oracle O_{sk}, we call the game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$, as shown in Fig. 1, the oracle-hiding game.*

Oracle-hiding game $\mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}}$	
1, $(pk, sk) \leftarrow \mathsf{KGen}$	
2, $\mathsf{OHG.A} \leftarrow \mathcal{A}^{H,G,O_{sk}}(sk)$	
3, $\mathcal{C}$ **perform following operation**	
$m^* \xleftarrow{\$} \mathcal{R}_1$, $r \xleftarrow{\$} \mathcal{R}_2$	$O_{sk}(\alpha)$
$m_0 \leftarrow \mathsf{cha}_1(pk, \mathsf{OHG.A}, m^*, r)$	1, **If** $\mathsf{OHG.B}$ **is defined and**
$y_0 = G(m^*\|\|m_0)$	$\alpha = \mathsf{ota}_2(pk, m^*\|\|m_1, H(m^*\|\|m_1))$
$m_1 \leftarrow \mathsf{cha}_2(pk, \mathsf{OHG.A}, y_s, m^*, r)$	**return** $\perp$
$y_1 = H(m^*\|\|m_1)$	**Else return** $\mathsf{ota}^{H,G}(sk, \alpha)$
$\mathsf{OHG.B} \leftarrow \mathsf{cha}_3(pk, \mathsf{OHG.A}, y_0, y_1, m^*, r)$	
4, $\mathsf{OHG.C} \leftarrow \mathcal{A}^{H,G,O_{sk}}(pk, \mathsf{OHG.B})$	
5, $t \leftarrow \mathsf{verify}(pk, sk, \mathsf{OHG.A}, m^*, r, s, \mathsf{OHG.C})$	
$\mathcal{C}$ **output** $t \in \{0,1\}$ **as game's output**	

Figure 1: The oracle-hiding game $\mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}}$. Here cha_1 to cha_3 and verify are deterministic algorithms used by challenger $\mathcal{C}$. $\mathsf{ota}^{H,G}$ is an oracle-testing algorithm and ota_2 is an internal deterministic algorithm of $\mathsf{ota}^{H,G}$.

We say that oracle-hiding game $\mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}}$ is in the ROM if $\mathcal{A}$ has only classical access to oracle H, G and O_{sk}, and oracle-hiding game $\mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}}$ is in the QROM if $\mathcal{A}$ can query oracle H, G and O_{sk} in superposition.

In fact, the $\mathsf{IND\text{-}CCA}$ (resp. $\mathsf{IND\text{-}qCCA}$) game of any FO-like transformation in the ROM (resp. QROM) can be rewritten as an oracle-hiding game in the ROM (resp. QROM), as long as we clearly specify the basic elements shown in Fig. 1 (such as the randomness space $\mathcal{R}_1$, $\mathcal{R}_2$ and algorithms cha_1 to cha_3). We emphasize that the oracle-testing algorithm $\mathsf{ota}^{H,G}$ appearing in Fig. 1 is actually an abstraction of the decapsulation algorithm of FO-like transformations, and thus the secret oracle O_{sk} is actually an abstraction of the decapsulation oracle of FO-like transformations.

With the extractable RO-simulator defined in [DFMS22], we then provide a lifting theorem for the oracle-hiding games, extending the ROM reductions to the QROM ones, as presented in Theorem 1.

Theorem 1 (Lifting Theorem of Oracle-hiding Game, informal[3]). *Let ε be the parameter induced by H, G and O_{sk}. Denote by q be the total query times to oracle H, G and O_{sk}. Let $\mathcal{C}$ be a challenger of the oracle-hiding game.*

Given any adversary $\mathcal{A}$ and oracle-hiding game $\mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}}$ in the ROM, there exist adversary $\mathcal{A}_1$ and $\mathcal{A}_2$, invoking $\mathcal{A}$ once in a black-box manner[4] and making no queries to oracle H, G and O_{sk}, such that

$$|\Pr[1 \leftarrow \mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}}] - \Pr[1 \leftarrow \mathsf{OHG}_{\mathcal{A}_1}^{\mathcal{C}}]| \le O(q) \cdot \Pr[1 \leftarrow \mathsf{OHG}_{\mathcal{A}_2}^{\mathcal{C}'}] + O(q) \cdot \varepsilon. \quad (1)$$

Here $\mathcal{C}'$ is identical with $\mathcal{C}$ except that it finally generates $t \in \{0,1\}$ by a new algorithm verify'.

[3]The lifting theorem is formally described in Section 4.1, and is divided into two parts, Lemma 3 and Theorem 4, for clarity.

[4]We stress that the rewinding procedure is not performed.

Then for any quantum adversary $\mathcal{B}$ and oracle-hiding game $\mathsf{OHG}_{\mathcal{B}}^{\mathcal{C}}$ in the QROM, by mimicking the construction of $\mathcal{A}_1$ and $\mathcal{A}_2$, we can directly construct quantum adversary $\mathcal{B}_1$ and $\mathcal{B}_2$, that invokes $\mathcal{B}$ in a black-box manner without any queries to oracle H, G and O_{sk} satisfy

$$|\Pr[1 \leftarrow \mathsf{OHG}_{\mathcal{B}}^{\mathcal{C}}] - \Pr[1 \leftarrow \mathsf{OHG}_{\mathcal{B}_1}^{\mathcal{C}}]| \leq O(q) \cdot \sqrt{\Pr[1 \leftarrow \mathsf{OHG}_{\mathcal{B}_2}^{\mathcal{C}'}]} + O(q) \cdot \sqrt{\varepsilon}. \tag{2}$$

Here, we take FO-like transformation $\mathsf{FO}_m^{\perp}$ for instance to illustrate (in a high level) how Theorem 1 lifts the classical $\mathsf{IND\text{-}CCA}$ reduction of $\mathsf{FO}_m^{\perp}$ in the ROM to the $\mathsf{IND\text{-}qCCA}$ reduction in the QROM.

Let game $\mathsf{Game}_{\mathcal{A},\mathsf{FO}_m^{\perp}}^{\mathsf{IND\text{-}CCA}}$ be the $\mathsf{IND\text{-}CCA}$ game of $\mathsf{FO}_m^{\perp}$ with classical adversary $\mathcal{A}$ in the ROM, then we can rewrite this game as an oracle-hiding game $\mathsf{OHG}_{\tilde{\mathcal{A}}}^{\mathcal{C}_{\mathsf{FO}}}$ by designing appropriate classical adversary $\tilde{\mathcal{A}}$ and challenger $\mathcal{C}_{\mathsf{FO}}$. Hence

$$\Pr\left[1 \leftarrow \mathsf{Game}_{\mathcal{A},\mathsf{FO}_m^{\perp}}^{\mathsf{IND\text{-}CCA}}\right] = \Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{A}}}^{\mathcal{C}_{\mathsf{FO}}}\right]. \tag{3}$$

By Eq. (1), there exists adversary $\tilde{\mathcal{A}}_1$ and $\tilde{\mathcal{A}}_2$ without any oracle queries satisfy

$$\left|\Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{A}}}^{\mathcal{C}_{\mathsf{FO}}}\right] - \Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{A}}_1}^{\mathcal{C}_{\mathsf{FO}}}\right]\right| \leq O(q) \cdot \Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{A}}_2}^{\mathcal{C}'}\right] + O(q) \cdot \varepsilon. \tag{4}$$

Then, we observe that for any adversary $\mathcal{A}$ without any oracle queries, the oracle-hiding game $\mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}}$ and oracle-hiding game $\mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}'}$ must satisfy

$$\Pr\left[1 \leftarrow \mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}_{\mathsf{FO}}}\right] = \frac{1}{2},\ \Pr\left[1 \leftarrow \mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}'}\right] = \mathsf{Adv}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}. \tag{5}$$

Here $\mathsf{Adv}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}$ is the $\mathcal{A}$'s $\mathsf{OW\text{-}CPA}$ advantage against the underlying PKE scheme PKE. Thus combing Eq. (3) to Eq. (5), we actually obtain the $\mathsf{IND\text{-}CCA}$ security reduction of $\mathsf{FO}_m^{\perp}$ in the ROM.

Based on the challenger $\mathcal{C}_{\mathsf{FO}}$, the $\mathsf{IND\text{-}qCCA}$ game $\mathsf{Game}_{\mathcal{B},\mathsf{FO}_m^{\perp}}^{\mathsf{IND\text{-}qCCA}}$ of $\mathsf{FO}_m^{\perp}$ with quantum adversary $\mathcal{B}$ in the QROM can be rewritten as an oracle-hiding game $\mathsf{OHG}_{\tilde{\mathcal{B}}}^{\mathcal{C}_{\mathsf{FO}}}$ by designing appropriate quantum adversary $\tilde{\mathcal{B}}$. Hence

$$\Pr\left[1 \leftarrow \mathsf{Game}_{\mathcal{B},\mathsf{FO}_m^{\perp}}^{\mathsf{IND\text{-}qCCA}}\right] = \Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{B}}}^{\mathcal{C}_{\mathsf{FO}}}\right]. \tag{6}$$

Now we can use Theorem 1 to directly obtain $\tilde{\mathcal{B}}_1$ and $\tilde{\mathcal{B}}_2$ without any oracle queries satisfy

$$\left|\Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{B}}}^{\mathcal{C}_{\mathsf{FO}}}\right] - \Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{B}}_1}^{\mathcal{C}_{\mathsf{FO}}}\right]\right| \leq O(q) \cdot \sqrt{\Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{B}}_2}^{\mathcal{C}'}\right]} + O(q) \cdot \sqrt{\varepsilon}. \tag{7}$$

By using Eq. (5), we get

$$\Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{B}}_1}^{\mathcal{C}_{\mathsf{FO}}}\right] = \frac{1}{2},\ \Pr\left[1 \leftarrow \mathsf{OHG}_{\tilde{\mathcal{B}}_2}^{\mathcal{C}'}\right] = \mathsf{Adv}_{\tilde{\mathcal{B}}_2,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}. \tag{8}$$

Combining Eq. (6) to Eq. (8), we actually obtain the $\mathsf{IND\text{-}qCCA}$ security reduction of $\mathsf{FO}_m^{\perp}$ in the QROM. That is to say, by using Theorem 1, we directly lift the classical $\mathsf{IND\text{-}CCA}$ reduction of $\mathsf{FO}_m^{\perp}$ in the ROM to the $\mathsf{IND\text{-}qCCA}$ reduction in the QROM with a square-root advantage loss.

Additionally, Theorem 1 might be of independent interest due to the abstraction of the oracle-hiding game.

Standard indistinguishability and anonymity of FO-like transformations: With the lifting theorem of oracle-hiding game, we provide the $\mathsf{IND\text{-}qCCA}$ reductions of FO-like transformation $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\perp}$, $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}_m^{\perp}$ in the QROM. The concrete security bounds of these transformations are as shown in Table 1.

Additionally, the lifting theorem also helps to prove the $\mathsf{ANO\text{-}qCCA}$ security of FO-like transformation $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\perp}$, $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}_m^{\perp}$ in the QROM. Furthermore, we also prove the $\mathsf{ANO\text{-}qCCA}$ security of PKE scheme $\mathsf{FO}^{\not\perp}$+DEM, $\mathsf{FO}^{\perp}$+DEM, $\mathsf{FO}_m^{\not\perp}$+$\mathsf{DEM}$ and $\mathsf{FO}_m^{\perp}$+DEM in the QROM, respectively. These results partly answers the first question in the affirmative.

Table 1: The concrete security bounds for several transformations in the QROM. Here q is adversary's total query times to the oracles. ϵ_{O} (resp. ϵ_{S}) is the success probability of an adversary against the OW-CPA (SDS-IND) security of the underlying PKE scheme. ϵ_{W} is the success probability of an adversary against the WANO-CPA security of the underlying PKE scheme. Disj is the statistical disjointness parameter of the underlying PKE scheme.

Transformation	Security	Correctness	Requirement	Security bound($\approx$)
$\mathsf{FO}_m^{\perp}$ [DFMS22]	IND-CCA	δ-correct	weakly γ-spread	$O(q^2)\sqrt[4]{\gamma}+O(q^2)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}$
$\mathsf{FO}_m^{\not\perp}, \mathsf{FO}^{\not\perp}$ [Xag22]*	ANO-CCA	δ-correct	—	$O(q^2)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}+\epsilon_{\mathsf{S}}+\mathsf{Disj}$
$\mathsf{FO}^{\not\perp}$ [GMP22]	ANO-CCA	δ-correct	—	$O(q^2)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}+\epsilon_{\mathsf{W}}+\ldots$**
$\mathsf{FO}^{\perp}, \mathsf{FO}_m^{\perp}$ Our work	IND-qCCA	δ-correct	weakly γ-spread	$O(q)\sqrt{\gamma}+O(q)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}$
$\mathsf{FO}^{\not\perp}, \mathsf{FO}_m^{\not\perp}$ Our work	IND-qCCA	δ-correct	weakly γ-spread	$O(q)\sqrt{\gamma}+O(q)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}$
$\mathsf{FO}^{\perp}, \mathsf{FO}_m^{\perp}$ Our work	ANO-qCCA	δ-correct	weakly γ-spread	$\epsilon_{\mathsf{S}}+O(q)\sqrt{\gamma}+O(q)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}$
$\mathsf{FO}^{\not\perp}, \mathsf{FO}_m^{\not\perp}$ Our work	ANO-qCCA	δ-correct	weakly γ-spread	$\epsilon_{\mathsf{S}}+O(q)\sqrt{\gamma}+O(q)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}$
$\mathsf{FO}^{\perp}+\mathsf{DEM}$ $\mathsf{FO}_m^{\perp}+\mathsf{DEM}$ Our work	ANO-qCCA	δ-correct	weakly γ-spread	$\epsilon_{\mathsf{S}}+O(q)\sqrt{\gamma}+O(q)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}$
$\mathsf{FO}^{\not\perp}+\mathsf{DEM}$ $\mathsf{FO}_m^{\not\perp}+\mathsf{DEM}$ Our work	ANO-qCCA	δ-correct	weakly γ-spread	$\epsilon_{\mathsf{S}}+O(q)\sqrt{\gamma}+O(q)\sqrt{\delta}+O(q)\sqrt{\epsilon_{\mathsf{O}}}$

* The ANO-CCA security of $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}^{\not\perp}$ has not been directly proven in [Xag22]. However, we can obtain the bound we presented here by combining Theorem 4.1 and Theorem D.1 of [Xag22].

** The ANO-CCA security reduction of $\mathsf{FO}^{\not\perp}$ in [GMP22] also needs the SCFR-CPA security of PKE scheme PKE_1.

As shown in Table 1, our IND-qCCA security bound of $\mathsf{FO}_m^{\perp}$ is tighter than the IND-CCA security bound of $\mathsf{FO}_m^{\perp}$ in [DFMS22][5].

In terms of the anonymity, our work has two requirements for the underlying PKE. One of the requirements is that the underlying PKE scheme should be OW-CPA-secure and SDS-IND-secure, which is also required in [Xag22]. The other requirement is that the PKE scheme should be weakly γ-spread, which has been analyzed in [HHM22] for several KEM submissions to the NIST PQC competition.

For FO-like transformation $\mathsf{FO}^{\not\perp}$, our ANO-qCCA security bound is more concise than that in [GMP22], and has no additional security requirements for the underlying PKE except the SDS-IND security. Moreover, Our ANO-qCCA security bound of $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}^{\not\perp}$ is nearly identical to the ANO-CCA security bound presented in [Xag22], with the only difference being the substitution of the term Disj with $O(q)\sqrt{\gamma}$.

Perhaps surprisingly, it can be further noticed that our ANO-qCCA security bounds of PKE scheme $\mathsf{FO}^{\not\perp}$+DEM, $\mathsf{FO}^{\perp}$+DEM, $\mathsf{FO}_m^{\not\perp}$+DEM and $\mathsf{FO}_m^{\perp}$+DEM are irrelevant to the security of the underlying DEM scheme. Specifically, the only security requirement of the ANO-qCCA security for those hybrid PKE schemes is that the underlying PKE scheme, is SDS-IND-secure and OW-CPA-secure. This finding may simplify the anonymity analysis of hybrid PKE scheme built via KEM-DEM paradigm with underlying KEM obtained from the NIST KEM submissions.

A new variant of O2H: Czajkowski et al. [CMSZ19] proposed the One-way to Hiding (O2H) Lemma for compressed oracles, that is a combination of the semi-classical O2H Theorem [AHU19] and the compressed oracle technique [Zha19]. We generalize this lemma to the compressed semi-classical O2H theorem, as shown in Theorem 2, by allowing quantum oracle algorithm $\mathcal{A}$ to make both compressed oracle queries and database read queries.

In our paper, the compressed semi-classical O2H theorem is only applied to prove the lifting theorem Theorem 1, but we emphasize that this theorem also might be of independent interest.

Theorem 2 (Compressed Semi-classical O2H, informal)**.** *Let H be the compressed oracle, S be a subset of the database and z be a random string. Let $H\backslash S$ be an oracle that first queries H and then*

[5] An IND-qCCA/ANO-qCCA secure scheme is also IND-CCA/ANO-CCA secure, due to the security definitions.

queries $\mathcal{O}_S^{CSC}$. *Let* $\mathcal{A}$ *be a quantum oracle algorithm that has quantum access to both* H *and database read oracle* oRead. *Suppose* $\mathcal{A}$ *queries* H *(resp.* oRead*) at most* q_1 *(resp.* q_2*) times. Define*

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

Here Find *is the event that* $\mathcal{O}_S^{CSC}$ *ever returns 1, then*

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

1.3 Techniques Overview

Our security reduction rely on Theorem 1, the lifting theorem for oracle-hiding games. We prove the IND-qCCA security of FO-like transformations by rewriting their IND-qCCA game in the QROM as the oracle-hiding game, computing ε for the oracle-hiding game, and apply Eq. (2) of Theorem 1 to derive their IND-qCCA security bounds.

However, in the ANO-qCCA game, the challenger needs to generate two public/secret key pair, (pk_0, sk_0) and (pk_1, sk_1), the challenge query are encrypted by pk_0 and pk_1, respectively, and the adversary has quantum access to two decryption oracles: one decrypting with sk_0 and the other with sk_1. This makes it difficult to rewrite the ANO-qCCA game FO-like transformations as the oracle-hiding game. Therefore, on the ANO-qCCA security, a more subtle argument is needed.

We resolve this obstacle in terms of the pseudorandomness of PKE/KEM defined in [Xag22]. Taking PKE for instance, this property states that a ciphertext is indistinguishable from a random string chosen by a simulator that takes the security parameter as input.

A strong pseudorandomness was proposed in [Xag22], and it was proved that the strong pseudorandomness implies the anonymity. Nevertheless, the strong pseudorandomness seems to be slightly stronger than our requirement, and a weaker property, named weak pseudorandomness, is defined in this paper and is proved to imply the anonymity. In the security game of weak pseudorandomness (WPR-qCCA game defined in Appendix G), only one public/secret key pair is used, we can then rewrite the game as the oracle-hiding game, and apply Theorem 1 to prove the weak pseudorandomness, and, consequently, the anonymity.

In this way, the ANO-qCCA security of $\mathsf{FO}^{\not\perp}$+DEM, $\mathsf{FO}^{\perp}$+DEM, $\mathsf{FO}_m^{\not\perp}$+DEM and $\mathsf{FO}_m^{\perp}$+DEM can be irrelevant to the security of the underlying DEM scheme and Disj used in [Xag22].

Proof sketch of Theorem 1: Note that Theorem 1 actually consists of two results: Eq. (1) for any oracle-hiding game in the ROM; Eq. (2) for any oracle-hiding game in the QROM.

- In the Section 4.2.1 of our paper, Eq. (1) is proved through a game sequence $\mathbf{G_0^c}$ to $\mathbf{G_4^c}$, where

$$\Pr\left[1 \leftarrow \mathbf{G_0^c}\right] = \Pr\left[1 \leftarrow \mathsf{OHG}_{\mathcal{A}}^{\mathcal{C}}\right],\ \Pr\left[1 \leftarrow \mathbf{G_4^c}\right] = \Pr\left[1 \leftarrow \mathsf{OHG}_{\mathcal{A}_1}^{\mathcal{C}}\right],$$

$$\sum_{\mathbf{i}=0}^{3}\left|\Pr\left[1 \leftarrow \mathbf{G_i^c}\right] - \Pr\left[1 \leftarrow \mathbf{G_{i+1}^c}\right]\right| \leq O(q)\cdot \Pr\left[1 \leftarrow \mathsf{OHG}_{\mathcal{A}_2}^{\mathcal{C}'}\right] + O(q)\cdot\varepsilon.$$

- In the Section 4.2.2 of our paper, Eq. (2) is proved through a game sequence $\mathbf{G_0^q}$ to $\mathbf{G_6^q}$, where

$$\Pr\left[1 \leftarrow \mathbf{G_0^q}\right] = \Pr\left[1 \leftarrow \mathsf{OHG}_{\mathcal{B}}^{\mathcal{C}}\right],\ \Pr\left[1 \leftarrow \mathbf{G_6^q}\right] = \Pr\left[1 \leftarrow \mathsf{OHG}_{\mathcal{B}_1}^{\mathcal{C}}\right],$$

$$\sum_{\mathbf{i}=0}^{5}\left|\Pr\left[1 \leftarrow \mathbf{G_i^q}\right] - \Pr\left[1 \leftarrow \mathbf{G_{i+1}^q}\right]\right| \leq O(q)\cdot \sqrt{\Pr\left[1 \leftarrow \mathsf{OHG}_{\mathcal{B}_2}^{\mathcal{C}'}\right]} + O(q)\cdot\sqrt{\varepsilon}.$$

Roughly speaking, the purpose of both game sequences $\mathbf{G_0^c}$ to $\mathbf{G_4^c}$ and $\mathbf{G_0^q}$ to $\mathbf{G_6^q}$ is to design an adversary, i.e., $\mathcal{A}_1$ in $\mathbf{G_4^c}$ and $\mathcal{B}_1$ in $\mathbf{G_6^q}$, that invokes the adversary of the first game and does not query any oracle.

To achieve it, the main problem is to simulate classical and quantum accessed random oracle H, G, as well as the secret oracle O_{sk}. Here, we provide a high-level explanation of how we simulate these oracles.

- For random oracle H and G, we simulate it *on-the-fly* by list $\mathfrak{L}_H$ and $\mathfrak{L}_G$, respectively. Now there exists a query transcript $\mathfrak{L}_H$ of random oracle H.
- For the quantum random oracle H, we simulate it by using the RO-interface $\mathsf{eCO.RO}$ of the extractable RO-simulator $\mathcal{S}(:= \{\mathsf{eCO.RO}, \mathsf{eCO.E}\})$. As for G, we simulate it with a $2q$-wise independent hash function.
- For the classical accessed secret oracle O_{sk}, we simulate it by a classical *plaintext-extractor* without using the secret key sk. For the secret oracle query, O_{sk} replies it by reading and extracting from the query transcript $\mathfrak{L}_H$.
- For the quantum accessed secret oracle O_{sk}, we simulate it by a quantum *plaintext-extractor* without using the secret key sk. The extractor is constructed with the extraction-interface $\mathsf{eCO.E}$ of the extractable RO-simulator.

Indeed, in our detailed proof of the lifting theorem Theorem 1, it can be observed that the quantum *plaintext-extractor*, constructed by using the extraction-interface $\mathsf{eCO.E}$, can be regarded as the quantum counterpart of the classical *plaintext-extractor*. Moreover, it can be noticed that an one-to-one correspondence exists between the operations of $\mathcal{A}_1$ and $\mathcal{B}_1$, and those of $\mathcal{A}_2$ and $\mathcal{B}_2$. This correspondence enables us to construct $\mathcal{B}_1$ and $\mathcal{B}_2$ directly by mimicking the construction of $\mathcal{A}_1$ and $\mathcal{A}_2$.

1.4 Related Works

[XY19] and [LW21] have argued the $\mathsf{IND\text{-}qCCA}$ security of FO-like transformations. However, their work mainly focused on FO-like transformations with implicit rejection type. As for explicit rejection type, only transformation HU, an adapted version of $\mathsf{QU}_m^{\not\perp}$, has been analysed in [XY19].

To the post-quantum security of FO-like transformations with explicit rejection type, there have been only [DFMS22] and [HHM22] providing the $\mathsf{IND\text{-}CCA}$ security reduction of $\mathsf{FO}_m^{\perp}$, as far as we know. Moreover, Hövelmanns et al. also showed that the $\mathsf{IND\text{-}CCA}$ security of $\mathsf{FO}_m^{\perp}$ implies the $\mathsf{IND\text{-}CCA}$ security of all remaining FO-like transformations [HHM22].

It should be noted that the $\mathsf{IND\text{-}CCA}$ security reductions of $\mathsf{FO}_m^{\perp}$ given in [DFMS22] and [HHM22] seem not to hold for the $\mathsf{IND\text{-}qCCA}$ security, where the adversary is allowed to query the decapsulation oracle in superposition. There are two reasons as follows.

1. Both [DFMS22] and [HHM22] use property 4.a and 4.b of Theorem 4.3 in [DFMS22] to prove the $\mathsf{IND\text{-}CCA}$ security, but these properties only hold for classical queries.
2. In the $\mathsf{IND\text{-}CCA}$ security reductions of [DFMS22] and [HHM22], a list is maintained to record the adversary's classical decapsulation queries. However, if the decapsulation oracle is quantum-accessible, this record procedure becomes infeasible due to the quantum no-cloning principle.

The post-quantum anonymity of FO-like transformation was first studied by Grubbs et al. [GMP22]. Theorem 7 of [GMP22] implies that the $\mathsf{ANO\text{-}CCA}$ security of PKE scheme $\mathsf{FO}^{\not\perp}$+DEM is guaranteed by the $\mathsf{ANO\text{-}CCA}$ security of KEM obtained by $\mathsf{FO}^{\not\perp}$, the $\mathsf{INT\text{-}CTXT}$ security of DEM, and other security requirements. Xagawa [Xag22] then proved that the $\mathsf{ANO\text{-}CCA}$ security of the hybrid PKE scheme $\mathsf{FO}^{\not\perp}$+DEM in the QROM can be implied by the $\mathsf{SPR\text{-}OTCCA}$ security of DEM, the $\mathsf{SPR\text{-}CCA}$ and $\mathsf{SSMT\text{-}CCA}$ security of KEM scheme obtained by $\mathsf{FO}^{\not\perp}$.

However, both in [GMP22] and [Xag22], the $\mathsf{ANO\text{-}CCA}$ security of hybrid PKE scheme $\mathsf{FO}^{\not\perp}$+DEM depends on the security requirement of the underlying DEM.

As the last point, there have been several works on the lifting theorem from ROM proofs to QROM proofs [BDF+11, CMS19, KS20, CFHL21, YZ21].

2 Preliminaries

2.1 Notations

The security parameter is denoted by λ. We denote by $\mathsf{boole}[A]$ a bit that is 1 if the predicate A keeps true and otherwise 0. For a finite set S, we denote the sampling of a uniformly random element x as $x \xleftarrow{\$} S$, and the cardinality of S as $|S|$. $x \leftarrow \mathcal{D}$ represents that the x is chosen subject to distribution $\mathcal{D}$. $\Pr[A : B]$ is the probability that the predicate A keeps true where all variables in A are conditioned according to predicate B. Let $y \leftarrow \mathcal{A}(x)$ represent the output of algorithm $\mathcal{A}$ on input x, $y \leftarrow \mathbf{G}$ represent that the game $\mathbf{G}$ finally outputs y. Denote by $\mathcal{F}_{m,n}$ the set of all functions with domain $\{0,1\}^m$ and codomain $\{0,1\}^n$. For a function or an algorithm f, denote by $\mathrm{Time}[f]$ the worst case of the running time of $f(x)$ for all input x.

2.2 Quantum Computation

We refer to [NC16] for detailed basics of quantum computation and quantum information, and we only introduce some important quantum notions used in this paper in Appendix A.

2.3 The Quantum Random Oracle Model

The random oracle model (ROM) is an ideal model in which a uniformly random function $H : \{0,1\}^m \to \{0,1\}^n$ is selected and all parties have access to a random oracle O_H, where O_H output $H(x)$ on input x. We can simulate the random oracle O_H efficiently for the classical query by *on-the-fly* technique. When a random oracle scheme is implemented, we select a concrete hash function as an instantiation of the random oracle. In the quantum setting, a quantum adversary can evaluate a hash function in superposition. To capture this issue, the quantum random oracle model (QROM) is considered and the adversary has access to the quantum random oracle O_H in this model [BDF+11]. The quantum random oracle O_H can be viewed as a unitary operation that maps $|x, y\rangle$ to $|x, y \oplus H(x)\rangle$, where $x \in \{0,1\}^m$ and $y \in \{0,1\}^n$. We will introduce several useful lemmas regarding the QROM in Appendix B.

2.4 The Compressed Standard Oracle

The compressed oracle technique is introduced by Zhandry in [Zha19], by using this technique, one can perfectly simulate the quantum accessible random oracle and record some information about the adversary's quantum query. In this subsection, we only introduce the database model and a specific version of the compressed oracle named compressed standard oracle. Moreover, we fix the query bound to the compressed standard oracle to be constant q since all results are about the adversary with fixed query times.

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

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

where $i \leq q$, $x_1, x_2, \ldots, x_i \neq \perp$ and $x_1 < x_2 < \cdots < x_i$, all $(\perp, 0^n)$ pairs are at the end of the collection. Let $\mathbf{D}_q$ be the set of all these databases. For a $x \in \{0,1\}^m$, we will write $D(x) = y$ if y exists such that $(x, y) \in D$, and $D(x) = \perp$ otherwise. Let $n(D)$ be the number of pairs $(x, y) \in D$ that $x \neq \perp$. For a pair $(x, y) \in \{0,1\}^m \times \{0,1\}^n$ and a database $D \in \mathbf{D}_q$ with $n(D) < q$ and $D(x) = \perp$, write $D \cup (x, y)$ to be the new database obtained by first deleting a $(\perp, 0^n)$ pair, then inserting (x, y) appropriately into D and maintain the ordering of the x values.

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

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

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

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

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

It is obvious that $\mathsf{StdDecomp}_x$ is a unitary operation and $\mathsf{StdDecomp}_x \circ \mathsf{StdDecomp}_x = \mathbf{I}$ for any $x \in \{0,1\}^m$, where $\mathbf{I}$ is the identity operator.

Definition 2 (Compressed Standard Oracle)**.** *Let* X *(resp.* Y*) be the quantum register defined over* $\{0,1\}^m$ *(resp.* $\{0,1\}^n$*). Let the initial state on database register* D_q *be* $|D^\perp\rangle$*, where* $D^\perp \in \mathbf{D}_q$ *is the database only contains* q *pairs* $(\perp, 0^n)$*. A query to the compressed standard oracle with input/output register* X/Y *is implemented by acting the following unitary operation* CStO *on registers* XYD_q*.*

$$\mathsf{CStO} := \sum_{x \in \{0,1\}^m} |x\rangle\langle x|_{\mathsf{X}} \otimes \mathsf{StdDecomp}_x \circ \mathsf{CNOT}^x_{\mathsf{YD}_q} \circ \mathsf{StdDecomp}_x. \tag{9}$$

Here $\mathsf{CNOT}^x_{\mathsf{YD}_q}$ *maps* $|y, D\rangle$ *(*$y \in \{0,1\}^n$*,* $D \in \mathbf{D}_q$*) to* $|y \oplus D(x), D\rangle$ *if* $D(x) \neq \perp$*, to* $|y, D\rangle$ *if* $D(x) = \perp$[6]*.*

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

Lemma 1 ([Zha19])**.** *For any adversary makes at most* q *times quantum queries, compressed standard oracle defined in Definition 2 and quantum random oracle* $H : \{0,1\}^m \to \{0,1\}^n$ *are perfectly indistinguishable.*

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

$$\mathsf{Read}_f|x, D, y\rangle = |x, D, y + f(x, D)\rangle. \tag{10}$$

Here $+ : \mathcal{Y} \times \mathcal{Y} \to \mathcal{Y}$ is some group operation on $\mathcal{Y}$. Note that Read_f does not change the database in the computational basis state, it only compute $f(x, D)$ and return it in register Y, therefore we call Read_f a database read operation.

For an adversary $\mathcal{A}$ with access to the compressed standard oracle, we say $\mathcal{A}$ can make database read queries if it can query oracle oRead_f with input/output register X/Y for a fixed function f, where oracle oRead_f is implemented by acting the database read operation Read_f defined in Eq. (10) on registers XYD_q.

2.5 The Extractable RO-Simulator

In [DFMS22], Don et al. generalized the compressed standard oracle and defined the extractable RO-simulator. Roughly speaking, this simulator simulates the quantum random oracle H by using the compressed standard oracle, and has an extraction-interface that can output a x satisfy $f(x, H(x)) = t$ for an input t. In the following, we introduce the extractable RO-simulator and prove a lemma that will be used in the next section. We stress that, identical with Section 2.4, the database register used here is also D_q. Therefore, different with the inefficient version defined in [DFMS22], the extractable RO-simulator described here is an efficient version and it at most simulates q times queries to the quantum random oracle H.

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

$$R^f_t := \{(x, y) \in \{0,1\}^m \times \{0,1\}^n | f(x, y) = t\}, \quad \Gamma_{R^f_t} := \max_{x \in \{0,1\}^m} |\{y \in \{0,1\}^n | f(x, y) = t\}|. \tag{11}$$

[6]The $\mathsf{CNOT}^x_{\mathsf{YD}_q}$ acts trivially on the state $|y, D\rangle$ that satisfies $D(x) = \perp$ is additionally defined in [DFMS22], which is also equivalent to the additional notation that "$y \oplus \perp = y$" defined in [Zha19].

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

$$\Sigma^x := \sum_{\substack{D\ s.t.\ (x,D(x))\in R_t^f \\ x'<x,(x',D(x'))\notin R_t^f}} |D\rangle\langle D| \ (x \in \{0,1\}^m), \quad \Sigma^{\perp} := \mathbf{I} - \sum_{x\in\{0,1\}^m} \Sigma^x. \tag{12}$$

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

Indeed, the measurement $\mathbb{M}^{R_t^f}$ will return the smallest x such that $(x, D(x)) \in R_t^f$. If such x does not exist, $\mathbb{M}^{R_t^f}$ will return $\perp$. Similar with [DFMS22], we also consider the purified measurement $\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}$ corresponding to $\mathbb{M}^{R_t^f}$ given by a unitary operation acts on registers $\mathsf{D}_q\mathsf{P}$ as

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

Here P is a quantum register defined over $\{0,1\}^{m+1}$[7], $D \in \mathbf{D}_q$ and $p \in \{0,1\}^{m+1}$.

Definition 3 (The (efficient version of the) extractable RO-simulator)**.** *The extractable RO-simulator $\mathcal{S}(f)$ with internal database register D_q is a black-box oracle with two interfaces, the RO-interface* eCO.RO *and the extraction-interface* $\mathsf{eCO.E}_f$. *$\mathcal{S}(f)$ prepares its database register D_q to be in state $|D^{\perp}\rangle$ at everything begins, where $D^{\perp} \in \mathbf{D}_q$ is the database only contains q pairs $(\perp, 0^n)$. Then, the RO-interface* eCO.RO *and the extraction-interface* $\mathsf{eCO.E}_f$ *act as*

- *Let X (resp. Y) be the quantum register defined over $\{0,1\}^m$ (resp. $\{0,1\}^n$), let T be the quantum register defined over $\mathcal{Y}$.*
- eCO.RO*: Upon a quantum RO-query, with query registers* XY*, $\mathcal{S}(f)$ applies* CStO *defined in Definition 2 to registers XYD_q.*
- $\mathsf{eCO.E}_f$*: Upon a quantum extraction-query, with query registers* TP*, $\mathcal{S}(f)$ applies*

$$\mathsf{Ext}_f := \sum_{t\in\mathcal{Y}} |t\rangle\langle t|_{\mathsf{T}} \otimes \mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}} \tag{13}$$

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

Moreover, by the Theorem 4.3 of [DFMS22], the total runtime of $\mathcal{S}(f)$ is bounded as[8]

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

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

The eCO.RO (resp. $\mathsf{eCO.E}_f$) can also be classically queried, in this case, the query registers XY (resp. TP) are measured after applying the unitary operation CStO (resp. Ext_f). The eCO.RO can also be queried in parallel, and k-parallel queries to eCO.RO can be processed by sequentially implementing CStO k times [CFHL21].

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

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

Here function $g : \mathcal{Y} \times \mathbf{D}_q \to \{0,1\}^{m+1}$ on input (t, D) output the smallest value x that satisfies $(x, D(x)) \in R_t^f$, if such x does not exist, function g output $\perp$. Therefore, by the definition of database read operation given in Section 2.4, Ext_f can also be viewed as a database read operation.

Next we introduce a lemma about the extractable RO-simulator $\mathcal{S}(f)$, the detailed proof is shown in Appendix C.

Lemma 2. *Let $\mathsf{StdDecomp}_x$ be the unitary operation introduced in Section 2.4, let $\Gamma_{R_t^f}$, $\Sigma^{\perp}$ and Ext_f be as in Eq. (11), (12) and (13), respectively. Then*

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

[8] Although [DFMS22] defined an inefficient version of the extractable RO-simulator, the total runtime of the efficient version is given instead in the Theorem 4.3 of [DFMS22].

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

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

Remark 1. *Note that the definition of Ext_f and $\Sigma^{\perp}$ are based on the efficient representation of the compressed oracle (i.e. the compressed standard oracle). But we stress that Lemma 2 can still be easily proved by using the Lemma 3.3 and Lemma 3.4 of [DFMS22], even these two lemmas are stated by using the inefficient representation of the compressed oracle. The reason is that the two representations are isometrically equivalent as discussed in the Sect. B of [DFMS22]. However, for convenience and completeness, we directly prove Lemma 2 in Appendix C by using the representation of the compressed standard oracle.*

2.6 Compressed Semi-Classical One Way to Hidding

In this section, we generalize the O2H variant Theorem 10 in [CMSZ19] by allowing that the algorithm $\mathcal{A}$ with access to the compresses standard oracle can also make database read queries. This new theorem may can be applied to more scenes in the QROM.

Compressed semi-classical oracle. Let $\mathbf{D}_q$ be the database set defined in Section 2.4, let S be a subset of $\mathbf{D}_q$. Define function f_S such that $f_S(D) = 1$ if $D \in S$ and $f_S(D) = 0$ otherwise. The compressed semi-classical oracle $\mathcal{O}_S^{CSC}$ performs the following operation on input state $\sum_{z\in\{0,1\}^*, D\in\mathbf{D}_q} \alpha_{z,D}|z, D\rangle$:

1. Initialize a single qubit L with $|0\rangle_L$, transform state $\sum_{z\in\{0,1\}^*, D\in\mathbf{D}_q} \alpha_{z,D}|z, D\rangle|0\rangle_L$ into state $\sum_{z\in\{0,1\}^*, D\in\mathbf{D}_q} \alpha_{z,D}|z, D\rangle|f_S(D)\rangle_L$.
2. Measure L and output the measurement outcome.

Denote Find as the event that $\mathcal{O}_S^{CSC}$ ever returns 1. Compared with the semi-classical oracle $\mathcal{O}_S^{SC}$, compressed semi-classical oracle $\mathcal{O}_S^{CSC}$ performs the projective measurement on the database register.

Remark 2. *The definition of $\mathcal{O}_S^{CSC}$ is based on the definition of Algorithm 4 (Measurement of a relation R) in [CMSZ19]. For computational basis state $|z, D\rangle$, the Algorithm 4 needs to compute the number of non-padding pairs (i.e. $n(D)$ in our paper) of the database D in a register and finally uncompute it, since in [CMSZ19], it is only reasonable to check if the non-padding pairs are in the relation R. We stress that $\mathcal{O}_S^{CSC}$ does not need to compute $n(D)$, because we do not care about the internal pairs of D and only care about if D belong to the subset S.*

Theorem 3 (Compressed semi-classical O2H with database read queries)**.** *Let $H : \{0,1\}^m \to \{0,1\}^n$ be a quantum random oracle that is implemented by the compressed standard oracle. Let f be a function with domain $\mathcal{X} \times \mathbf{D}_q$ and codomain $\mathcal{Y}$, D_q be the database register defined over $\mathbf{D}_q$. Let S be a subset of $\mathbf{D}_q$ that $D^{\perp} \notin S$ and z be a random string, where $D^{\perp}$ is the database only contain q pairs $(\perp, 0^n)$, S and z may have arbitrary joint distribution $\mathcal{D}$. Let $H\backslash S$ be an oracle that first queries H and then queries $\mathcal{O}_S^{CSC}$.*

Let $\mathcal{A}$ be a quantum oracle algorithm (not necessarily unitary) that is given access to H and oRead_f, and we suppose $\mathcal{A}$ queries H (resp. oRead_f) at most $q_1 \leq q$[9] (resp. q_2) times. Here oracle oRead_f is implemented by the database read operation Read_f defined in Eq. (10). Define

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

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

Let $\mathrm{J}_S := \sum_{D\in S} |D\rangle\langle D|$ be the projector acts on the database register D_q, let CStO be as in Eq. (9), we then have
$$P_{\text{find}} \leq q_1 \cdot \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}} \|[\mathrm{J}_S, \mathsf{CStO}]\|^2.$$

[9]This limitation on q_1 is because that the database register D_q can only be used to perfectly simulate q times quantum random oracle queries at most.

The detailed proof of Theorem 3 is similar to the proof of the semi-classical O2H theorem [AHU19] and we present it in Appendix D.

3 The Oracle-Hiding Game

In this section, we define a type of games called oracle-hiding games, which involves a classical challenger and an efficient adversary. The definitions introduced as follows are the foundation of the lifting theorem, provided in the next section.

Definition 4 (Oracle-Testing Algorithm)**.** *Let key generator* KGen *be a polynomial time algorithm, which on input* 1^λ*, outputs a public/secret key pair* $(\mathsf{pk}, \mathsf{sk})$*. Let* $O_0 \xleftarrow{\$} \mathcal{F}_{m(\lambda),n(\lambda)}$ *and* $O_1 \xleftarrow{\$} \mathcal{F}_{m'(\lambda),n'(\lambda)}$ *be random oracles, where* $m(\lambda)$*,* $n(\lambda)$*,* $m'(\lambda)$ *and* $n'(\lambda)$ *are functions of* λ*. The oracle-testing algorithm* $\mathsf{ota}^{O_0,O_1}(1^\lambda, \mathsf{sk}, \cdot)$ *is an algorithm that has access to random oracle* O_0 *and* O_1*, takes as input a* $\alpha \in \mathcal{X}$ *and is executed as follows.*

1. *Compute* $\beta := \mathsf{ota}_1(1^\lambda, \mathsf{sk}, \alpha) \in \{0,1\}^{m'(\lambda)} \cup \perp$*. If* $\beta = \perp$*, return* $\mathsf{f}_{\mathsf{ota}}(\alpha) \in \{0,1\}^{l(\lambda)}$*.*
2. *Else, compute* $\mathsf{ota}_2(1^\lambda, \mathsf{pk}, \beta, O_1(\beta)) \in \mathcal{X}$*. If* $\mathsf{ota}_2(1^\lambda, \mathsf{pk}, \beta, O_1(\beta)) \neq \alpha$*, return* $\mathsf{f}_{\mathsf{ota}}(\alpha) \in \{0,1\}^{l(\lambda)}$*.*
   (a) *Else, compute* $\gamma := \mathsf{ota}_3(1^\lambda, \mathsf{pk}, \alpha, \beta) \in \{0,1\}^{m(\lambda)}$*, return* $\mathsf{ota}_4(1^\lambda, \mathsf{pk}, \alpha, \beta, O_0(\gamma)) \in \{0,1\}^{l(\lambda)}$*.*

Here $\mathsf{ota}_1(1^\lambda, \mathsf{sk}, \cdot)$*,* $\mathsf{ota}_2(1^\lambda, \mathsf{pk}, \cdot)$*,* $\mathsf{ota}_3(1^\lambda, \mathsf{pk}, \cdot)$ *and* $\mathsf{ota}_4(1^\lambda, \mathsf{pk}, \cdot)$ *are deterministic polynomial time algorithms,* $\mathsf{f}_{\mathsf{ota}}$ *is a fixed function,* $l(\lambda)$ *is a function of* λ*.*

Define a subset of $\{0,1\}^{n'(\lambda)}$ *to be*

$$\mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta} := \{r \in \{0,1\}^{n'(\lambda)} : \mathsf{ota}_2(1^\lambda, \mathsf{pk}, \beta, r) = \alpha\}. \tag{14}$$

Define parameter $\mathsf{ota.time}$*,* $\mathsf{ota.max}$ *and* $\mathsf{ota.union}$ *to be:*

$$\begin{aligned}
\mathsf{ota.time} &:= \mathrm{Time}[\mathsf{ota}_2] + \mathrm{Time}[\mathsf{ota}_3] + \mathrm{Time}[\mathsf{ota}_4],\\
\mathsf{ota.max} &:= \frac{1}{2^{n'(\lambda)}} \underset{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KGen}(1^\lambda)}{\mathbb{E}} \max_{\alpha\in\mathcal{X},\beta\in\{0,1\}^{m'(\lambda)}} \left|\mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}\right|,\\
\mathsf{ota.union} &:= \frac{1}{2^{n'(\lambda)}} \underset{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KGen}(1^\lambda)}{\mathbb{E}} \max_{\beta\in\{0,1\}^{m'(\lambda)}} \left|\bigcup_{\alpha\in\mathsf{Set}.\beta} \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}\right|,
\end{aligned} \tag{15}$$

where $\mathsf{Set}.\beta := \{\alpha \in \mathcal{X} : \mathsf{ota}_1(1^\lambda, \mathsf{sk}, \alpha) \neq \beta\}$*.*

Definition 5 (Oracle-Hiding Game in the ROM/QROM)**.** *For a classical challenger* $\mathcal{C}(1^\lambda)$ *and an efficient adversary* $\mathcal{A}(1^\lambda)$*, we call game* $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$*, as shown in Fig. 2, an oracle-hiding game if the following conditions are satisfied:*

- $\mathcal{A}(1^\lambda)$ *has access to random oracle* O_0*, random oracle* O_1 *and secret oracle* O_{ota}*, where* O_{ota} *uses the oracle-testing algorithm* $\mathsf{ota}^{O_0,O_1}(1^\lambda, \mathsf{sk}, \cdot)$ *to reply its queries.*
- $\mathcal{C}(1^\lambda)$ *uses random coins* m^**,* r *and* s*, where* s *is sampled from* $\{0,1\}$ *subject to some distribution.*
- $\mathcal{C}(1^\lambda)$ *does not query* O_{ota} *and queries* O_0 *(resp.* O_1*) only by* $m^*||m_0$ *(resp.* $m^*||m_1$*).*
- $\mathsf{cha}_1(1^\lambda, \mathsf{pk}, \cdot)$*,* $\mathsf{cha}_2(1^\lambda, \mathsf{pk}, \cdot)$*,* $\mathsf{cha}_3(1^\lambda, \mathsf{pk}, \cdot)$ *and* $\mathsf{verify}(1^\lambda, \mathsf{pk}, \mathsf{sk}, \cdot)$ *used by* $\mathcal{C}(1^\lambda)$ *are deterministic algorithms.*
- *It can be checked efficiently whether* $\alpha = \mathsf{ota}_2(1^\lambda, \mathsf{pk}, m^*||m_1, O_1(m^*||m_1))$*, by using* $\mathsf{OHG.B}$ *and* pk*. This check takes very little running time and can be ignored.*

We say that game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$ *is in the ROM if* $\mathcal{A}(1^\lambda)$ *has only classical access to* O_0*,* O_1 *and* O_{ota}*. If* $\mathcal{A}(1^\lambda)$ *has quantum oracle access to* O_0*,* O_1 *and* O_{ota}*, game* $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$ *is in the QROM. Then define*

$$\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A},\mathcal{C}}(1^\lambda) := \Pr\left[1 \leftarrow \mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}\right].$$

Game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$

1, $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$

2, $\mathsf{OHG.A} \leftarrow \mathcal{A}^{O_0,O_1,O_{\mathsf{ota}}}(1^\lambda,\mathsf{pk})$

3, $\mathcal{C}(1^\lambda)$ **perform following operation**

$m^* \xleftarrow{\$} \mathcal{R}_1,\ r \xleftarrow{\$} \mathcal{R}_2,\ s \in \{0,1\}$

$m_s \leftarrow \mathsf{cha}_1(1^\lambda,\mathsf{pk},\mathsf{OHG.A},m^*,r)$

$y_s = O_s(m^*||m_s)$

$m_{1-s} \leftarrow \mathsf{cha}_2(1^\lambda,\mathsf{pk},\mathsf{OHG.A},y_s,m^*,r)$

$y_{1-s} = O_{1-s}(m^*||m_{1-s})$

$\mathsf{OHG.B} \leftarrow \mathsf{cha}_3(1^\lambda,\mathsf{pk},\mathsf{OHG.A},y_s,y_{1-s},m^*,r)$

4, $\mathsf{OHG.C} \leftarrow \mathcal{A}^{O_0,O_1,O_{\mathsf{ota}}}(1^\lambda,\mathsf{pk},\mathsf{OHG.B})$

5, $t \leftarrow \mathsf{verify}(1^\lambda,\mathsf{pk},\mathsf{sk},\mathsf{OHG.A},m^*,r,s,\mathsf{OHG.C})$

$\mathcal{C}(1^\lambda)$ **output** $t \in \{0,1\}$ **as game's output**

$O_0(x)$

1, $O \xleftarrow{\$} \mathcal{F}_{m(\lambda),n(\lambda)}$, **return** $O(x)$

$O_1(x)$

1, $O' \xleftarrow{\$} \mathcal{F}_{m'(\lambda),n'(\lambda)}$, **return** $O'(x)$

$O_{\mathsf{ota}}(\alpha)$

1, **If** $\mathsf{OHG.B}$ **is defined and**

$\alpha = \mathsf{ota}_2(1^\lambda,\mathsf{pk},m^*||m_1,O_1(m^*||m_1))$

return $\perp$

Else return $\mathsf{ota}^{O_0,O_1}(1^\lambda,\mathsf{sk},\alpha)$

Figure 2: The detailed process of game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$. We default that the length of m^* is less than or equal to $m(\lambda)$ and $m'(\lambda)$ for any parameter λ.

In oracle-hiding game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$, by using $O_0(m^*||m_0)$ and $O_1(m^*||m_1)$, the challenger computes the adversary's input $\mathsf{OHG.B}$. The secret oracle O_{ota} is implemented by using the oracle-testing algorithm, and it outputs $\perp$ for $\alpha = \mathsf{ota}_2(1^\lambda,\mathsf{pk},m^*||m_1,O_1(m^*||m_1))$ after $\mathsf{OHG.B}$ is defined.

Therefore, even though the adversary has access to secret oracle O_{ota}, it cannot obtain the output $\mathsf{ota}_4(1^\lambda,\mathsf{pk},\alpha,m^*||m_1,O_0(\gamma))$[10] by querying O_{ota} on α. This means that, in game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$, the random coin m^* is hidden in adversary's input by using the random oracle O_0 and random oracle O_1, the value $m^*||m_1$ is hidden by using O_{ota}.

4 Lifting Theorem for Oracle-Hiding Game

In this section, we give a lifting theorem for the oracle-hiding game from ROM to QROM.

4.1 Statement of Lifting Theorem

First, we introduce a lemma of the oracle-hiding game in the ROM, and its detailed proof is given in the next section.

Lemma 3. *For any oracle-hiding game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$ in the ROM, suppose that the query times of O_0, O_1 and O_{ota} are q_0, q_1 and q_{ota}, respectively. Then there exist adversary $\mathcal{A}_1(1^\lambda)$ and $\mathcal{A}_2(1^\lambda)$, which make no queries to oracles they have access to and invoke adversary $\mathcal{A}(1^\lambda)$ once in a black-box manner (without rewinding), such that*

$$\left|\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A},\mathcal{C}}(1^\lambda) - \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}_1,\mathcal{C}}(1^\lambda)\right| \leq q_{\mathsf{ota}} \cdot \mathsf{ota.max} + q_1 \cdot \mathsf{ota.union} + (q_0+q_1) \cdot \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}_2,\mathcal{C}_{\mathsf{find}}}(1^\lambda), \tag{16}$$

where challenger $\mathcal{C}_{\mathsf{find}}(1^\lambda)$ is identical with $\mathcal{C}(1^\lambda)$, except that it finally outputs $t = \mathsf{boole}[\mathsf{OHG.C} = m^*]$ as game's output. Moreover, the running time of $\mathcal{A}_1(1^\lambda)$ and that of $\mathcal{A}_2(1^\lambda)$ can be bounded by

$$\mathrm{Time}[\mathcal{A}_1(1^\lambda)] \approx \mathrm{Time}[\mathcal{A}_2(1^\lambda)] \leq \mathrm{Time}[\mathcal{A}(1^\lambda)] + (q_0+q_1)\cdot O(\lambda) + q_{\mathsf{ota}} \cdot \mathsf{ota.time}.$$

Remark 3. *The detailed construction of adversary $\mathcal{A}_1(1^\lambda)$ and $\mathcal{A}_2(1^\lambda)$ is complicated, and thus we omit them in Lemma 3. They are clearly described in the proof of Lemma 3 in the next section.*

[10] Here $\gamma = \mathsf{ota}_3(1^\lambda,\mathsf{pk},\alpha,m^*||m_1)$.

Then we present our lifting theorem for oracle-hiding game as follows.

Theorem 4 (Lifting Theorem for Oracle-Hiding Game)**.** *For any oracle-hiding game* $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{B}(1^\lambda),\mathcal{C}(1^\lambda)}$ *in the QROM, suppose that the query times of* O_0, O_1 *and* O_{ota} *are* q_0, q_1 *and* q_{ota}*, respectively.*

By mimicking the construction of adversary $\mathcal{A}_1(1^\lambda)$ *and* $\mathcal{A}_2(1^\lambda)$ *in Lemma 3, we can directly construct adversary* $\mathcal{B}_1(1^\lambda)$ *and* $\mathcal{B}_2(1^\lambda)$*, which make no query to the oracle they have access to and invoke adversary* $\mathcal{B}$ *once in a black-box manner (without rewinding) such that*

$$\begin{aligned}\left|\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{B},\mathcal{C}}(1^\lambda) - \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{B}_1,\mathcal{C}}(1^\lambda)\right| \le 40 q_{\mathsf{ota}} \cdot \sqrt{\mathsf{ota.max}} + 8(q_1+1)\cdot\sqrt{\mathsf{ota.union}} + 64 q_1 \cdot \mathsf{ota.union} \\ + 4(q_0+q_1+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{B}_2,\mathcal{C}_{\mathsf{find}}}(1^\lambda)}.\end{aligned} \tag{17}$$

where challenger $\mathcal{C}_{\mathsf{find}}(1^\lambda)$ *is identical with* $\mathcal{C}(1^\lambda)$*, except that it finally outputs* $t = \mathsf{boole}[\mathsf{OHG.C} = m^*]$ *as game's output. Moreover, the running time of* $\mathcal{B}_1(1^\lambda)$ *and that of* $\mathcal{B}_2(1^\lambda)$ *can be bounded by*

$$\mathrm{Time}[\mathcal{B}_1(1^\lambda)] \approx \mathrm{Time}[\mathcal{B}_2(1^\lambda)] \le \mathrm{Time}[\mathcal{B}(1^\lambda)] + O((q_0+q_1)\cdot q_{\mathsf{ota}} \cdot \mathsf{ota.time} + (q_0+q_1)^2).$$

Remark 4. *Similar with the Lemma 3, we omit the detailed construction of adversary* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *in Theorem 4 since they are complicated. In the proof of Theorem 4 in the next section, we will clearly give the detailed construction of adversary* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *and show that how to mimic the construction of adversary* $\mathcal{A}_1$ *(resp.* $\mathcal{A}_2$*) to get the construction of adversary* $\mathcal{B}_1$ *(resp.* $\mathcal{B}_2$*).*

Indeed, Theorem 4 shows that the adversary $\mathcal{B}_1$ and $\mathcal{B}_2$ satisfying Eq. (17) can be obtained by mimicking the construction of $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfying Eq. (16), respectively. It is also noted that the upper bound shown in Eq. (17) is almost identical with Eq. (16), except for a square-root advantage loss. In other words, Theorem 4 shows that the result on the oracle-hiding game in the ROM can be lifted to the QROM with a square-root advantage loss.

4.2 Proof of Lifting Theorem

In this section, we give the detailed proof of Lemma 3 and Theorem 4. For notational clarity, we sometimes omit the security parameter λ in the following text.

4.2.1 Proof of Lemma 3

Proof. The basic idea of this proof is to gradually change the simulation of random oracle O_0, random oracle O_1 and secret oracle O_{ota} by a sequence of games. The overview of all games is given in Fig. 3.

Game $\mathbf{G_0^c}$: This game is identical with the oracle-hiding game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}(1^\lambda),\mathcal{C}(1^\lambda)}$ in the ROM except that the random oracle O_0 and O_1 is simulated *on-the-fly* by using the query/reply record list $\mathfrak{L}_0$ and $\mathfrak{L}_1$, respectively.

Notice that the line 4 and line 5 of secret oracle O_{ota} in game $\mathbf{G_0^c}$ tests whether $O_1(\beta)$ belongs to $\mathsf{ota.sub}^{\alpha,\beta}_{\mathsf{pk}}$ to determine whether $\mathsf{ota}_2(\mathsf{pk},\beta,O_1(\beta))$ equals α. This is unproblematic since they are equivalent by the definition of the subset $\mathsf{ota.sub}^{\alpha,\beta}_{\mathsf{pk}}$ defined in Eq. (14). Then, we have

$$\Pr[1 \leftarrow \mathbf{G_0^c}] = \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A},\mathcal{C}}(1^\lambda). \tag{18}$$

Game $\mathbf{G_1^c}$: In this game, the simulation of secret oracle O_{ota} on query α is changed that it adds a new rule:

For the query α, if $\beta := \mathsf{ota}_1(\mathsf{sk},\alpha) \ne \perp$ and $\mathfrak{L}_1(\beta) = \perp$, return $\mathsf{f}_{\mathsf{ota}}(\alpha)$.

Here $\mathfrak{L}_1$ is the list just before the simulation of oracle O_{ota} on query α.

For any fixed $(\mathsf{pk},\mathsf{sk})$ that is generated by KGen, suppose the adversary's i-th query to secret oracle O_{ota} is α_i $(i = 1,\ldots,q_{\mathsf{ota}})$, define event DIFF_i^0 (resp. DIFF_i^1) as:

In game $\mathbf{G_0^c}$ (resp. game $\mathbf{G_1^c}$), α_i satisfies $\beta_i := \mathsf{ota}_1(\mathsf{sk},\alpha_i) \ne \perp$, $\mathfrak{L}_1(\beta_i) = \perp$ and $O_1(\beta_i) \in \mathsf{ota.sub}^{\alpha_i,\beta_i}_{\mathsf{pk}}$.

Here $\mathfrak{L}_1$ is the list just before the i-th O_{ota} query. By simulation, it is easily to check that the secret oracle O_{ota} in game $\mathbf{G}_0^{\mathbf{c}}$ and game $\mathbf{G}_1^{\mathbf{c}}$ will output same value for the i-th query α_i if event DIFF_i^0 and DIFF_i^1 do not occur. Thus, game $\mathbf{G}_0^{\mathbf{c}}$ and game $\mathbf{G}_1^{\mathbf{c}}$ proceed identically if event $\vee_{i=1}^{q_{\mathsf{ota}}}\mathsf{DIFF}_i^0$ and $\vee_{i=1}^{q_{\mathsf{ota}}}\mathsf{DIFF}_i^1$ do not occur. This implies that

$$\Pr\left[\vee_{i=1}^{q_{\mathsf{ota}}}\mathsf{DIFF}_i^0\right] = \Pr\left[\vee_{i=1}^{q_{\mathsf{ota}}}\mathsf{DIFF}_i^1\right],$$
$$\Pr[1 \leftarrow \mathbf{G}_0^{\mathbf{c}} : (\mathsf{pk},\mathsf{sk}) \wedge \neg(\vee_{i=1}^{q_{\mathsf{ota}}}\mathsf{DIFF}_i^0)] = \Pr[1 \leftarrow \mathbf{G}_1^{\mathbf{c}} : (\mathsf{pk},\mathsf{sk}) \wedge \neg(\vee_{i=1}^{q_{\mathsf{ota}}}\mathsf{DIFF}_i^1)].$$

Here $1 \leftarrow \mathbf{G}_0^{\mathbf{c}} : (\mathsf{pk},\mathsf{sk})$ denote the event that game $\mathbf{G}_0^{\mathbf{c}}$ finally return 1 for the fixed $(\mathsf{pk},\mathsf{sk})$. Then by the difference lemma of [Sho04],

$$|\Pr[1 \leftarrow \mathbf{G}_0^{\mathbf{c}} : (\mathsf{pk},\mathsf{sk})] - \Pr[1 \leftarrow \mathbf{G}_1^{\mathbf{c}} : (\mathsf{pk},\mathsf{sk})]| \leq \Pr\left[\vee_{i=1}^{q_{\mathsf{ota}}}\mathsf{DIFF}_i^1\right]. \tag{19}$$

GAMES $\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_4^{\mathbf{c}}$

1, $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_4^{\mathbf{c}}$
2, $\mathsf{OHG.A} \leftarrow \mathcal{A}^{O_0,O_1,O_{\mathsf{ota}}}(\mathsf{pk})$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_3^{\mathbf{c}}$
  $\mathsf{OHG.A} \leftarrow \mathcal{A}_1(\mathsf{pk})$ //$\mathbf{G}_4^{\mathbf{c}}$
3, $\mathcal{C}$ **perform following operation**
  $m^* \xleftarrow{\$} \mathcal{R}_1$, $r \xleftarrow{\$} \mathcal{R}_2$, $s \in \{0,1\}$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_4^{\mathbf{c}}$
  $m_s \leftarrow \mathsf{cha}_1(\mathsf{pk},\mathsf{OHG.A},m^*,r)$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_4^{\mathbf{c}}$
  $y_s = O_s(m^*||m_s)$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_2^{\mathbf{c}}$,$\mathbf{G}_4^{\mathbf{c}}$
  $y_s = r_s$ //$\mathbf{G}_3^{\mathbf{c}}$
  $m_{1-s} \leftarrow \mathsf{cha}_2(\mathsf{pk},\mathsf{OHG.A},y_s,m^*,r)$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_4^{\mathbf{c}}$
  $y_{1-s} = O_{1-s}(m^*||m_{1-s})$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_2^{\mathbf{c}}$,$\mathbf{G}_4^{\mathbf{c}}$
  $y_{1-s} = r_{1-s}$ //$\mathbf{G}_3^{\mathbf{c}}$
  $\mathsf{OHG.B} \leftarrow \mathsf{cha}_3(\mathsf{pk},\mathsf{OHG.A},y_s,y_{1-s},m^*,r)$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_4^{\mathbf{c}}$
4, $\mathsf{OHG.C} \leftarrow \mathcal{A}^{O_0,O_1,O_{\mathsf{ota}}}(\mathsf{pk},\mathsf{OHG.B})$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_3^{\mathbf{c}}$
  $\mathsf{OHG.C} \leftarrow \mathcal{A}_1(\mathsf{pk},\mathsf{OHG.B})$ //$\mathbf{G}_4^{\mathbf{c}}$
5, $t \leftarrow \mathsf{verify}(\mathsf{pk},\mathsf{sk},\mathsf{OHG.A},m^*,r,s,\mathsf{OHG.C})$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_4^{\mathbf{c}}$
  $\mathcal{C}$ **output** $t \in \{0,1\}$ **as game's output**

$O_0(x)$
1, $O \xleftarrow{\$} \mathcal{F}_{m,n}$, **return** $O(x)$ //$\mathbf{G}_0^{\mathbf{c}}$,$\mathbf{G}_4^{\mathbf{c}}$
2, **If** $\exists\, y$ **s.t.** $(x,y) \in \mathfrak{L}_0$, **return** y //$\mathbf{G}_1^{\mathbf{c}}$-$\mathbf{G}_3^{\mathbf{c}}$
3, **Else** $y \xleftarrow{\$} \{0,1\}^n$, $\mathfrak{L}_0 := \mathfrak{L}_0 \cup (x,y)$, //$\mathbf{G}_1^{\mathbf{c}}$-$\mathbf{G}_3^{\mathbf{c}}$
  return y

$O_1(x)$
1, $O' \xleftarrow{\$} \mathcal{F}_{m,n}$, **return** $O'(x)$ //$\mathbf{G}_0^{\mathbf{c}}$,$\mathbf{G}_4^{\mathbf{c}}$
2, **If** $\exists\, y$ **s.t.** $(x,y) \in \mathfrak{L}_1$, **return** y //$\mathbf{G}_1^{\mathbf{c}}$-$\mathbf{G}_3^{\mathbf{c}}$
3, **Else** $y \xleftarrow{\$} \{0,1\}^{n'}$, $\mathfrak{L}_1 := \mathfrak{L}_1 \cup (x,y)$, //$\mathbf{G}_1^{\mathbf{c}}$-$\mathbf{G}_3^{\mathbf{c}}$
  return y

$O_{\mathsf{ota}}(\alpha)$
1, **If** $\mathsf{OHG.B}$ **is defined and** //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_1^{\mathbf{c}}$,$\mathbf{G}_4^{\mathbf{c}}$
  $\alpha = \mathsf{ota}_2(1^\lambda,\mathsf{pk},m^*||m_1,O_1(m^*||m_1))$
  return $\perp$
2, **Else if** $\mathsf{ota}_1(\mathsf{sk},\alpha) = \perp$, **return** $\mathsf{f}_{\mathsf{ota}}(\alpha)$ //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_1^{\mathbf{c}}$,$\mathbf{G}_4^{\mathbf{c}}$
3, **Else if** $\beta := \mathsf{ota}_1(\mathsf{sk},\alpha) \neq \perp$ **and** //$\mathbf{G}_1^{\mathbf{c}}$
  $\mathfrak{L}_1(\beta) = \perp$, **return** $\mathsf{f}_{\mathsf{ota}}(\alpha)$
4, **Else if** $\beta := \mathsf{ota}_1(\mathsf{sk},\alpha) \neq \perp$ **and** //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_1^{\mathbf{c}}$,$\mathbf{G}_4^{\mathbf{c}}$
  $O_1(\beta) \notin \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$, **return** $\mathsf{f}_{\mathsf{ota}}(\alpha)$
5, **Else if** $\beta := \mathsf{ota}_1(\mathsf{sk},\alpha) \neq \perp$ **and** //$\mathbf{G}_0^{\mathbf{c}}$-$\mathbf{G}_1^{\mathbf{c}}$,$\mathbf{G}_4^{\mathbf{c}}$
  $O_1(\beta) \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$,
  compute $\gamma := \mathsf{ota}_3(\mathsf{pk},\alpha,\beta)$
  return $\mathsf{ota}_4(\mathsf{pk},\alpha,\beta,O_0(\gamma))$

$O_{\mathsf{ota}}(\alpha)$
1, **Return** $\mathsf{Search}(\mathfrak{L}_1,\alpha)$ //$\mathbf{G}_2^{\mathbf{c}}$-$\mathbf{G}_3^{\mathbf{c}}$

Figure 3: Summary of games for the proof of Lemma 3. The query/reply record list $\mathfrak{L}_0$ (resp. $\mathfrak{L}_1$) used to simulated random oracle O_0 (resp. O_1) is a set of pair $(x,y) \in \{0,1\}^m \times \{0,1\}^n$ (resp. $(x,y) \in \{0,1\}^{m'} \times \{0,1\}^{n'}$). Initially, list $\mathfrak{L}_0$ and $\mathfrak{L}_1$ are empty set. We say $\mathfrak{L}_1(x) = \perp$ if there does not exist y s.t. $(x,y) \in \mathfrak{L}_1$, we also denote y as $\mathfrak{L}_1(x)$ if a pair $(x,y) \in \mathfrak{L}_1$.

Note that $\mathfrak{L}_1(\beta_i) = \perp$ indicates β_i has never been queried to random oracle O_1 by the adversary, and hence $O_1(\beta_i)$ must be uniformly random in $\{0,1\}^{n'}$ by the basic rules of the on-the-fly simulation. Then we have

$$\begin{aligned}\Pr\left[\vee_{i=1}^{q_{\mathsf{ota}}}\mathsf{DIFF}_i^1\right] \leq \sum_{i=1}^{q_{\mathsf{ota}}}\Pr\left[\mathsf{DIFF}_i^1\right] &\leq \sum_{i=1}^{q_{\mathsf{ota}}}\Pr[O_1(\beta_i) \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha_i,\beta_i} : \mathfrak{L}_1(\beta_i) = \perp] \\ &\leq q_{\mathsf{ota}} \cdot \max_{\alpha\in\mathcal{X},\beta\in\{0,1\}^{m'}} \frac{1}{2^{n'}}\left|\mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}\right|.\end{aligned} \tag{20}$$

Combining Eq. (19) with Eq. (20) and then averaging over $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$, we finally obtain

$$\begin{aligned}|\Pr[1 \leftarrow \mathbf{G}_0^{\mathbf{c}}] - \Pr[1 \leftarrow \mathbf{G}_1^{\mathbf{c}}]| &\leq q_{\mathsf{ota}} \cdot \mathop{\mathbb{E}}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KGen}} \frac{1}{2^{n'}} \max_{\alpha\in\mathcal{X},\beta\in\{0,1\}^{m'}}\left|\mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}\right| \\ &\stackrel{(a)}{=} q_{\mathsf{ota}} \cdot \mathsf{ota.max}.\end{aligned} \tag{21}$$

Here (a) uses Eq. (15).

Game $\mathbf{G_2^c}$: In this game, the secret oracle O_{ota} is simulated by using the operation Search, which is operated on input $(\mathfrak{L}_1, \alpha)$ as follows:

1. If $\mathsf{OHG.B}$ is defined and $\alpha = \mathsf{ota}_2(\mathsf{pk}, m^*||m_1, O_1(m^*||m_1))$, return $\perp$.
2. Else do: Find the smallest β such that $\mathfrak{L}_1(\beta) \neq \perp$ and $\mathfrak{L}_1(\beta) \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$. If such β exists, compute $\gamma := \mathsf{ota}_3(\mathsf{pk}, \alpha, \beta)$ and return $\mathsf{ota}_4(\mathsf{pk}, \alpha, \beta, O_0(\gamma))$, else return $\mathsf{f}_{\mathsf{ota}}(\alpha)$.

Notice that by Definition 5, whether $\alpha = \mathsf{ota}_2(\mathsf{pk}, m^*||m_1, O_1(m^*||m_1))$ can be determined by using $\mathsf{OHG.B}$ and only pk, thus the simulation of secret oracle O_{ota} in game $\mathbf{G_2^c}$ makes no use of the secret key sk any more.

In the following analysis, we consider a fixed $(\mathsf{pk}, \mathsf{sk})$ that is generated by KGen. In game $\mathbf{G_1^c}$, the simulation of secret oracle O_{ota} still uses secret key sk since it needs to compute $\mathsf{ota}_1(\mathsf{sk}, \alpha)$ for query α, and we also observe that O_{ota} does not directly return $\mathsf{f}_{\mathsf{ota}}(\alpha)$ for query α only when $\mathfrak{L}_1(\beta) \neq \perp$ and $\mathfrak{L}_1(\beta) \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$, where $\beta := \mathsf{ota}_1(\mathsf{sk}, \alpha) \neq \perp$. This means that the value $\mathsf{ota}_1(\mathsf{sk}, \alpha)$ must be already recorded in the list $\mathfrak{L}_1$ if O_{ota} does not directly return $\perp$ for query α. Based on this observation, in game $\mathbf{G_2^c}$, we use operation Search to extract $\mathsf{ota}_1(\mathsf{sk}, \alpha)$ from the list $\mathfrak{L}_1$ and to avoid computing $\mathsf{ota}_1(\mathsf{sk}, \alpha)$ like game $\mathbf{G_1^c}$ when we simulate secret oracle O_{ota} on query α.

In order to bound the difference between the probability that game $\mathbf{G_1^c}$ and game $\mathbf{G_2^c}$ output 1, we need to analyze under what conditions the output of the secret oracle O_{ota} in game $\mathbf{G_1^c}$ and game $\mathbf{G_2^c}$ are different. Indeed, the secret oracle O_{ota} in game $\mathbf{G_1^c}$ and game $\mathbf{G_2^c}$ only have different output on query α if α and the list $\mathfrak{L}_1$ just before this query are following cases:

1. $\mathsf{ota}_1(\mathsf{sk}, \alpha) = \perp$, and there exists a β s.t. $\mathfrak{L}_1(\beta) \neq \perp$ and $\mathfrak{L}_1(\beta) \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$.
2. $\beta := \mathsf{ota}_1(\mathsf{sk}, \alpha) \neq \perp$, $\mathfrak{L}_1(\beta) = \perp$, and there exists a β' s.t. $\mathfrak{L}_1(\beta') \neq \perp$ and $\mathfrak{L}_1(\beta') \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'}$.
3. $\beta := \mathsf{ota}_1(\mathsf{sk}, \alpha) \neq \perp$, $\mathfrak{L}_1(\beta) \neq \perp$, $\mathfrak{L}_1(\beta) \notin \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$, and there exists a β' s.t. $\mathfrak{L}_1(\beta') \neq \perp$ and $\mathfrak{L}_1(\beta') \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'}$.
4. $\beta := \mathsf{ota}_1(\mathsf{sk}, \alpha) \neq \perp$, $\mathfrak{L}_1(\beta) \neq \perp$, $\mathfrak{L}_1(\beta) \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$, and there exists a β' s.t. $\beta' < \beta$, $\mathfrak{L}_1(\beta') \neq \perp$ and $\mathfrak{L}_1(\beta') \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'}$.

We note that the list $\mathfrak{L}_1$ in above four cases both satisfy the property that there exist α and β' s.t. $\beta' \neq \mathsf{ota}_1(\mathsf{sk}, \alpha)$, $\mathfrak{L}_1(\beta') \neq \perp$ and $\mathfrak{L}_1(\beta') \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'}$, we will call list $\mathfrak{L}_1$ a bad list if it satisfies this property in the following. Then we can conclude that the secret oracle O_{ota} in game $\mathbf{G_1^c}$ and game $\mathbf{G_2^c}$ will output the same value on any query α if the list $\mathfrak{L}_1$ just before this query is not a bad list.

Let BAD_1 (resp. BAD_2) be the event that in once query of secret oracle O_{ota} in game $\mathbf{G_1^c}$ (resp. game $\mathbf{G_2^c}$), the list $\mathfrak{L}_1$ just before this query is a bad list. Hence, if event BAD_1 and BAD_2 do not occur, game $\mathbf{G_1^c}$ and game $\mathbf{G_2^c}$ proceed identically. This implies that

$$\Pr[\mathsf{BAD}_1] = \Pr[\mathsf{BAD}_2],$$
$$\Pr[1 \leftarrow \mathbf{G_1^c} : (\mathsf{pk}, \mathsf{sk}) \wedge \neg \mathsf{BAD}_1] = \Pr[1 \leftarrow \mathbf{G_2^c} : (\mathsf{pk}, \mathsf{sk}) \wedge \neg \mathsf{BAD}_2].$$

Then by the difference lemma of [Sho04],

$$|\Pr[1 \leftarrow \mathbf{G_1^c} : (\mathsf{pk}, \mathsf{sk})] - \Pr[1 \leftarrow \mathbf{G_2^c} : (\mathsf{pk}, \mathsf{sk})]| \leq \Pr[\mathsf{BAD}_2]. \tag{22}$$

In game $\mathbf{G_1^c}$ and game $\mathbf{G_2^c}$, we note that the simulation of secret oracle O_{ota} does not change the list $\mathfrak{L}_1$ and only the simulation of random oracle O_1 will update the list $\mathfrak{L}_1$. Let BAD' be the event that in game $\mathbf{G_2^c}$, just after once simulation of random oracle O_1, the list $\mathfrak{L}_1$ becomes a bad list. Let BAD'_i $(1 \leq i \leq q_1)$ be the event that in game $\mathbf{G_2^c}$, $\mathfrak{L}_1$ is not a bad list during the first $i-1$ times simulation of random oracle O_1, but becomes a bad list just after the i-th simulation[11]. Then

$$\Pr[\mathsf{BAD}_2] \leq \Pr[\mathsf{BAD}'] = \sum_{i=1}^{q_1} \Pr[\mathsf{BAD}'_i]. \tag{23}$$

[11]Since the initial list $\mathfrak{L}_1$ is an empty set and obvious not a bad list, BAD'_1 actually the event that in game $\mathbf{G_2}$, just after the 1-th simulation of random oracle O_1, the list $\mathfrak{L}_1$ becomes a bad list.

Notice that a non-bad list $\mathfrak{L}_1$ satisfies that there is no α and β' s.t. $\beta' \neq \mathsf{ota}_1(\mathsf{sk}, \alpha)$, $\mathfrak{L}_1(\beta') \neq \perp$ and $\mathfrak{L}_1(\beta') \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'}$. Hence once event BAD'_i occurs, suppose the i-th query of random oracle O_1 is β', then we can conclude that a pair $(\beta', \mathfrak{L}_1(\beta'))$ must be added after the i-th simulation of random oracle O_1 and this pair satisfies that there exists a α s.t. $\beta' \neq \mathsf{ota}_1(sk, \alpha)$, $\mathfrak{L}_1(\beta') \neq \perp$ and $\mathfrak{L}_1(\beta') \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'}$. In other word, the $(\beta', \mathfrak{L}_1(\beta'))$ newly added must satisfies $\mathfrak{L}_1(\beta') \in \underset{\alpha \in \mathsf{Set}.\beta'}{\cup} \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'}$, where set $\mathsf{Set}.\beta' := \{\alpha \in \mathcal{X} : \mathsf{ota}_1(sk, \alpha) \neq \beta'\}$. For the newly added $(\beta', \mathfrak{L}_1(\beta'))$, $\mathfrak{L}_1(\beta')$ is uniformly random in $\{0,1\}^{n'}$ by the basic rules of the *on-the-fly* simulation, then we have

$$\Pr[\mathsf{BAD}'_i] \leq \frac{1}{2^{n'}} \max_{\beta' \in \{0,1\}^{m'}} \left| \underset{\alpha \in \mathsf{Set}.\beta'}{\cup} \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'} \right|. \tag{24}$$

Combining Eq. (22), (23), (24) and then averaging over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$, we finally obtain

$$\begin{aligned} |\Pr[1 \leftarrow \mathbf{G_1^c}] - \Pr[1 \leftarrow \mathbf{G_2^c}]| &\leq q_1 \cdot \underset{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}}{\mathbb{E}} \frac{1}{2^{n'}} \max_{\beta' \in \{0,1\}^{m'}} \left| \underset{\alpha \in \mathsf{Set}.\beta'}{\cup} \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta'} \right| \\ &\overset{(b)}{=} q_1 \cdot \mathsf{ota.union}. \end{aligned} \tag{25}$$

Here (b) uses Eq. (15).

<u>Game $\mathbf{G_3^c}$</u>: This game is the same game as game $\mathbf{G_2^c}$, except that we replace the value of y_0 (resp. y_1) used to generate $\mathsf{OHG.B}$ with r_0 (resp. r_1) uniformly sampled from $\{0,1\}^n$ (resp. $\{0,1\}^{n'}$).

After $\mathsf{OHG.B}$ is defined in game $\mathbf{G_2^c}$, the list $\mathfrak{L}_1$ can be written as $\mathfrak{L}_1 := \mathfrak{L}'_1 \cup \{(m^*||m_1, y_1)\}$ since the challenger queried random oracle O_1 on input $m^*||m_1$. Note that the operation Search will directly return $\perp$ after $\mathsf{OHG.B}$ is defined if the input $\alpha = \mathsf{ota}_2(\mathsf{pk}, m^*||m_1, O_1(m^*||m_1))$, by the construction of Search, this makes the output of Search on any input $(\mathfrak{L}_1, \alpha)$ cannot be $\mathsf{ota}_4(\mathsf{pk}, \alpha, m^*||m_1, O_0(\gamma))$, where $\gamma = \mathsf{ota}_3(\mathsf{pk}, \alpha, m^*||m_1)$. Thus we can conclude that after $\mathsf{OHG.B}$ is defined in game $\mathbf{G_2^c}$, the adversary cannot get the information about $(m^*||m_1, y_1)$ by making queries to the secret oracle O_{ota}.

Hence, if the random oracle O_0 and O_1 in game $\mathbf{G_2^c}$ is never queried by the adversary with input form of $m^*||*$, the $O_0(m^*||m_0)$ and $O_1(m^*||m_1)$ used by the challenger to generate $\mathsf{OHG.B}$ is uniformly random in adversary's view. Let QUERY_2 (resp. QUERY_3) be an event as:

In game $\mathbf{G_2^c}$ (resp. game $\mathbf{G_3^c}$), the random oracle O_0 and O_1 is ever queried by the adversary with input form of $m^*||*$,

now we can conclude that game $\mathbf{G_2^c}$ and game $\mathbf{G_3^c}$ proceed identically if event QUERY_2 and QUERY_3 do not occur. This implies that

$$\begin{aligned} \Pr[\mathsf{QUERY}_2] &= \Pr[\mathsf{QUERY}_3], \\ \Pr[1 \leftarrow \mathbf{G_2^c} \wedge \neg\mathsf{QUERY}_2] &= \Pr[1 \leftarrow \mathbf{G_3^c} \wedge \neg\mathsf{QUERY}_3]. \end{aligned}$$

Then by the difference lemma of [Sho04],

$$|\Pr[1 \leftarrow \mathbf{G_2^c}] - \Pr[1 \leftarrow \mathbf{G_3^c}]| \leq \Pr[\mathsf{QUERY}_3]. \tag{26}$$

<u>Game $\mathbf{G_4^c}$</u>: This game is the same game as game $\mathbf{G_3^c}$, except that the following changes:

- The adversary is changed to a new adversary $\mathcal{A}_1$, it does not query any oracles and invokes adversary $\mathcal{A}$ once in a black-box manner (without rewinding) as follows:
  1. After get the public key pk, invoke adversary $\mathcal{A}$ to get $\mathsf{OHG.A}$ and send it to the challenger. After get the $\mathsf{OHG.B}$ computed by the challenger, invoke adversary $\mathcal{A}$ to get $\mathsf{OHG.C}$ and send it to the challenger. The oracle queries performed by $\mathcal{A}$ is answer as:
     (a) When the random oracle O_0 (resp. O_1) is queried by $\mathcal{A}$, $\mathcal{A}_1$ answer it on-the-fly by using the query/reply list $\mathfrak{L}_0$ (resp. $\mathfrak{L}_1$).
     (b) When the secret oracle O_{ota} is queried by $\mathcal{A}$, $\mathcal{A}_1$ answer it by the operation Search as the game $\mathbf{G_3^c}$.

- The random oracle O_0, random oracle O_1 and secret oracle O_{ota} in game $\mathbf{G_4^c}$ is simulated the same as game $\mathbf{G_0^c}$[12], and the value of y_s (resp. y_{1-s}) used to generate $\mathsf{OHG.B}$ in game $\mathbf{G_4^c}$ is replaced with $O_s(m^*||m_s)$ (resp. $O_{1-s}(m^*||m_{1-s})$).

Compared with game $\mathbf{G_3^c}$, the change in game $\mathbf{G_4^c}$ is only conceptual. Thus, let QUERY_4 be the event that the adversary $\mathcal{A}_1$ in game $\mathbf{G_4^c}$ ever answered a query to the random oracle O_0 or O_1 with the input form of $m^*||*$, we have

$$\Pr\left[\mathsf{QUERY}_3\right] = \Pr\left[\mathsf{QUERY}_4\right], \ \Pr[1 \leftarrow \mathbf{G_3^c}] = \Pr[1 \leftarrow \mathbf{G_4^c}]. \tag{27}$$

Moreover, we observe that game $\mathbf{G_4^c}$ is identical with game $\mathbf{G_0^c}$ except that the adversary is replaced to $\mathcal{A}_1$, then game $\mathbf{G_4^c}$ is the oracle-hiding game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}_1(1^\lambda),\mathcal{C}(1^\lambda)}$ and

$$\Pr[1 \leftarrow \mathbf{G_4^c}] = \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}_1,\mathcal{C}}(1^\lambda). \tag{28}$$

As for the probability that event QUERY_4 occurs, we consider oracle-hiding game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}_2(1^\lambda),\mathcal{C}(1^\lambda)}$ with a new challenger $\mathcal{C}_{\mathsf{find}}$ and a new adversary $\mathcal{A}_2$ as follows:

- The challenger $\mathcal{C}_{\mathsf{find}}$ is identical with $\mathcal{C}$ except that $\mathcal{C}_{\mathsf{find}}$ finally output $t = \mathsf{boole}[\mathsf{OHG.C} = m^*]$ as game's output.

- The adversary $\mathcal{A}_2$ is identical with $\mathcal{A}_1$, except that $\mathcal{A}_2$ picks $i \xleftarrow{\$} \{1, \ldots, q_0 + q_1\}$ at everything begins and record the i-th random oracle query $m'||*$ it needs to answer, where m' have the same length as m^*. Then $\mathcal{A}_2$ output $\mathsf{OHG.C} = m'$.

One can check that if QUERY_4 occurs, the oracle-hiding game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{A}_2(1^\lambda),\mathcal{C}(1^\lambda)}$ will output 1 with probability $1/(q_0 + q_1)$, hence we obtain

$$\Pr\left[\mathsf{QUERY}_4\right] \leq (q_0 + q_1) \cdot \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}_2,\mathcal{C}_{\mathsf{find}}}(1^\lambda). \tag{29}$$

Tracing through the above game sequence from game $\mathbf{G_0^c}$ to $\mathbf{G_4^c}$, combining Eq. (18), (21), (25), (26), (27), (28) and (29), we finally obtain

$$\left|\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A},\mathcal{C}}(1^\lambda) - \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}_1,\mathcal{C}}(1^\lambda)\right| \leq q_{\mathsf{ota}} \cdot \mathsf{ota.max} + q_1 \cdot \mathsf{ota.union} + (q_0 + q_1) \cdot \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}_2,\mathcal{C}_{\mathsf{find}}}(1^\lambda).$$

As for the running time of $\mathcal{A}_1$ and $\mathcal{A}_2$, by their construction, we know that they invoke adversary $\mathcal{A}$ only once and simulate random oracle O_0 (resp. O_1) *on-the-fly* q_0 (resp. q_1) times, simulate secret oracle O_{ota} by operation Search q_{ota} times, hence we have

$$\mathrm{Time}[\mathcal{A}_1(1^\lambda)] \approx \mathrm{Time}[\mathcal{A}_2(1^\lambda)] \leq \mathrm{Time}[\mathcal{A}(1^\lambda)] + (q_0 + q_1) \cdot O(\lambda) + q_{\mathsf{ota}} \cdot \mathsf{ota.time}.$$

The definition of $\mathsf{ota.time}$ is given in Definition 4. □

4.2.2 Proof of Theorem 4

Before we prove Theorem 4, we first show that how to simulate quantum accessible secret oracle O_{ota} for an oracle-hiding game in the QROM. The notation and simulation method introduced here will be used in the proof of Theorem 4.

Since secret oracle O_{ota} is mainly processed by the oracle-testing algorithm $\mathsf{ota}^{O_0,O_1}(\mathsf{sk},\cdot)$ (Definition 4), we first consider how to evaluate $\mathsf{ota}^{O_0,O_1}(\mathsf{sk},\cdot)$ in superposition. Let $\mathsf{X}_{\mathsf{ota}}$ be the adversary's input register of secret oracle O_{ota} defined over $\mathcal{X}$, let Y be a quantum register defined over $\{0,1\}^{m'+1}$[13]. Define unitary operation $\mathrm{U}_{\mathsf{test}}$ acts on registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}$ as

$$\mathrm{U}_{\mathsf{test}}|\alpha\rangle|0^m\rangle := \begin{cases} |\alpha\rangle|\beta\rangle & \text{if } \beta := \mathsf{ota}_1(\mathsf{sk},\alpha) \neq \bot \wedge \mathsf{ota}_2(\mathsf{pk},\beta,O_1(\beta)) = \alpha \\ |\alpha\rangle|\bot\rangle & \text{otherwise.} \end{cases} \tag{30}$$

[12] To avoid confusion, we stress that this O_0, O_1 and O_{ota} are oracles queried in game $\mathbf{G_4^c}$, they are independent with the oracle O_0, O_1 and O_{ota} appeared in the description of adversary $\mathcal{A}_1$.

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

Intuitively, $\mathrm{U}_{\mathsf{test}}$ can implement all the test performed by $\mathsf{ota}^{O_0,O_1}(\mathsf{sk},\cdot)$ in superposition, hence what we need to do next is to compute the output of $\mathsf{ota}^{O_0,O_1}(\mathsf{sk},\cdot)$ by using the β computed by $\mathrm{U}_{\mathsf{test}}$. Let $\mathsf{Y}_{\mathsf{ota}}$ be the adversary's output register of secret oracle O_{ota} defined over $\{0,1\}^{l+1}$[14], define unitary operation $\mathrm{U}_{\mathsf{comp}}$ acts on registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}_{\mathsf{ota}}\mathsf{Y}$ as

$$\mathrm{U}_{\mathsf{comp}}|\alpha\rangle|y\rangle|\beta\rangle := \begin{cases} |\alpha\rangle|y \oplus \mathsf{ota}_4(\mathsf{pk},\alpha,\beta,O_0(\gamma))\rangle|\beta\rangle & \text{if } \beta \neq \perp \\ |\alpha\rangle|y \oplus \mathsf{f}_{\mathsf{ota}}(\alpha)\rangle|\beta\rangle & \text{if } \beta = \perp. \end{cases} \tag{31}$$

Here $\gamma := \mathsf{ota}_3(\mathsf{pk},\alpha,\beta)$. The detailed quantum circuit implementation of $\mathrm{U}_{\mathsf{test}}$ and $\mathrm{U}_{\mathsf{comp}}$ is given in Appendix E, which twice queries to random oracle O_1 and random oracle O_0 is needed, respectively. Then, the quantum accessible secret oracle O_{ota} can be simulated as follows:

- If the $\mathsf{OHG.B}$ is not defined, unitary operation

$$\mathrm{U}_{\mathsf{ota}} := \mathrm{U}_{\mathsf{test}}^{\dagger} \circ \mathrm{U}_{\mathsf{comp}} \circ \mathrm{U}_{\mathsf{test}}$$

  is applied to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}_{\mathsf{ota}}\mathsf{Y}$.

- If the $\mathsf{OHG.B}$ is defined, unitary operation

$$\mathrm{U}_{\mathsf{ota}}^{*} := \mathrm{U}_{\perp} \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})$$

  is applied to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}_{\mathsf{ota}}\mathsf{Y}$.

Here the register Y is initialized with state $|0^m\rangle$ for everything begins, $\mathrm{P}_{\mathsf{hide}} := |y\rangle\langle y|$, where $y = \mathsf{ota}_2(\mathsf{pk}, m^*||m_1, O_1(m^*||m_1))$, is a projector acts on register $\mathsf{X}_{\mathsf{ota}}$, $\mathrm{U}_{\perp}$ is a unitary operation acts on register $\mathsf{Y}_{\mathsf{ota}}$ that maps $|y\rangle$ to $|y \oplus \perp\rangle$. By the construction of $\mathrm{U}_{\mathsf{ota}}$, we observe that the register Y always in state $|0^m\rangle$ before and after once simulation of secret oracle O_{ota}.

Proof. Similar to the proof of Lemma 3, the basic idea of this proof is to gradually change the simulation of random oracle O_0, random oracle O_1 and secret oracle O_{ota} by a sequence of games. Note that O_0, O_1 and O_{ota} can be quantum accessed if the oracle-hiding game in the QROM, hence we actually consider the quantum simulation of O_0, O_1 and O_{ota} in this proof, which is different with the proof of Lemma 3. The overview of all games is given in Fig. 4.

Game $\mathbf{G_0^q}$: This game is identical with the oracle-hiding game $\mathsf{OHG}_{\mathcal{B}(1^\lambda),\mathcal{C}(1^\lambda)}^{O_0,O_1,O_{\mathsf{ota}}}$ in the QROM except that following changes:

- The random oracle O_0 and O_1 is simulated by the unitary operation U_O and $\mathrm{U}_{O'}$, respectively.
- The secret oracle O_{ota} is simulated by $\mathrm{U}_{\mathsf{ota}}$ and $\mathrm{U}_{\mathsf{ota}}^{*}$ defined above before and after $\mathsf{OHG.B}$ is defined, respectively.

Obviously,

$$\Pr[1 \leftarrow \mathbf{G_0^q}] = \mathsf{Adv}_{\mathcal{B},\mathcal{C}}^{\mathsf{OHG}}(1^\lambda). \tag{32}$$

Game $\mathbf{G_1^q}$: Compare with game $\mathbf{G_0^q}$, there are only two changes as:

- The random oracle O_0 is simulated by unitary operation U_f, where $f : \{0,1\}^m \to \{0,1\}^n$ is a $2q_0$-wise independent function.
- Let D_{q_1} be the database register defined over set $\mathbf{D}_{q_1}$ (Section 2.4). Let $\mathcal{S}(f_1)$ be the extractable RO-simulator defined in Section 2.5 with internal database register D_{q_1}, where function $f_1 : \{0,1\}^{m'} \times \{0,1\}^{n'} \to \mathcal{X} \cup \perp$ is

$$f_1(x,y) = \begin{cases} z & \text{if } \mathsf{ota}_2(\mathsf{pk},x,y) = z \wedge \mathsf{ota}_1(\mathsf{sk},z) = x \\ \perp & \text{otherwise.} \end{cases}$$

  The random oracle O_1 in game $\mathbf{G_1^q}$ is simulated by invoking the RO-interface $\mathsf{eCO.RO}$ of $\mathcal{S}(f_1)$.

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

Since the extraction-interface $\mathsf{eCO.E}_{f_1}$ of $\mathcal{S}(f_1)$ is never used and the random oracle O_0 and O_1 are queried at most q_0 and q_1 times, respectively, above simulations are perfect by Lemma 9 and Lemma 1. Hence

$$\Pr[1 \leftarrow \mathbf{G_0^q}] = \Pr[1 \leftarrow \mathbf{G_1^q}]. \tag{33}$$

In game $\mathbf{G_1^q}$, we stress that the secret oracle O_{ota} is simulated by

$$\mathrm{U}^1_{\mathsf{ota}} := \tilde{\mathrm{U}}^\dagger_{\mathsf{test}} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \tilde{\mathrm{U}}_{\mathsf{test}} \text{ and } \mathrm{U}^{1,*}_{\mathsf{ota}} := \mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}^1_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})$$

before and after $\mathsf{OHG.B}$ is defined, respectively. Here $\tilde{\mathrm{U}}_{\mathsf{test}}$ (resp. $\tilde{\mathrm{U}}_{\mathsf{comp}}$) have the identical implementation with $\mathrm{U}_{\mathsf{test}}$ (resp. $\mathrm{U}_{\mathsf{comp}}$) except that the internal twice queries to random oracle O_1 (resp. O_0) is simulated by U_f (resp. $\mathsf{eCO.RO}$).

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

1, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$ //$\mathbf{G_0^q}$-$\mathbf{G_5^q}$

2, $\mathsf{OHG.A} \leftarrow \mathcal{B}^{O_0,O_1,O_{\mathsf{ota}}}(\mathsf{pk})$ //$\mathbf{G_0^q}$-$\mathbf{G_5^q}$

  $\mathsf{OHG.A} \leftarrow \mathcal{B}_1(\mathsf{pk})$ //$\mathbf{G_6^q}$

3, $\mathcal{C}$ **perform following operation**

  $m^* \xleftarrow{\$} \mathcal{R}_1$, $r \xleftarrow{\$} \mathcal{R}_2$, $s \in \{0,1\}$ //$\mathbf{G_0^q}$-$\mathbf{G_5^q}$

  $m_s \leftarrow \mathsf{cha}_1(\mathsf{pk}, \mathsf{OHG.A}, m^*, r)$ //$\mathbf{G_0^q}$-$\mathbf{G_5^q}$

  $y_s = O_s(m^*||m_s)$ //$\mathbf{G_0^q}$-$\mathbf{G_4^q}$,$\mathbf{G_6^q}$

  $y_s = r_s$ //$\mathbf{G_5^q}$

  $m_{1-s} \leftarrow \mathsf{cha}_2(\mathsf{pk}, \mathsf{OHG.A}, y_s, m^*, r)$ //$\mathbf{G_0^q}$-$\mathbf{G_5^q}$

  $y_{1-s} = O_{1-s}(m^*||m_{1-s})$ //$\mathbf{G_0^q}$-$\mathbf{G_4^q}$,$\mathbf{G_6^q}$

  $y_{1-s} = r_{1-s}$ //$\mathbf{G_5^q}$

  $\mathsf{OHG.B} \leftarrow \mathsf{cha}_3(\mathsf{pk}, \mathsf{OHG.A}, y_s, y_{1-s}, m^*, r)$ //$\mathbf{G_0^q}$-$\mathbf{G_5^q}$

4, $\mathsf{OHG.C} \leftarrow \mathcal{B}^{O_0,O_1,O_{\mathsf{ota}}}(\mathsf{pk}, \mathsf{OHG.B})$ //$\mathbf{G_0^q}$-$\mathbf{G_5^q}$

  $\mathsf{OHG.C} \leftarrow \mathcal{B}_1(\mathsf{pk}, \mathsf{OHG.B})$ //$\mathbf{G_6^q}$

5, $t \leftarrow \mathsf{verify}(\mathsf{pk}, \mathsf{sk}, \mathsf{OHG.A}, m^*, r, s, \mathsf{OHG.C})$ //$\mathbf{G_0^q}$-$\mathbf{G_6^q}$

  $\mathcal{C}$ **output** t **as game's output**

$O_0(|x,y\rangle)$

1, $O \xleftarrow{\$} \mathcal{F}_{m,n}$, **return** //$\mathbf{G_0^q}$, $\mathbf{G_6^q}$

  $\mathrm{U}_O|x,y\rangle := |x, y \oplus O(x)\rangle$

2, **Return** $\mathrm{U}_f|x,y\rangle := |x, y \oplus f(x)\rangle$ //$\mathbf{G_1^q}$-$\mathbf{G_5^q}$

$O_1(|x,y\rangle)$

1, $O' \xleftarrow{\$} \mathcal{F}_{m',n'}$, **return** //$\mathbf{G_0^q}$, $\mathbf{G_6^q}$

  $\mathrm{U}_{O'}|x,y\rangle := |x, y \oplus O'(x)\rangle$

2, **Query** $\mathsf{eCO.RO}$ by $|x,y\rangle$ //$\mathbf{G_1^q}$-$\mathbf{G_5^q}$

$O_{\mathsf{ota}}(|\alpha,\beta\rangle)$

1, **If** $\mathsf{OHG.B}$ **is not defined, return**

  $\mathrm{U}_{\mathsf{ota}}|\alpha,\beta\rangle = \mathrm{U}^\dagger_{\mathsf{test}} \circ \mathrm{U}_{\mathsf{comp}} \circ \mathrm{U}_{\mathsf{test}}|\alpha,\beta\rangle$ //$\mathbf{G_0^q}$, $\mathbf{G_6^q}$

  $\mathrm{U}^1_{\mathsf{ota}}|\alpha,\beta\rangle = \tilde{\mathrm{U}}^\dagger_{\mathsf{test}} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \tilde{\mathrm{U}}_{\mathsf{test}}|\alpha,\beta\rangle$ //$\mathbf{G_1^q}$

  $\mathrm{U}^2_{\mathsf{ota}}|\alpha,\beta\rangle = \mathsf{eCO.E}_{f_1} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \mathsf{eCO.E}_{f_1}|\alpha,\beta\rangle$ //$\mathbf{G_2^q}$

  $\mathrm{U}^3_{\mathsf{ota}}|\alpha,\beta\rangle = \mathsf{eCO.E}_{f_2} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \mathsf{eCO.E}_{f_2}|\alpha,\beta\rangle$ //$\mathbf{G_3^q}$

  $\mathrm{U}^4_{\mathsf{ota}}|\alpha,\beta\rangle = \mathsf{eCO.E}'_{f_2} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \mathsf{eCO.E}'_{f_2}|\alpha,\beta\rangle$ //$\mathbf{G_4^q}$

  $\mathrm{U}^5_{\mathsf{ota}}|\alpha,\beta\rangle = \mathsf{eCO.E}_{f_2} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \mathsf{eCO.E}_{f_2}|\alpha,\beta\rangle$ //$\mathbf{G_5^q}$

  Else return

  $\mathrm{U}^*_{\mathsf{ota}}|\alpha,\beta\rangle = (\mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}}))|\alpha,\beta\rangle$ //$\mathbf{G_0^q}$, $\mathbf{G_6^q}$

  $\mathrm{U}^{1,*}_{\mathsf{ota}}|\alpha,\beta\rangle = (\mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}^1_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}}))|\alpha,\beta\rangle$ //$\mathbf{G_1^q}$

  $\mathrm{U}^{2,*}_{\mathsf{ota}}|\alpha,\beta\rangle = (\mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}^2_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}}))|\alpha,\beta\rangle$ //$\mathbf{G_2^q}$

  $\mathrm{U}^{3,*}_{\mathsf{ota}}|\alpha,\beta\rangle = (\mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}^3_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}}))|\alpha,\beta\rangle$ //$\mathbf{G_3^q}$

  $\mathrm{U}^{4,*}_{\mathsf{ota}}|\alpha,\beta\rangle = (\mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}^4_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}}))|\alpha,\beta\rangle$ //$\mathbf{G_4^q}$

  $\mathrm{U}^{5,*}_{\mathsf{ota}}|\alpha,\beta\rangle = (\mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}^5_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}}))|\alpha,\beta\rangle$ //$\mathbf{G_5^q}$

$\mathcal{S}(f) = \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_1}/\mathsf{eCO.E}_{f_2}/\mathsf{eCO.E}'_{f_2}\}$

1, $\mathsf{eCO.RO}$: **apply unitary operation** CStO

2, $\mathsf{eCO.E}_{f_1}$: **apply unitary operation** Ext_{f_1}

  $\mathsf{eCO.E}_{f_2}$: **apply unitary operation** Ext_{f_2}

  $\mathsf{eCO.E}'_{f_2}$: **apply unitary operation**

    $\mathsf{StdDecomp}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{StdDecomp}_{m^*||m_1}$

Figure 4: Summary of games for the proof of Theorem 4. Note that the oracle O_0, O_1 and O_{ota} in these games can be quantum accessed, for brevity, we just write the input state of O_0 and O_1 both as $|x,y\rangle$ and the input state of O_{ota} as $|\alpha,y\rangle$.

Game $\mathbf{G_2^q}$: This game is the same as game $\mathbf{G_1^q}$, except that the performing of $\tilde{\mathrm{U}}_{\mathsf{test}}$ on registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}$ is replaced by invoking the extraction-interface $\mathsf{eCO.E}_{f_1}$ on registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}$ in the simulation of secret oracle O_{ota}.

By the Definition 3, a query to $\mathsf{eCO.E}_{f_1}$ with registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}$ is processed by applying unitary operation

$$\mathsf{Ext}_{f_1} := \sum_{\alpha \in \mathcal{X}} |\alpha\rangle\langle\alpha|_{\mathsf{X}_{\mathsf{ota}}} \otimes \mathrm{M}^{R^{f_1}_\alpha}_{\mathsf{D}_{q_1}\mathsf{Y}}$$

to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}\mathsf{D}_{q_1}$[15]. Note that $(\mathsf{Ext}_{f_1})^\dagger = \mathsf{Ext}_{f_1}$, thus the secret oracle O_{ota} in game $\mathbf{G_2^q}$ is simulated by $\mathrm{U}^2_{\mathsf{ota}} := \mathsf{Ext}_{f_1} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \mathsf{Ext}_{f_1}$ and $\mathrm{U}^{2,*}_{\mathsf{ota}} := \mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}^2_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})$ before and after $\mathsf{OHG.B}$ is defined, respectively.

[15] Note that the codomain of function f_1 is the union of $\mathcal{X}$ and $\perp$. However, we ignore the extraction with input $\perp$ in Ext_{f_1}, which is different with its definition as shown in Definition 3. That is to say, we restrict the adversary $\mathcal{B}$ from querying secret oracle by $\perp$ in our proof. Indeed, this is reasonable since $\perp$ just an abort symbol and $\perp \notin \mathcal{X}$.

For a computational basis state $|\alpha, 0^{m'}, D\rangle$ on registers $\mathsf{X}_{\mathsf{ota}}\mathsf{YD}_{q_1}$, we have

$$\mathsf{Ext}_{f_1}|\alpha, 0^{m'}, D\rangle = |\alpha, \beta, D\rangle,$$

where β is the smallest value that satisfies $(\beta, D(\beta)) \in R_\alpha^{f_1}$, by the definition of relation $R_\alpha^{f_1}$ in Eq. (11), this means that $\mathsf{ota}_1(\mathsf{sk}, \alpha) = \beta$, $D(\beta) \neq \perp$ and $\mathsf{ota}_2(\mathsf{pk}, \beta, D(\beta)) = \alpha$. If such β does not exist, we have $\mathsf{Ext}_{f_1}|\alpha, 0^{m'}, D\rangle = |\alpha, \perp, D\rangle$.

Intuitively, since $\mathsf{ota}_2(\mathsf{pk}, \beta, D(\beta)) = \alpha$ is equivalent with $D(\beta) \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$, the check in the simulation of secret oracle O_{ota} in game $\mathbf{G_1^c}$ in the proof of Lemma 3 is quantum implemented by $\mathsf{eCO.E}_{f_1}$ except that the classical list is replaced with the database. Thus, the simulation of secret oracle O_{ota} in game $\mathbf{G_2^q}$ can be viewed as a quantum counterpart of the simulation of secret oracle O_{ota} in game $\mathbf{G_1^c}$ in the proof of Lemma 3.

Different with the proof of Lemma 3, which uses some classical events to analysis the difference between the simulation of secret oracle O_{ota} of game $\mathbf{G_0^c}$ and game $\mathbf{G_1^c}$, we actually use some special projectors to analysis the difference between $\mathrm{U}_{\mathsf{ota}}^1$ and $\mathrm{U}_{\mathsf{ota}}^2$. Roughly speaking, we divide the internal state of game $\mathbf{G_1^q}$ and game $\mathbf{G_2^q}$ into some different parts by the projector and then consider the difference for each of these parts after once application of $\mathrm{U}_{\mathsf{ota}}^1$ and $\mathrm{U}_{\mathsf{ota}}^2$. We next introduce the following lemma, that is detailed proved in Appendix F.1.

Lemma 4. $|\Pr[1 \leftarrow \mathbf{G_1^q}] - \Pr[1 \leftarrow \mathbf{G_2^q}]| \leq 8q_{\mathsf{ota}} \cdot \sqrt{\mathsf{ota.max}}$.

<u>Game $\mathbf{G_3^q}$</u>: This game is the same as game $\mathbf{G_2^q}$, except that the extraction-interface $\mathsf{eCO.E}_{f_1}$ is replaced into $\mathsf{eCO.E}_{f_2}$, where function $f_2 : \{0,1\}^m \times \{0,1\}^n \to \mathcal{X}$ is $f_2(x, y) = \mathsf{ota}_2(\mathsf{pk}, x, y)$.

Similar to $\mathsf{eCO.E}_{f_1}$, a query to $\mathsf{eCO.E}_{f_2}$ with registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}$ is processed by applying unitary operation

$$\mathsf{Ext}_{f_2} := \sum_{\alpha \in \mathcal{X}} |\alpha\rangle\langle\alpha|_{\mathsf{X}_{\mathsf{ota}}} \otimes \mathrm{M}_{\mathsf{D}_{q_1}\mathsf{Y}}^{R_\alpha^{f_2}}$$

to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{YD}_{q_1}$. Then the secret oracle O_{ota} in game $\mathbf{G_3^q}$ is simulated by $\mathrm{U}_{\mathsf{ota}}^3 := \mathsf{Ext}_{f_2} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \mathsf{Ext}_{f_2}$ and $\mathrm{U}_{\mathsf{ota}}^{3,*} := \mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}_{\mathsf{ota}}^3 \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})$ before and after $\mathsf{OHG.B}$ is defined, respectively.

For a computational basis state $|\alpha, 0^{m'}, D\rangle$ on registers $\mathsf{X}_{\mathsf{ota}}\mathsf{YD}_{q_1}$, we have

$$\mathsf{Ext}_{f_2}|\alpha, 0^{m'}, D\rangle = |\alpha, \beta, D\rangle,$$

where β is the smallest value that satisfies $(\beta, D(\beta)) \in R_\alpha^{f_2}$, if such β does not exist, we have $\mathsf{Ext}_{f_2}|\alpha, 0^{m'}, D\rangle = |\alpha, \perp, D\rangle$. By the definition of relation $R_\alpha^{f_2}$ defined in Eq. (11), if $\beta \neq \perp$, it satisfies $D(\beta) \neq \perp$ and $\mathsf{ota}_2(\mathsf{pk}, \beta, D(\beta)) = \alpha$.

Intuitively, the simulation of secret oracle O_{ota} in game $\mathbf{G_2^q}$ first extract the smallest β satisfies $\mathsf{ota}_2(\mathsf{pk}, \beta, D(\beta)) = \alpha$ (or $D(\beta) \in \mathsf{ota.sub}_{\mathsf{pk}}^{\alpha,\beta}$) from the database by using $\mathsf{eCO.E}_{f_2}$, and then compute the output of O_{ota} by using this β. Hence the simulation of secret oracle O_{ota} in game $\mathbf{G_2^q}$ can be viewed as a quantum counterpart of the operation Search used in game $\mathbf{G_2^c}$ of the proof of Lemma 3.

In order to bound the difference between the probability that game $\mathbf{G_2^q}$ and game $\mathbf{G_3^q}$ outputs 1, we need to analyze under what types of database D, Ext_{f_1} and Ext_{f_2} will have different output on input state $|\alpha, 0^{m'}, D\rangle$. Fortunately, by the almost identical[16] analysis from game $\mathbf{G_1^c}$ to game $\mathbf{G_2^c}$ in the proof of Lemma 3, Ext_{f_1} and Ext_{f_2} only have different output on input state $|\alpha, 0^{m'}, D\rangle$ if $D \in S$, where

$$S := \{D \in \mathbf{D}_{q_1} : \exists \alpha, \beta' \ s.t. \ \beta' \neq \mathsf{ota}_1(\mathsf{sk}, \alpha) \wedge \mathsf{ota}_2(\mathsf{pk}, \beta', D(\beta')) = \alpha\}. \tag{34}$$

Thus, we can conclude that $\mathsf{eCO.E}_{f_1}$ and $\mathsf{eCO.E}_{f_2}$ proceed identically for any input state $|\alpha, 0^{m'}, D\rangle$ if $D \notin S$.

Obvious we have $D^\perp \notin S$, then by using the compressed semi-classical O2H with database read queries Theorem 3, we can prove the following lemma, the detailed proof is shown in Appendix F.2.

Lemma 5. $|\Pr[1 \leftarrow \mathbf{G_2^q}] - \Pr[1 \leftarrow \mathbf{G_3^q}]| \leq 8 \cdot \sqrt{q_1(q_1 + 1) \cdot \mathsf{ota.union}} + 64q_1 \cdot \mathsf{ota.union}$.

[16] Indeed, the only difference is that the list $\mathfrak{L}_1$ needs to replaced into the database D.

Game $\mathbf{G_4^q}$: This game is the same as game $\mathbf{G_3^q}$, except that the extraction-interface $\mathsf{eCO.E}_{f_2}$ is implemented by unitary operation $\mathsf{StdDecomp}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{StdDecomp}_{m^*||m_1}$ after the $\mathsf{OHG.B}$ is defined.

In what follows, we abbreviate $\mathsf{StdDecomp}_{m^*||m_1}$ into $\mathsf{S}_{m^*||m_1}$ for convenience. Define

$$\mathrm{U}_{\mathsf{ota}}^4 := \mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1}.$$

Then, in game $\mathbf{G_4^q}$, the secret oracle O_{ota} is simulated by $\mathrm{U}_{\mathsf{ota}}^3$ and $\mathrm{U}_{\mathsf{ota}}^{4,*} := \mathrm{U}_\perp \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}_{\mathsf{ota}}^4 \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})$ before and after $\mathsf{OHG.B}$ is defined, respectively.

For fixed $(\mathsf{pk}, \mathsf{sk})$, the parameter $\Gamma_{R_t^{f_2}}$ related to function f_2 defined in Eq. (11) is

$$\Gamma_{R_t^{f_2}} := \max_{x \in \{0,1\}^{m'}} |\{y \in \{0,1\}^n | \mathsf{ota}_2(\mathsf{pk}, x, y) = t\}| = \max_{x \in \{0,1\}^{m'}} |\mathsf{ota.sub}_{\mathsf{pk}}^{x,t}|.$$

Then by using Lemma 2, we have

$$\|[\mathsf{Ext}_f, \mathsf{S}_{m^*||m_1}]\| \leq 16 \cdot \sqrt{\max_{t \in \mathcal{X}} \Gamma_{R_t^{f_2}} / 2^n} \leq 16 \cdot \sqrt{\max_{x \in \{0,1\}^{m'}, t \in \mathcal{X}} |\mathsf{ota.sub}_{\mathsf{pk}}^{x,t}|}. \tag{35}$$

Notice that $\mathsf{S}_{m^*||m_1} \circ \mathsf{S}_{m^*||m_1} = \mathbf{I}$, thus we can conclude that $\mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1}$ is indistinguishable with Ext_{f_2} except the error shown in (35). Then by a similar proof with Lemma 4, we have

$$\begin{aligned} |\Pr[1 \leftarrow \mathbf{G_3^q}] - \Pr[1 \leftarrow \mathbf{G_4^q}]| &\leq 32 q_{\mathsf{ota}} \cdot \sqrt{\mathop{\mathbb{E}}_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)} \max_{x \in \{0,1\}^{m'}, t \in \mathcal{X}} |\mathsf{ota.sub}_{\mathsf{pk}}^{x,t}|} \\ &\overset{(a)}{=} 32 q_{\mathsf{ota}} \cdot \sqrt{\mathsf{ota.max}}. \end{aligned} \tag{36}$$

Here (a) uses Eq. (15).

Game $\mathbf{G_{4a}^q}$: Let $\mathsf{X}_0/\mathsf{Y}_0$ and $\mathsf{X}_1/\mathsf{Y}_1$ be the adversary's input/output register of random oracle O_0 and O_1, respectively. Initialize register Z to 0. Define H as a constant zero function. This game is the same as game $\mathbf{G_4^q}$, except that H is queried on input/output register X_0/Z (resp. X_1/Z) just before every time the simulation of random oracle O_0 (resp. O_1) on input/output register $\mathsf{X}_0/\mathsf{Y}_0$ (resp. $\mathsf{X}_1/\mathsf{Y}_1$).

Compared with game $\mathbf{G_4^q}$, the change in game $\mathbf{G_{4a}^q}$ is only conceptual, thus

$$\Pr[1 \leftarrow \mathbf{G_4^q}] = \Pr[1 \leftarrow \mathbf{G_{4a}^q}]. \tag{37}$$

Game $\mathbf{G_{4b}^q}$: Define set $S_{m^*} := \{x \in \{0,1\}^{m'} : x = m^*||*\}$. This game is the same as game $\mathbf{G_{4a}^q}$, except that the semi-classical oracle $\mathcal{O}_{S_{m^*}}^{SC}$ is queried on input/output register X_0 (resp. X_1) just before the queries of H on input/output register X_0/Z (resp. X_1/Z).

Indeed, we can rewrite game $\mathbf{G_{4a}^q}$ as a quantum oracle algorithm $\mathcal{B}^H$ with input $z \in \{0,1\}^*$, then game $\mathbf{G_{4b}^q}$ can be rewritten as $\mathcal{B}^{H \setminus S_{m^*}}$ with input $z \in \{0,1\}^*$ correspondingly. By using the semi-classical O2H Lemma 10, we have

$$|\Pr[1 \leftarrow \mathbf{G_{4a}^q}] - \Pr[1 \leftarrow \mathbf{G_{4b}^q}]| \leq \sqrt{(q_0 + q_1 + 1) \cdot \Pr[\mathsf{Find}_{4b}^q]}, \tag{38}$$

where Find_{4b}^q denotes the event that the semi-classical oracle $\mathcal{O}_{S_{m^*}}^{SC}$ in game $\mathbf{G_{4b}^q}$ ever outputs 1.

If Find_{4b}^q does not occur, the input state of O_0 on registers $\mathsf{X}_0/\mathsf{Y}_0$ after the query of $\mathcal{O}_{S_{m^*}}^{SC}$ can be written as $\sum_{x \notin S_{m^*}, y} |x, y\rangle$. Thus, O_0 is not queried with input $x \in S_{m^*}$ by the adversary $\mathcal{A}$ in game $\mathbf{G_{4b}^q}$. That is to say, the $O_0(m^*||m_0)$ used by the challenger to generate $\mathsf{OHG.B}$ is uniformly random in adversary's view.

As for the $O_1(m^*||m_1)$, if Find_{4b}^q does not occur, after $\mathsf{OHG.B}$ is defined, the corresponding state on the database register D_{q_1} can be abbreviated as[17]

$$\sum_{D \in \mathbf{D}_{q_1}, n(D) < q_1} \mathsf{S}_{m^*||m_1} |D \cup (m^*||m_1, O_1(m^*||m_1))\rangle.$$

[17]Here we omit the coefficient and other registers that may entangled with D_{q_1}.

Note that the extraction-interface $\mathsf{eCO.E}_{f_2}$ in game $\mathbf{G^q_{4b}}$ is processed by $\mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1}$ after $\mathsf{OHG.B}$ is defined. By the property that $\mathsf{S}_{m^*||m_1} \circ \mathsf{S}_{m^*||m_1} = \mathbf{I}$ and Ext_{f_2} does not change the database in the computational basis, we can conclude that the internal state on database register D_{q_1} always in the form of $\sum_{D \in \mathbf{D}_{q_1}, n(D)<q_1} \mathsf{S}_{m^*||m_1} |D \cup (m^*||m_1, O_1(m^*||m_1))\rangle$ before and after once application of $\mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1}$. This means that, if Find^q_{4b} does not occur in game $\mathbf{G^q_{4b}}$, the simulation of random oracle O_1 at point $m^*||m_1$ is not disturbed by the invoking of the extraction-interface $\mathsf{eCO.E}_{f_2}$ and the adversary only query O_1 with input state $\sum_{x \notin S_{m^*}} |x\rangle$. Hence the $O_1(m^*||m_1)$ used by the challenger to generate $\mathsf{OHG.B}$ is also uniformly random in adversary's view.

In addition, we can prove the following lemma:

Lemma 6. *For the state $\mathsf{S}_{m^*||m_1}|\alpha, D \cup (m^*||m_1, O_1(m^*||m_1)), 0^{m'}\rangle$ on registers $\mathsf{X_{ota}D}_{q_1}\mathsf{Y}$, if $\alpha \neq \mathsf{ota}_2(\mathsf{pk}, m^*||m_1, O_1(m^*||m_1))$, suppose unitary operation $\mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1}$ acts on*

$$\mathsf{S}_{m^*||m_1}|\alpha, D \cup (m^*||m_1, O_1(m^*||m_1)), 0^{m'}\rangle$$

will return β to register Y and

$$\mathsf{Ext}_{f_2}|\alpha, D, 0^{m'}\rangle = |\alpha, D, \beta'\rangle.$$

Then we have $\beta = \beta'$.

Proof. Since $\mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1}$ acts on state $\mathsf{S}_{m^*||m_1}|\alpha, D \cup (m^*||m_1, O_1(m^*||m_1)), 0^{m'}\rangle$ return β to register Y and $\mathsf{S}_{m^*||m_1} \circ \mathsf{S}_{m^*||m_1} = \mathbf{I}$, we have

$$\begin{aligned}
&\mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1} \circ \mathsf{S}_{m^*||m_1}|\alpha, D \cup (m^*||m_1, O_1(m^*||m_1)), 0^{m'}\rangle \\
=& \mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2}|\alpha, D \cup (m^*||m_1, O_1(m^*||m_1)), 0^{m'}\rangle \\
=& \mathsf{S}_{m^*||m_1}|\alpha, D \cup (m^*||m_1, O_1(m^*||m_1)), \beta\rangle,
\end{aligned}$$

where β is the smallest value that satisfies $\mathsf{ota}_2(\mathsf{pk}, \beta, D(\beta)) = \alpha$. Notice that in above state, $\alpha \neq \mathsf{ota}_2(\mathsf{pk}, m^*||m_1, O_1(m^*||m_1))$, hence the β in above formula can not be $m^*||m_1$.

This means that, even if database $D \cup (m^*||m_1, O_1(m^*||m_1))$ contains more information than D, the return of Ext_{f_2} on input state $|\alpha, D \cup (m^*||m_1, O_1(m^*||m_1)), 0^{m'}\rangle$ is irrelevant to those additional information if $\alpha \neq \mathsf{ota}_2(\mathsf{pk}, m^*||m_1, O_1(m^*||m_1))$. Thus, Ext_{f_2} returns the same value on state $|\alpha, D \cup (m^*||m_1, O_1(m^*||m_1)), 0^{m'}\rangle$ and $|\alpha, D, 0^{m'}\rangle$, i.e., $\beta = \beta'$. ☐

The above lemma implies that in game $\mathbf{G^q_{4b}}$, if the challenger does not query RO-interface $\mathsf{eCO.RO}$ by $m^*||m_1$ to get $O_1(m^*||m_1)$ and uniformly random choose $O_1(m^*||m_1)$ from $\{0,1\}^n$ instead, the operation $\mathsf{S}_{m^*||m_1} \circ \mathsf{Ext}_{f_2} \circ \mathsf{S}_{m^*||m_1}$ used by the extraction-interface $\mathsf{eCO.E}_{f_2}$ after $\mathsf{OHG.B}$ is defined, can be reduced to operation Ext_{f_2} directly.

According to above analysis, we can conclude that game $\mathbf{G^q_{4b}}$ and following game $\mathbf{G^q_{4c}}$ are indistinguishable if the event Find^q_{4b} and Find^q_{4c} do not occur, where Find^q_{4c} denotes the event that the semi-classical oracle $\mathcal{O}^{SC}_{S_{m^*}}$ in game $\mathbf{G^q_{4c}}$ ever outputs 1.

Game $\mathbf{G^q_{4c}}$: This game is the same as game $\mathbf{G^q_{4b}}$, except that the following two changes:

- The $y_0 = O_0(m^*||m_0)$ and $y_1 = O_1(m^*||m_1)$ used to generate $\mathsf{OHG.B}$ is replaced with r_0 and r_1 uniformly sampled from $\{0,1\}^n$ and $\{0,1\}^{n'}$, respectively.
- The unitary operation implements the extraction-interface $\mathsf{eCO.E}_{f_2}$ is changed back to Ext_{f_2}.

This implies that

$$\Pr[\mathsf{Find}^q_{4b}] = \Pr[\mathsf{Find}^q_{4c}],$$
$$\Pr[1 \leftarrow \mathbf{G^q_{4b}} \wedge \neg\mathsf{Find}^q_{4b}] = \Pr[1 \leftarrow \mathbf{G^q_{4c}} \wedge \neg\mathsf{Find}^q_{4b}].$$

Then by the difference lemma of [Sho04],

$$|\Pr[1 \leftarrow \mathbf{G^q_{4b}}] - \Pr[1 \leftarrow \mathbf{G^q_{4c}}]| \leq \Pr[\mathsf{Find}^q_{4c}]. \tag{39}$$

Game $\mathbf{G^q_5}$: This game is the same as game $\mathbf{G^q_{4c}}$ except that the H and semi-classical oracle $\mathcal{O}^{SC}_{S_{m^*}}$ are no longer queried.

Similar with the analysis between game $\mathbf{G^q_{4a}}$ and game $\mathbf{G^q_{4b}}$, we have

$$|\Pr[1 \leftarrow \mathbf{G^q_{4c}}] - \Pr[1 \leftarrow \mathbf{G^q_5}]| \leq \sqrt{(q_0 + q_1) \cdot \Pr[\mathsf{Find}^q_{4c}]}, \tag{40}$$

Game $\mathbf{G^q_6}$: This game is the same game as game $\mathbf{G^q_5}$, except that the following changes:

- The adversary is changed to a new adversary $\mathcal{B}_1$, it does not query any oracles and invokes adversary $\mathcal{B}$ once in a black-box manner (without rewinding) as follows:
  1. After get the public key pk, adversary $\mathcal{B}_1$ chooses a $2q_0$-wise independent function f and implements the extractable RO-simulator $\mathcal{S}(f_2) = \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_2}\}$ with internal database register D_{q_1}.
  2. Adversary $\mathcal{B}_1$ invokes adversary $\mathcal{B}$ to get $\mathsf{OHG.A}$ and send it to the challenger. After get the value $\mathsf{OHG.B}$ computed by the challenger, invoke adversary $\mathcal{B}$ to get $\mathsf{OHG.C}$ and send it to the challenger. The oracle query performed by $\mathcal{B}$ is answer as:
     (a) When the random oracle O_0 is queried by $\mathcal{B}$, $\mathcal{B}_1$ answer it by using the unitary operation $\mathrm{U}_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$.
     (b) When the random oracle O_1 is queried by $\mathcal{B}$, $\mathcal{B}_1$ answer it by using the RO-interface $\mathsf{eCO.RO}$.
     (c) When the secret oracle O_{ota} is queried by $\mathcal{B}$, $\mathcal{B}_1$ answer it by using the $\mathrm{U}^3_{\mathsf{ota}} := \mathsf{Ext}_{f_2} \circ \tilde{\mathrm{U}}_{\mathsf{comp}} \circ \mathsf{Ext}_{f_2}$ and $\mathrm{U}^{3,*}_{\mathsf{ota}} := \mathrm{U}_{\perp} \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}^3_{\mathsf{ota}} \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})$ before and after $\mathsf{OHG.B}$ being defined, respectively.
- The random oracle O_0 and O_1, secret oracle O_{ota} in game $\mathbf{G^q_6}$ is simulated the same as game $\mathbf{G^q_0}$[18], and the value of y_0 (resp. y_1) used to generate $\mathsf{OHG.B}$ in game $\mathbf{G^q_6}$ is replaced with $O_0(m^*||m_0)$ (resp. $O_1(m^*||m_1)$).

Compared with game $\mathbf{G^q_5}$, the change in game $\mathbf{G^q_6}$ is only conceptual, thus

$$\Pr[1 \leftarrow \mathbf{G^q_5}] = \Pr[1 \leftarrow \mathbf{G^q_6}]. \tag{41}$$

Moreover, we observe that game $\mathbf{G^q_6}$ is identical with game $\mathbf{G^q_0}$ except that the adversary is replaced to $\mathcal{B}_1$, then

$$\Pr[1 \leftarrow \mathbf{G^q_6}] = \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{B}_1,\mathcal{C}}(1^\lambda). \tag{42}$$

As for the probability that event Find^q_{4c} occurs, we consider oracle-hiding game $\mathsf{OHG}^{O_0,O_1,O_{\mathsf{ota}}}_{\mathcal{B}_2(1^\lambda),\mathcal{C}_{\mathsf{find}}(1^\lambda)}$ in the QROM with a new challenger $\mathcal{C}_{\mathsf{find}}$ and a new adversary $\mathcal{B}_2$ as follows:

- The challenger $\mathcal{C}_{\mathsf{find}}$ is identical with $\mathcal{C}$ except that $\mathcal{C}_{\mathsf{find}}$ finally output $t = \mathsf{boole}[\mathsf{OHG.C} = m^*]$ as game's output.

- The adversary $\mathcal{B}_2$ is identical with $\mathcal{B}_1$, except that $\mathcal{B}_2$ picks $i \xleftarrow{\$} \{1, \ldots, q_0 + q_1\}$ at everything begins and then measures the query input registers (just before) the i-th random oracle query in the computational basis to get measurement outcome $m'||*$, where m' has the same length as m^*. Then $\mathcal{B}_2$ output $\mathsf{OHG.C} = m'$.

Then by using Lemma 11, we have

$$\Pr[\mathsf{Find}^q_{4c}] \leq 4(q_0 + q_1) \cdot \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{B}_2,\mathcal{C}_{\mathsf{find}}}(1^\lambda). \tag{43}$$

Tracing through the above game sequence from game $\mathbf{G^q_0}$ to game $\mathbf{G^q_6}$, combining Eq. (32), (33) and (36-43), Lemma 4 and Lemma 5, we finally obtain

$$\begin{aligned}\left|\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{B},\mathcal{C}}(1^\lambda) - \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{B}_1,\mathcal{C}}(1^\lambda)\right| &\leq 40q_{\mathsf{ota}} \cdot \sqrt{\mathsf{ota.max}} + 8(q_1 + 1) \cdot \sqrt{\mathsf{ota.union}} + 64q_1 \cdot \mathsf{ota.union} \\ &\quad + 4(q_0 + q_1 + 1) \cdot \sqrt{\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{B}_2,\mathcal{C}_{\mathsf{find}}}(1^\lambda)}.\end{aligned}$$

[18] To avoid confusion, we stress that this O_0, O_1 and O_{ota} are oracles queried in game $\mathbf{G^q_6}$, they are independent with the oracle O_0, O_1 and O_{ota} appeared in the description of adversary $\mathcal{B}_1$.

As for the running time of $\mathcal{B}_1$ and $\mathcal{B}_2$, by their construction, we know that they invoke adversary $\mathcal{B}$ only once and simulate random oracle O_0 by a $2q_0$-wise independent function q_0 times, simulate the random oracle O_1 and secret oracle O_{ota} by the extractable RO-simulator $\mathcal{S}(f_2) = \{\mathsf{eCO.RO}, \mathsf{eCO.E}_{f_2}\}$. The RO-interface $\mathsf{eCO.RO}$ and extraction-interface $\mathsf{eCO.E}_{f_2}$ is invoked q_0+q_1 and $2q_{\mathsf{ota}}$) times, respectively. Hence by the Definition 3, we have

$$\mathrm{Time}[\mathcal{B}_1(1^\lambda)] \approx \mathrm{Time}[\mathcal{B}_2(1^\lambda)] \leq \mathrm{Time}[\mathcal{B}(1^\lambda)] + O((q_0+q_1)\cdot q_{\mathsf{ota}}\cdot \mathsf{ota.time} + (q_0+q_1)^2).$$

The definition of $\mathsf{ota.time}$ is given in Definition 4. □

4.2.3 The Construction of Adversary $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{B}_1$ and $\mathcal{B}_2$

Compared with construction of adversary $\mathcal{A}_1$ in the proof of Lemma 3, the construction of adversary $\mathcal{B}_1$ given in the proof of Theorem 4 only change the simulation of the oracles, we also give an overview in Table 2.

Table 2: The overview of adversary $\mathcal{A}_1$ and $\mathcal{B}_1$.

Adversary	Main procedure	Random oracle O_0	Random oracle O_1	Secret oracle O_{ota}
$\mathcal{A}_1$	invokes $\mathcal{A}$	on-the-fly	on-the-fly	Search
$\mathcal{B}_1$	invokes $\mathcal{B}$	$2q_0$-wise function f	$\mathsf{eCO.RO}$	$\mathsf{eCO.E}_{f_2}\circ \tilde{\mathrm{U}}_{\mathsf{comp}}\circ \mathsf{eCO.E}_{f_2}$

In fact, $\mathcal{B}_1$ can also simulate the random oracle O_0 by the RO interface $\mathsf{eCO.RO}$ of a new extractable RO-simulator, but this will require more quantum resources. Overall, we observe that the operations of adversary $\mathcal{A}_1$ and $\mathcal{B}_1$ are one-to-one corresponding. Their operations both are invoking the underlying adversary and simulating oracles for the underlying adversary. Although the simulation methods of $\mathcal{A}_1$ and $\mathcal{B}_1$ are different, the simulation methods used by $\mathcal{B}_1$ can all be regarded as the quantum counterpart of $\mathcal{A}_1$. This is why we wrote in Theorem 4 that we can directly construct $\mathcal{B}_1$ by mimicking the construction of $\mathcal{A}_1$.

As for the adversary $\mathcal{A}_2$ in the proof of Lemma 3 and the adversary $\mathcal{B}_2$ in the proof of Theorem 4, their operations are also one-to-one corresponding:

- Construction of $\mathcal{A}_2$: Run $\mathcal{A}_1$, picks $i \xleftarrow{\$} \{1,\ldots,q_0+q_1\}$ and record the i-th random oracle query $m'||*$. Then output $\mathsf{OHG.C} = m'$.
- Construction of $\mathcal{B}_2$: Run $\mathcal{B}_1$, picks $i \xleftarrow{\$} \{1,\ldots,q_0+q_1\}$ and measure the i-th random oracle query to get measurement outcome $m'||*$. Then output $\mathsf{OHG.C} = m'$.

As $\mathcal{B}_1$ needs to handle quantum queries, $\mathcal{B}_2$ changed the "record query" used by $\mathcal{A}_2$ to "measure query". Obviously, similar to $\mathcal{A}_1$ and $\mathcal{B}_1$, we can directly construct $\mathcal{B}_2$ by mimicking the construction of $\mathcal{A}_2$.

5 Applications of Theorem 4

In this section, we apply our lifting theorem Theorem 4 to prove the $\mathsf{IND\text{-}qCCA}$ and $\mathsf{ANO\text{-}qCCA}$ security of the FO-like transformation in the QROM. The formal definition of cryptographic primitives and security notions used in this section are shown in Appendix G, along with the definition of correctness and spreadness of PKE schemes. Similar with Section 4.2, we sometimes omit the security parameter λ for notational clarity. Moreover, we only consider QPT adversary in this section.

To a a PKE scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\{0,1\}^u$ and randomness space $\{0,1\}^v$, and random oracles $H : \{0,1\}^u \to \{0,1\}^v$, $G : \{0,1\}^* \to \{0,1\}^k$ and a pseudorandom function (PRF) f with key space $\mathcal{K}^{prf}$ we associate

$$\begin{aligned}
\mathsf{KEM}_m^{\perp} &= \mathsf{FO}_m^{\perp}[\mathsf{PKE}, H, G] = (\mathsf{Gen}, \mathsf{Encaps}_m, \mathsf{Decaps}_m^{\perp}),\\
\mathsf{KEM}^{\perp} &= \mathsf{FO}^{\perp}[\mathsf{PKE}, H, G] = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps}^{\perp}),\\
\mathsf{KEM}_m^{\not\perp} &= \mathsf{FO}_m^{\not\perp}[\mathsf{PKE}, H, G] = (\mathsf{Gen}_m^{\not\perp}, \mathsf{Encaps}, \mathsf{Decaps}_m^{\not\perp}),\\
\mathsf{KEM}^{\not\perp} &= \mathsf{FO}^{\not\perp}[\mathsf{PKE}, H, G] = (\mathsf{Gen}^{\not\perp}, \mathsf{Encaps}, \mathsf{Decaps}^{\not\perp}).
\end{aligned}$$

Their constituting algorithms are shown in Fig. 5.

$\underline{\mathsf{Gen}^{\not\perp}}$ $\boxed{\mathsf{Gen}_m^{\not\perp}}$

1: $(pk, sk) \leftarrow \mathsf{Gen}$

2: $s \xleftarrow{\$} \{0,1\}^u$ $\boxed{s \xleftarrow{\$} \mathcal{K}^{prf}}$

3: $sk' := sk||s$

4: **Return** (pk, sk)

$\underline{\mathsf{Encaps}\,(pk)}$ $\boxed{\mathsf{Encaps}_m\,(pk)}$

1: $m \xleftarrow{\$} \{0,1\}^u$

2: $c := \mathsf{Enc}\,(pk, m; H(m))$

3: $K := G(m, c)$ $\boxed{K := G(m)}$

4: return (K, c)

$\underline{\mathsf{Decaps}^{\perp}\,(sk, c)}$ $\boxed{\mathsf{Decaps}_m^{\perp}\,(sk, c)}$

1: $m' := \mathsf{Dec}\,(sk, c)$

2: **If** $c \neq \mathsf{Enc}\,(pk, m'; H(m'))$ **or** $m' = \perp$

  return $\perp$

3: **else return** $K := G(m', c)$ $\boxed{K := G(m')}$

$\underline{\mathsf{Decaps}^{\not\perp}\,(sk' = sk||s, c)}$ $\boxed{\mathsf{Decaps}_m^{\not\perp}\,(sk' = sk||s, c)}$

1: $m' := \mathsf{Dec}\,(sk, c)$

2: **If** $c \neq \mathsf{Enc}\,(pk, m'; H(m'))$ **or** $m' = \perp$

  return $K := G(s, c)$ $\boxed{K := f(s, c)}$

3: **else return** $K := G(m', c)$ $\boxed{K := G(m')}$

Figure 5: KEM scheme $\mathsf{KEM}_m^{\perp} = (\mathsf{Gen}, \mathsf{Encaps}_m, \mathsf{Decaps}_m^{\perp})$, $\mathsf{KEM}^{\perp} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps}^{\perp})$, $\mathsf{KEM}_m^{\not\perp} = (\mathsf{Gen}_m^{\not\perp}, \mathsf{Encaps}, \mathsf{Decaps}_m^{\not\perp})$ and $\mathsf{KEM}^{\not\perp} = (\mathsf{Gen}^{\not\perp}, \mathsf{Encaps}, \mathsf{Decaps}^{\not\perp})$.

To a DEM scheme DEM=(E,D) with key space $\{0,1\}^k$, we associate

$$\mathsf{PKE}_m^{\perp} = \mathsf{KEM}_m^{\perp} + \mathsf{DEM} = (\mathsf{Gen}, \mathsf{Enc}_m, \mathsf{Dec}_m^{\perp}),$$
$$\mathsf{PKE}^{\perp} = \mathsf{KEM}^{\perp} + \mathsf{DEM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}^{\perp}),$$
$$\mathsf{PKE}_m^{\not\perp} = \mathsf{KEM}_m^{\not\perp} + \mathsf{DEM} = (\mathsf{Gen}_m^{\not\perp}, \mathsf{Enc}, \mathsf{Dec}_m^{\not\perp}),$$
$$\mathsf{PKE}^{\not\perp} = \mathsf{KEM}^{\not\perp} + \mathsf{DEM} = (\mathsf{Gen}^{\not\perp}, \mathsf{Enc}, \mathsf{Dec}^{\not\perp}).$$

Their constituting algorithms are shown in Fig. 6. Here "A+B" refer to a PKE scheme built via the KEM-DEM paradigm with KEM scheme A and DEM scheme B.

$\underline{\mathsf{Gen}^{\not\perp}}$ $\boxed{\mathsf{Gen}_m^{\not\perp}}$

1: $(pk, sk) \leftarrow \mathsf{Gen}$

2: $s \xleftarrow{\$} \{0,1\}^u$ $\boxed{s \xleftarrow{\$} \mathcal{K}^{prf}}$

3: $sk' := sk||s$

4: **Return** (pk, sk)

$\underline{\mathsf{Enc}\,(pk, m)}$ $\boxed{\mathsf{Enc}_m\,(pk, m)}$

1: $\delta \xleftarrow{\$} \{0,1\}^u$

2: $c_1 := \mathsf{Enc}\,(pk, \delta; H(\delta))$

3: $K := G(\delta, c_1)$ $\boxed{K := G(\delta)}$

4: $c_2 := \mathsf{E}(K, m)$

5: return (c_1, c_2)

$\underline{\mathsf{Dec}^{\perp}\,(sk, c_1, c_2)}$ $\boxed{\mathsf{Dec}_m^{\perp}\,(sk, c_1, c_2)}$

1: $\delta' := \mathsf{Dec}\,(sk, c_1)$

2: **If** $c_1 \neq \mathsf{Enc}\,(pk, \delta'; H(\delta'))$ **or** $\delta' = \perp$

  return $\perp$

3: **else compute** $K := G(\delta', c_1)$ $\boxed{K := G(\delta')}$

  return $m' := \mathsf{D}(K, c_2)$

$\underline{\mathsf{Dec}^{\not\perp}\,(sk' = sk||s, c_1, c_2)}$ $\boxed{\mathsf{Dec}_m^{\not\perp}\,(sk' = sk||s, c_1, c_2)}$

1: $\delta' := \mathsf{Dec}\,(sk, c_1)$

2: **If** $c_1 \neq \mathsf{Enc}\,(pk, \delta'; H(\delta'))$ **or** $\delta' = \perp$

  compute $K := G(s, c_1)$ $\boxed{K := f(s, c_1)}$

  return $m' := \mathsf{D}(K, c_2)$

3: **else compute** $K := G(\delta', c)$ $\boxed{K := G(\delta')}$

  return $m' := \mathsf{D}(K, c_2)$

Figure 6: PKE scheme $\mathsf{PKE}_m^{\perp} = (\mathsf{Gen}, \mathsf{Enc}_m, \mathsf{Dec}_m^{\perp})$, $\mathsf{PKE}^{\perp} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}^{\perp})$, $\mathsf{PKE}_m^{\not\perp} = (\mathsf{Gen}^{\not\perp}, \mathsf{Enc}, \mathsf{Dec}_m^{\not\perp})$ and $\mathsf{PKE}^{\not\perp} = (\mathsf{Gen}^{\not\perp}, \mathsf{Enc}, \mathsf{Dec}^{\not\perp})$.

Before we giving the ANO-qCCA security reduction, we introduce a theorem indicates that weak pseudorandomness of PKE immediately implies anonymity of PKE. The detailed proof of this theorem is similar to the proof of Theorem 2.5 in [Xag22] and we present it in Appendix H.1.

Theorem 5. *Denote* Π *as a PKE scheme,* $\mathcal{S}$ *as a QPT simulator of the* WPR-qCCA *game of* Π*, then for any adversary* $\mathcal{A}$ *against the* ANO-qCCA *game of* Π*, there exists adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{ANO\text{-}qCCA}}_{\mathcal{A},\Pi} \leq 2 \cdot \mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{B},\mathcal{S},\Pi}$$

and $\mathrm{Time}[\mathcal{B}] \approx \mathrm{Time}[\mathcal{A}]$.

5.1 The IND-qCCA security of $\mathsf{KEM}^{\perp}_m$, $\mathsf{KEM}^{\perp}$, $\mathsf{KEM}^{\not\perp}_m$ and $\mathsf{KEM}^{\not\perp}$ in the QROM

Here we only provide the IND-qCCA security reduction of $\mathsf{KEM}^{\perp}_m$ in the QROM, the reduction of $\mathsf{KEM}^{\perp}$, $\mathsf{KEM}^{\not\perp}_m$ and $\mathsf{KEM}^{\not\perp}$ can be obtained in a similar way and they are presented in Appendix H.2.

Theorem 6. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct and weakly* γ*-spread. Let* $\mathcal{A}$ *be an* IND-qCCA *adversary against* $\mathsf{KEM}^{\perp}_m$ *in the QROM, making at most* q_H*,* q_G *and* q_D *queries to random oracle* H*,* G *and the decryption oracle, respectively. Then there exists an* OW-CPA *adversary* $\mathcal{A}_1$ *against* PKE *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}^{\perp}_m} \leq 40q_D \cdot \sqrt{\gamma} + 8(q_H+1) \cdot \sqrt{\delta} + 64q_H \cdot \delta + 4(q_H+q_G+1) \cdot \sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}}}.$$

The running time of adversary $\mathcal{A}_1$ *can be bounded by*

$$\mathrm{Time}[\mathcal{A}_1] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_C \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Proof. The IND-qCCA game $\mathbf{G}_{\mathcal{A}}$ of $\mathsf{KEM}^{\perp}_m$ with adversary $\mathcal{A}$ in the QROM is shown in Fig. 7. Then we have

$$\mathsf{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}^{\perp}_m} = \left|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}] - \frac{1}{2}\right|. \tag{44}$$

Game $\mathbf{G}_{\mathcal{A}}$	
1, $(pk, sk) \leftarrow \mathsf{Gen}$	
2, $b \xleftarrow{\$} \{0,1\}$, $m^* \xleftarrow{\$} \{0,1\}^u$	$\mathsf{Deca}(c)$
$c^* = \mathsf{Enc}(pk, m^*, H(m^*))$	1, **If** $c = c^*$, **return** $\perp$
$K_0^* = G(m^*)$, $K_1^* \xleftarrow{\$} \{0,1\}^k$	**Else return** $\mathsf{Deca}^{\perp}_m(sk, c)$
3, $b' \leftarrow \mathcal{A}^{H,G,\mathsf{Deca}}(pk, c^*, K_b^*)$	
4, **Return** $\mathsf{boole}[b = b']$	

Figure 7: Game $\mathbf{G}_{\mathcal{A}}$ with adversary $\mathcal{A}$ in the QROM. Here $\{0,1\}^k$ is the key space of $\mathsf{KEM}^{\perp}_m$, $\mathcal{A}$ can query random oracle H, G and the decapsulation oracle Deca in superposition.

Define $\mathsf{f}_{\mathsf{dec}}$ be a function that $\mathsf{f}_{\mathsf{dec}}(x) = \perp$ for any x. We first rewrite the decapsulation algorithm $\mathsf{Deca}^{\perp}_m(sk, \cdot)$ shown in Fig. 5 as a new oracle algorithm $\mathsf{dec}^{G,H}(sk, \cdot)$ as follows.

1. For input c, compute $m := \mathsf{Dec}(sk, c)$. If $m = \perp$, return $\mathsf{f}_{\mathsf{dec}}(c)$.
2. Else, compute $\mathsf{Enc}(pk, m, H(m))$. If $\mathsf{Enc}(pk, m, H(m)) \neq c$, return $\mathsf{f}_{\mathsf{dec}}(c)$.
   (a) Else, compute $m' := \mathsf{dec}_1(pk, c, m)$ and return $\mathsf{dec}_2(pk, c, m, G(m'))$.
   - $\mathsf{dec}_1(pk, \cdot)$ is a deterministic algorithm that returns y for input (x, y).
   - $\mathsf{dec}_2(pk, \cdot)$ is a deterministic algorithm that returns x for input (x, y, z).

Indeed, oracle algorithm $\mathsf{dec}^{G,H}(sk, \cdot)$ can be regarded as an oracle-testing algorithm. More detailed, in Table 3, we provide the correspondence between the basic components, e.g. the internal algorithms, of oracle algorithm $\mathsf{dec}^{G,H}(sk, \cdot)$ and oracle-testing algorithm $\mathsf{ota}^{O_0,O_1}(\mathsf{sk}, \cdot)$ introduced in Definition 4.

Table 3: The correspondence between the basic components of algorithm $\mathsf{dec}^{G,H}(sk,\cdot)$ and oracle-testing algorithm $\mathsf{ota}^{O_0,O_1}(\mathsf{sk},\cdot)$.

	Key generator	Random oracle	function	Internal algorithms
$\mathsf{ota}^{O_0,O_1}(\mathsf{sk},\cdot)$	$(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KGen}$	O_0/O_1	$\mathsf{f_{ota}}$	$\mathsf{ota}_1(\mathsf{sk},\cdot)/\mathsf{ota}_2(\mathsf{pk},\cdot)/\mathsf{ota}_3(\mathsf{pk},\cdot)/\mathsf{ota}_4(\mathsf{pk},\cdot)$
$\mathsf{dec}^{H,G}(sk,\cdot)$	$(pk,sk)\leftarrow\mathsf{Gen}$	G/H	$\mathsf{f_{dec}}$	$\mathsf{Dec}(sk,\cdot)/\mathsf{Enc}(pk,\cdot)/\mathsf{dec}_1(pk,\cdot)/\mathsf{dec}_2(pk,\cdot)$

As for the corresponding parameter $\mathsf{dec.time}$, $\mathsf{dec.max}$ and $\mathsf{dec.union}$ defined in Eq. (15), by the δ-correctness and weakly γ-spreadness of PKE and their definitions in Appendix G, the following inequalities are obtained:

$$\mathsf{dec.time}\approx\mathrm{Time}[\mathsf{Enc}],\ \mathsf{dec.max}\le\gamma,\ \mathsf{dec.union}\le\delta. \tag{45}$$

Based on the oracle-testing algorithm $\mathsf{dec}^{G,H}(sk,\cdot)$, we design an oracle-hiding game $\mathsf{OHG}^{G,H,O_{\mathsf{dec}}}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}$ in the QROM as shown in Fig. 8, where $\mathcal{A}_{\mathsf{dec}}$ and $\mathcal{C}_{\mathsf{dec}}$ satisfy the following properties:

- Without any computations, $\mathcal{A}_{\mathsf{dec}}$ generates $\mathsf{OHG.A}$ as $\perp$ directly.
- $\mathsf{cha}_1(pk,\cdot)$ and $\mathsf{cha}_2(pk,\cdot)$, performed by $\mathcal{C}_{\mathsf{dec}}$, both return $\varnothing$ for any input, where $\varnothing$ satisfies $x||\varnothing:=x$ for any x.
- $\mathsf{cha}_3(pk,\cdot)$, performed by $\mathcal{C}_{\mathsf{dec}}$, generates $\mathsf{OHG.B}$ as $(\mathsf{Enc}(pk,m^*,y_1),y_0)$ (resp. $(\mathsf{Enc}(pk,m^*,y_1),K)$) for input $(\mathsf{OHG.A},y_0,y_1,m^*,(b,K))$ if $b=0$ (resp. $b=1$).
- $\mathcal{A}_{\mathsf{dec}}$ just runs $\mathcal{A}$ of game $\mathbf{G}_{\mathcal{A}}$[19], and returns the output b' of $\mathcal{A}$ as $\mathsf{OHG.C}$.
- The algorithm $\mathsf{verify}(pk,sk,\cdot)$, performed by $\mathcal{C}_{\mathsf{dec}}$, returns $t=\mathsf{boole}[b=\mathsf{OHG.C}]$ for input $(\mathsf{OHG.A},m^*,(b,K),s,\mathsf{OHG.C})$ directly.

<u>Oracle-hiding game $\mathsf{OHG}^{G,H,O_{\mathsf{dec}}}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}$</u>

1, $(pk,sk)\leftarrow\mathsf{Gen}$

2, $(\mathsf{OHG.A}=\perp)\leftarrow\mathcal{A}^{G,H,O_{\mathsf{dec}}}_{\mathsf{dec}}(pk)$

3, $\mathcal{C}_{\mathsf{dec}}$ **perform following operation**

$m^*\xleftarrow{\$}\mathcal{M}$, $(b,K)\xleftarrow{\$}\{0,1\}\times\mathcal{K}$, $s=0$

$\varnothing\leftarrow\mathsf{cha}_1(pk,\mathsf{OHG.A},m^*,(b,K))$

$y_0=G(m^*)$

$\varnothing\leftarrow\mathsf{cha}_2(pk,\mathsf{OHG.A},y_0,m^*,(b,K))$

$y_1=H(m^*)$

$\mathsf{OHG.B}\leftarrow\mathsf{cha}_3(pk,\mathsf{OHG.A},y_0,y_1,m^*,(b,K))$

$(\mathsf{OHG.B}=(\mathsf{Enc}(pk,m^*,y_1),y_0)$ if $b=0)$

$(\mathsf{OHG.B}=(\mathsf{Enc}(pk,m^*,y_1),K)$ if $b=1)$

4, $(\mathsf{OHG.C}=b')\leftarrow\mathcal{A}^{G,H,O_{\mathsf{dec}}}_{\mathsf{dec}}(pk,\mathsf{OHG.B})$

5, $t\leftarrow\mathsf{verify}(pk,sk,\mathsf{OHG.A},m^*,(b,K),s,\mathsf{OHG.C})$

$(t=\mathsf{booole}[b=\mathsf{OHG.C}])$

$\mathcal{C}_{\mathsf{dec}}$ **output** $t\in\{0,1\}$ **as game's output**

<u>$G(x)$</u>

1, $O\xleftarrow{\$}\mathcal{F}_{*,k}$, **return** $O(x)$

<u>$H(x)$</u>

1, $O'\xleftarrow{\$}\mathcal{F}_{u,v}$, **return** $O'(x)$

<u>$O_{\mathsf{dec}}(c)$</u>

1, **If** $\mathsf{OHG.B}$ **is defined and**

$c=\mathsf{Enc}(pk,m^*,H(m^*))$

return $\perp$

Else return $\mathsf{dec}^{G,H}(sk,c)$

Figure 8: The oracle-hiding game $\mathsf{OHG}^{G,H,O_{\mathsf{dec}}}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}$ in the QROM.

[19]When the random oracle H, G and the decapsulation oracle Deca is queried by $\mathcal{A}$, $\mathcal{A}_{\mathsf{dec}}$ answers it by querying H, G and secret oracle O_{dec}, respectively. Note that the test performed by O_{dec} is exactly the check that $c=c^*$. Hence $\mathcal{A}_{\mathsf{dec}}$ simulates $\mathcal{A}$'s view in the game $\mathbf{G}_{\mathcal{A}}$ perfectly.

Obviously, the running time of $\mathcal{A}_{\mathsf{dec}}$ and that of $\mathcal{A}$ are almost the same. And it is concluded that the final output of game $\mathbf{G}_{\mathcal{A}}$ and oracle-hiding game $\mathsf{OHG}^{G,H,O_{\mathsf{dec}}}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}$ must be the same. Because these two games actually perform the same computations, even though their symbolic representations are different. Hence, we have

$$\mathrm{Time}[\mathcal{A}_{\mathsf{dec}}] \approx \mathrm{Time}[\mathcal{A}],\ \Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}] = \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}(1^{\lambda}). \tag{46}$$

By the properties of $\mathcal{A}_{\mathsf{dec}}$ given above, we know the query numbers of random oracle H, G and secret oracle O_{dec} in the oracle-hiding game $\mathsf{OHG}^{G,H,O_{\mathsf{dec}}}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}$ is q_H, q_G and q_D, respectively. Then by using Theorem 4 and Eq. (45), there exist adversary $\mathcal{A}^1_{\mathsf{dec}}$ and $\mathcal{A}^2_{\mathsf{dec}}$, making no queries to any oracle, satisfying that

$$\begin{aligned}\left|\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}(1^{\lambda}) - \mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}^1_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}(1^{\lambda})\right| &\le 40q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta \\ &\quad + 4(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}^2_{\mathsf{dec}},\mathcal{C}^{\mathsf{find}}_{\mathsf{dec}}}(1^{\lambda})},\end{aligned} \tag{47}$$

and

$$\mathrm{Time}[\mathcal{A}^1_{\mathsf{dec}}] \approx \mathrm{Time}[\mathcal{A}^2_{\mathsf{dec}}] \le \mathrm{Time}[\mathcal{A}_{\mathsf{dec}}] + O(q_H\cdot q_C\cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2), \tag{48}$$

where challenger $\mathcal{C}^{\mathsf{find}}_{\mathsf{dec}}$ is identical with $\mathcal{C}_{\mathsf{dec}}$, except that algorithm verify used by $\mathcal{C}^{\mathsf{find}}_{\mathsf{dec}}$ outputs $t = \mathsf{boole}[\mathsf{OHG.C} = m^*]$ for the input $(\mathsf{OHG.A}, m^*, (b,K), s, \mathsf{OHG.C})$.

Regarding $\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}^1_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}(1^{\lambda})$ and $\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}^2_{\mathsf{dec}},\mathcal{C}^{\mathsf{find}}_{\mathsf{dec}}}(1^{\lambda})$, it is noted that $\mathcal{A}^1_{\mathsf{dec}}$ and $\mathcal{A}^2_{\mathsf{dec}}$ makes no queries to any oracle. Therefore, the value y_0 and y_1, which are shown in Fig. 8 and generated by challenger $\mathcal{C}_{\mathsf{dec}}$ and $\mathcal{C}^{\mathsf{find}}_{\mathsf{dec}}$, are uniformly random in the view of $\mathcal{A}^1_{\mathsf{dec}}$ and $\mathcal{A}^2_{\mathsf{dec}}$ in oracle-hiding game $\mathsf{OHG}^{G,H,O_{\mathsf{dec}}}_{\mathcal{A}^1_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}$ and $\mathsf{OHG}^{G,H,O_{\mathsf{dec}}}_{\mathcal{A}^2_{\mathsf{dec}},\mathcal{C}^{\mathsf{find}}_{\mathsf{dec}}}$, respectively. Hence, it can be concluded that the bit b chosen by challenger $\mathcal{C}_{\mathsf{dec}}$ in oracle-hiding game $\mathsf{OHG}^{G,H,O_{\mathsf{dec}}}_{\mathcal{A}^1_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}$ is independent from $\mathcal{A}^1_{\mathsf{dec}}$'s view. Then we have

$$\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}^1_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}(1^{\lambda}) = \frac{1}{2}. \tag{49}$$

Moreover, it is concluded that there exists adversary $\mathcal{A}_1$ against the $\mathsf{OW\text{-}CPA}$ security of the underlying PKE such that

$$\mathsf{Adv}^{\mathsf{OHG}}_{\mathcal{A}^2_{\mathsf{dec}},\mathcal{C}^{\mathsf{find}}_{\mathsf{dec}}}(1^{\lambda}) = \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}},\ \mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}^2_{\mathsf{dec}}]. \tag{50}$$

Combining Eq. (44) and Eq. (46) to (50), we finally obtain

$$\mathsf{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}^{\perp}_m} \le 40q_D\cdot\sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 4(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}}},$$

and

$$\mathrm{Time}[\mathcal{A}_1] \le \mathrm{Time}[\mathcal{A}] + O(q_H\cdot q_C\cdot\mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

□

5.2 The $\mathsf{ANO\text{-}qCCA}$ security of $\mathsf{KEM}^{\perp}_m$, $\mathsf{KEM}^{\perp}$, $\mathsf{KEM}^{\not\perp}_m$ and $\mathsf{KEM}^{\not\perp}$ in the QROM

We first prove the $\mathsf{SPR\text{-}qCCA}$ security of KEM scheme $\mathsf{KEM}^{\perp}_m$ in the QROM.

Theorem 7. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, weakly γ-spread and* $\mathsf{SDS\text{-}IND}$*-secure w.r.t. QPT simulator $\mathcal{S}$. Let $\mathcal{A}$ be a* $\mathsf{SPR\text{-}qCCA}$ *adversary against* $\mathsf{KEM}^{\perp}_m$ *in the QROM, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively*[20]*. Then there exist an* $\mathsf{OW\text{-}CPA}$ *adversary $\mathcal{A}_1$ against the* PKE *and a* $\mathsf{SDS\text{-}IND}$ *adversary $\mathcal{A}_2$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{SPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S},\mathsf{KEM}^{\perp}_m} \le 24q_D\cdot\sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 2(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}}} + \mathsf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}.$$

The running time of adversary $\mathcal{A}_1$ and $\mathcal{A}_2$ can be bounded as

[20] Following [JZC+18, GMP22], we make the convention that q_H and q_G counts the total number of times H and G is queried in the $\mathsf{SPR\text{-}qCCA}$ game, respectively.

$$\mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}_2] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Proof. Based on the $\mathsf{SPR\text{-}qCCA}$ game of $\mathsf{KEM}_m^{\perp}$ with adversary $\mathcal{A}$ and simulator $\mathcal{S}$, define game $\mathbf{G}_{\mathcal{A}}^{b=0}$ and game $\mathbf{G}_{\mathcal{A}}^{b=1}$ as shown in Fig. 9, then we have

$$|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}]| = 2 \cdot \mathsf{Adv}_{\mathcal{A},\mathcal{S},\mathsf{KEM}_m^{\perp}}^{\mathsf{SPR\text{-}qCCA}}. \tag{51}$$

Game $\mathbf{G}_{\mathcal{A}}^{b=0}$

1: $(pk, sk) \leftarrow \mathsf{Gen}$, $b = 0$
2: $m^* \xleftarrow{\$} \{0,1\}^u$
   $c_0^* := \mathsf{Enc}(pk, m^*, H(m^*))$
   $K_0^* := G(m^*)$
3: $b' \leftarrow \mathcal{A}^{H,G,O_{\mathsf{dec}}^{c_0^*}}(pk, c_0^*, K_0^*)$
4: **Return** b'

$O_{\mathsf{dec}}^{c_0^*}(c)$

1: **If** $c = c_0^*$, **return** $\perp$
   Else return $\mathsf{Deca}_m^{\perp}(c)$

$O_{\mathsf{dec}}^{c_1^*}(c)$

1: **If** $c = c_1^*$, **return** $\perp$
   Else return $\mathsf{Deca}_m^{\perp}(c)$

Game $\mathbf{G}_{\mathcal{A}}^{b=1}$

1: $(pk, sk) \leftarrow \mathsf{Gen}$, $b = 1$
2: $m^* \xleftarrow{\$} \{0,1\}^u$
   $c_1^* := \mathcal{S}(1^\lambda)$
   $K_1^* \xleftarrow{\$} \{0,1\}^k$
3: $b' \leftarrow \mathcal{A}^{H,G,O_{\mathsf{dec}}^{c_1^*}}(pk, c_1^*, K_1^*)$
4: **Return** b'

Figure 9: Game $\mathbf{G}_{\mathcal{A}}^{b=0}$ and game $\mathbf{G}_{\mathcal{A}}^{b=1}$. Here adversary $\mathcal{A}$ can query its oracles in superposition.

By using lifting theorem Theorem 4, we can prove following lemma, its detailed proof is shown in Appendix H.3

Lemma 7. *There exists adversary $\mathcal{B}$ and $\mathcal{A}_1$ without query any oracles it can access such that*

$$|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}]| \leq 40q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H \cdot \delta + 4(q_H + q_G + 1)\cdot\sqrt{\mathsf{Adv}_{\mathcal{A}_1,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}},$$

and

$$|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]| \leq 8q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H \cdot \delta.$$

The running time of adversary $\mathcal{B}$ and $\mathcal{A}_1$ can be bounded as

$$\mathrm{Time}[\mathcal{B}] \approx \mathrm{Time}[\mathcal{A}_1] \leq \mathrm{Time}[\mathcal{A}] + O((q_G + q_H)\cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + (q_G+q_H)^2).$$

Notice that the adversary $\mathcal{B}$ in Lemma 7 does not query any oracles it can access, hence in game $\mathbf{G}_{\mathcal{B}}^{b=0}$ and game $\mathbf{G}_{\mathcal{B}}^{b=1}$, the K_0^* and K_1^* both are uniformly random in adversary $\mathcal{B}$'s view. It is easy to obtain that there exist an adversary $\mathcal{A}_2$ against the $\mathsf{SDS\text{-}IND}$ security of PKE that satisfying $\mathrm{Time}[\mathcal{A}_2] \approx \mathrm{Time}[\mathcal{B}]$ and

$$|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]| = 2 \cdot \mathsf{Adv}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}^{\mathsf{SDS\text{-}IND}}. \tag{52}$$

Thus by using the upper bound given in Lemma 7, we have

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A},\mathcal{S},\mathsf{KEM}_m^{\perp}}^{\mathsf{SPR\text{-}qCCA}} &= |\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}]|/2 \\
&\leq |\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}]|/2 + |\Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]|/2 \\
&\quad + |\Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}]|/2 \\
&\leq 24q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 2(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}_{\mathcal{A}_1,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}} + \mathsf{Adv}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}^{\mathsf{SDS\text{-}IND}}.
\end{aligned}$$

□

Corollary 1. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* $\mathsf{OW\text{-}CPA}$*-secure and* $\mathsf{SDS\text{-}IND}$*-secure, then* $\mathsf{KEM}_m^{\perp}$ *is* $\mathsf{ANO\text{-}qCCA}$*-secure in the QROM.*

This follows from the Theorem 2.5 of [Xag22], which indicates that the $\mathsf{SPR\text{-}qCCA}$ security of KEM schemes implies its $\mathsf{ANO\text{-}qCCA}$ security[21]. Similar with the proof of Theorem 7, we can also prove the $\mathsf{SPR\text{-}qCCA}$ security of KEM scheme $\mathsf{KEM}^{\perp}$, $\mathsf{KEM}_m^{\not\perp}$ and $\mathsf{KEM}^{\not\perp}$ in the QROM. We give these proofs in Appendix H.4. Then by using the Theorem 2.5 of [Xag22] again, we obtain following corollary:

Corollary 2. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* $\mathsf{OW\text{-}CPA}$*-secure and* $\mathsf{SDS\text{-}IND}$*-secure, then* $\mathsf{KEM}^{\perp}$, $\mathsf{KEM}_m^{\not\perp}$ *and* $\mathsf{KEM}^{\not\perp}$ *is* $\mathsf{ANO\text{-}qCCA}$*-secure in the QROM.*

[21] Note that the Theorem 2.5 of [Xag22] actually states that the $\mathsf{SPR\text{-}CCA}$ security of KEM schemes implies its $\mathsf{ANO\text{-}CCA}$ security. Although their proof is not specific to the "qCCA" case, it can be easily modified to accommodate it.

5.3 The ANO-qCCA security of $\mathsf{PKE}_m^{\perp}$, $\mathsf{PKE}^{\perp}$, $\mathsf{PKE}_m^{\not\perp}$ and $\mathsf{PKE}^{\not\perp}$ in the QROM

We first prove the WPR-qCCA security of KEM scheme $\mathsf{KEM}_m^{\perp}$ in the QROM.

Theorem 8. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, weakly γ-spread and* SDS-IND*-secure w.r.t. QPT simulator $\mathcal{S}$. Let $\mathcal{A}$ be a* WPR-qCCA *adversary against* $\mathsf{PKE}_m^{\perp}$ *in the QROM, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively*[22]*. Then there exist a QPT simulator $\mathcal{S}'$ of* $\mathsf{PKE}_m^{\perp}$*, an* OW-CPA *adversary $\mathcal{A}_1$ against the* PKE *and a* SDS-IND *adversary $\mathcal{A}_2$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S}',\mathsf{PKE}_m^{\perp}} \le 24q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 2(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}}} + \mathsf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}.$$

The running time of adversary $\mathcal{A}_1$ and $\mathcal{A}_2$ can be bounded as

$$\mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}_2] \le \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Proof. Based on the WPR-qCCA game of $\mathsf{PKE}_m^{\perp}$ with adversary $\mathcal{A}$ and simulator $\mathcal{S}'$, define game $\mathbf{G}_{\mathcal{A}}^{b=0}$ and game $\mathbf{G}_{\mathcal{A}}^{b=1}$ as shown in Fig. 10, then we have

$$|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}]| = 2 \cdot \mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S}',\mathsf{PKE}_m^{\perp}}. \tag{53}$$

As shown in Fig. 10, $\mathcal{S}'$ generates ciphertext (c_1, c_2) by first runs $\mathcal{S}$ to get ciphertext c_1, then randomly choose $K \in \{0,1\}^k$ and compute $c_2 := \mathsf{E}(K, m^*)$. Hence $\mathcal{S}'$ also a QPT simulator.

Game $\mathbf{G}_{\mathcal{A}}^{b=0}$

1: $(pk, sk) \leftarrow \mathsf{Gen}$, $b = 0$
2: $m^* \leftarrow \mathcal{A}^{H,G,O^0_{\mathsf{dec}}}(pk)$
3: $\delta^* \xleftarrow{\$} \{0,1\}^u$
$\quad c_1 := \mathsf{Enc}(pk, \delta^*, H(\delta^*))$
$\quad K := G(\delta^*)$
$\quad c_2 := \mathsf{E}(K, m^*)$, $c_0^* := (c_1, c_2)$
4: $b' \leftarrow \mathcal{A}^{H,G,O^0_{\mathsf{dec}}}(pk, c_0^*)$
5: **Return** b'

$O^0_{\mathsf{dec}}(c)$

1: **If** c_0^* **is defined and** $c = c_0^*$, **return** $\perp$
Else return $\mathsf{Dec}_m^{\perp}(c)$

$O^1_{\mathsf{dec}}(c)$

1: **If** c_1^* **is defined and** $c = c_1^*$, **return** $\perp$
Else return $\mathsf{Dec}_m^{\perp}(c)$

Game $\mathbf{G}_{\mathcal{A}}^{b=1}$

1: $(pk, sk) \leftarrow \mathsf{Gen}$, $b = 1$
2: $m^* \leftarrow \mathcal{A}^{H,G,O^1_{\mathsf{dec}}}(pk)$
3: $\mathcal{S}'(1^\lambda, m^*)$ **perform**:
$\quad c_1 \leftarrow \mathcal{S}(1^\lambda)$
$\quad K \xleftarrow{\$} \{0,1\}^k$
$\quad c_2 := \mathsf{E}(K, m^*)$, $c_1^* := (c_1, c_2)$
4: $b' \leftarrow \mathcal{A}^{H,G,O^1_{\mathsf{dec}}}(pk, c_1^*)$
5: **Return** b'

Figure 10: Game $\mathbf{G}_{\mathcal{A}}^{b=0}$ and game $\mathbf{G}_{\mathcal{A}}^{b=1}$. Here adversary $\mathcal{A}$ can query its oracles in superposition.

By using lifting theorem Theorem 4, we can prove following lemma, its detailed proof is similar with Lemma 7 and we omit it.

Lemma 8. *There exists adversary $\mathcal{B}$ and $\mathcal{A}_1$ without query any oracles it can access such that*

$$|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}]| \le 40q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 4(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}}},$$

and

$$|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]| \le 8q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta.$$

The running time of adversary $\mathcal{B}$ and $\mathcal{A}_1$ can be bounded as

$$\mathrm{Time}[\mathcal{B}] \approx \mathrm{Time}[\mathcal{A}_1] \le \mathrm{Time}[\mathcal{A}] + O((q_G+q_H)\cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + (q_G+q_H)^2).$$

Notice that the adversary $\mathcal{B}$ in Lemma 8 does not query any oracles it can access, hence in game $\mathbf{G}_{\mathcal{B}}^{b=0}$, the $H(\delta^*)$ and $G(\delta^*)$ used to generate c_0^* both are uniformly random in adversary $\mathcal{B}$'s view. This means that the c_2 game $\mathbf{G}_{\mathcal{B}}^{b=0}$ and $\mathbf{G}_{\mathcal{B}}^{b=1}$ have the same distribution in adversary $\mathcal{B}$'s view. Then we define an adversary $\mathcal{A}_2$ against the SDS-IND security of PKE with simulator $\mathcal{S}$ as follows:

[22] Following [JZC+18, GMP22], we make the convention that q_H and q_G counts the total number of times H and G is queried in the SPR-qCCA game, respectively.

1. $\mathcal{A}_2$ gets pk and a ciphertext c from the challenger, where c is generated by either the encryption algorithm Enc or simulator $\mathcal{S}(1^\lambda)$.
2. $\mathcal{A}_2$ runs $\mathcal{B}(pk)$ to get m^*.
3. $\mathcal{A}_2$ computes $c' := \mathsf{E}(K, m^*)$, where $K \xleftarrow{\$} \{0,1\}^u$, then runs $\mathcal{B}(pk, (c, c'))$ to get b' and output it.

Obviously, $\mathcal{A}_2$ perfectly simulate game $\mathbf{G}_{\mathcal{B}}^{b=0}$ (resp. game $\mathbf{G}_{\mathcal{B}}^{b=1}$) if c is generate by the encryption algorithm Enc (resp. simulator $\mathcal{S}(1^\lambda)$). Hence we have $\mathrm{Time}[\mathcal{A}_2] \approx \mathrm{Time}[\mathcal{B}]$ and

$$|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]| = 2 \cdot \mathsf{Adv}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}^{\mathsf{SDS\text{-}IND}}. \tag{54}$$

Thus by using the upper bound given in Lemma 8, we have

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A},\mathcal{S}',\mathsf{PKE}_m^{\perp}}^{\mathsf{WPR\text{-}qCCA}} &= |\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}]|/2 \\
&\leq |\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}]|/2 + |\Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]|/2 \\
&\quad + |\Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}]|/2 \\
&\leq 24q_D \cdot \sqrt{\gamma} + 8(q_H+1) \cdot \sqrt{\delta} + 64q_H \cdot \delta + 2(q_H+q_G+1) \cdot \sqrt{\mathsf{Adv}_{\mathcal{A}_1,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}} + \mathsf{Adv}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}^{\mathsf{SDS\text{-}IND}}.
\end{aligned}$$

$\square$

Remark 5. *Note that our* WPR-qCCA *security reduction of PKE scheme* $\mathsf{PKE}_m^{\perp} = \mathsf{KEM}_m^{\perp} + \mathsf{DEM}$ *does not require any security assumptions about DEM scheme* DEM*. Intuitively speaking, the reason is that the computation of* c_1 *shown in Fig. 10 is independent of* m^**, hence we can design adversary* $\mathcal{A}_2$ *using the adversary* $\mathcal{B}$ *of Lemma 8 and directly reduce the* WPR-qCCA *security to the underlying strongly disjoint-simulatable security of* PKE*.*

Corollary 3. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* OW-CPA*-secure and* SDS-IND*-secure, then* $\mathsf{PKE}_m^{\perp}$ *is* ANO-qCCA*-secure in the QROM.*

This follows from Theorem 5, which states that the WPR-qCCA security of a PKE schemes implies its ANO-qCCA security. Similar with the proof of Theorem 8, we can also prove the WPR-qCCA security of PKE scheme $\mathsf{PKE}^{\perp}$, $\mathsf{PKE}_m^{\not\perp}$ and $\mathsf{PKE}^{\not\perp}$ in the QROM, the corresponding theorem is given in Appendix H.5. Then by using Theorem 5 again, we obtain following corollary:

Corollary 4. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* OW-CPA*-secure and* SDS-IND*-secure, then* $\mathsf{PKE}^{\perp}$*,* $\mathsf{PKE}_m^{\not\perp}$ *and* $\mathsf{PKE}^{\not\perp}$ *is* ANO-qCCA*-secure in the QROM.*

References

[ABN10] Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In *Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7*, pages 480–497. Springer, 2010.

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In *Annual International Cryptology Conference*, pages 269–295. Springer, 2019.

[BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248, pages 566–582. Springer, 2001.

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

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *International conference on the theory and application of cryptology and information security*, pages 41–69. Springer, 2011.

[BHH+19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *Theory of Cryptography Conference*, pages 61–90. Springer, 2019.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, volume 2595, pages 62–75. Springer, 2002.

[BZ13] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Annual cryptology conference*, pages 361–379. Springer, 2013.

[CFHL21] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. 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*, pages 598–629. Springer, 2021.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 1–29. Springer, 2019.

[CMSZ19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. *IACR Cryptol. ePrint Arch.*, page 428, 2019.

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, 2003.

[Den03] Alexander W. Dent. A designer's guide to kems. In *IMA International Conference on Cryptography and Coding*, pages 133–151. Springer, 2003.

[DFMS22] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 677–706. Springer, 2022.

[DHK+22] Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Group action key encapsulation and non-interactive key exchange in the QROM. *IACR Cryptol. ePrint Arch.*, page 1230, 2022.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *J. Cryptol.*, 26(1):80–101, 2013.

[GMP22] Paul Grubbs, Varun Maram, and Kenneth G. Paterson. Anonymous, robust post-quantum public key encryption. In *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III*, volume 13277, pages 402–432. Springer, 2022.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *Theory of Cryptography Conference*, pages 341–371. Springer, 2017.

[HHM22] Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. 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*, volume 13794 of *Lecture Notes in Computer Science*, pages 414–443. Springer, 2022.

[HKSU20] Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 389–422. Springer, 2020.

[JZC+18] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *Annual International Cryptology Conference*, pages 96–125. Springer, 2018.

[JZM19] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In *International Conference on Post-Quantum Cryptography*, pages 227–248. Springer, 2019.

[KN22] Fuyuki Kitagawa and Ryo Nishimaki. KDM security for the fujisaki-okamoto transformations in the QROM. In *Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings, Part II*, volume 13178, pages 286–315. Springer, 2022.

[KS20] Juliane Krämer and Patrick Struck. Encryption schemes using random oracles: From classical to post-quantum security. In *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings*, volume 12100 of *Lecture Notes in Computer Science*, pages 539–558. Springer, 2020.

[KSS+20] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 703–728. Springer, 2020.

[LW21] Xu Liu and Mingqiang Wang. Qcca-secure generic key encapsulation mechanism with tighter security in the quantum random oracle model. In *IACR International Conference on Public-Key Cryptography*, pages 3–26. Springer, 2021.

[NC16] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition).* Cambridge University Press, 2016.

[NIS17] NIST. National institute for standards and technology. post quantum crypto project. https://csrc.nist.gov/projects/post-quantum-cryptography, 2017.

[NIS22] NIST. National institute for standards and technology. post quantum crypto project. selected algorithms. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022, 2022.

[SGX23] Tianshu Shan, Jiangxia Ge, and Rui Xue. Qcca-secure generic transformations in the quantum random oracle model. In *Public-Key Cryptography - PKC 2023 - 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7-10, 2023, Proceedings, Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 36–64. Springer, 2023.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2):303–332, 1999.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptol. ePrint Arch.*, page 332, 2004.

[Xag22] Keita Xagawa. Anonymity of NIST PQC round 3 kems. 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*, volume 13277, pages 551–581. Springer, 2022.

[XY19] Keita Xagawa and Takashi Yamakawa. (tightly) qcca-secure key-encapsulation mechanism in the quantum random oracle model. In *International Conference on Post-Quantum Cryptography*, pages 249–268. Springer, 2019.

[YZ21] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 568–597. Springer, 2021.

[Zha12] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In *Annual International Cryptology Conference*, pages 758–775. Springer, 2012.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Annual International Cryptology Conference*, pages 239–268. Springer, 2019.

A Quantum Background

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

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

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

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

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

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

There is a well-known fact that we can construct a unitary variant $\mathcal{A}^O_{\mathrm{U}}(z)$ for any quantum oracle algorithm $\mathcal{A}^O(z)$ with some constant factor computational overhead and these two algorithms have same query width and query depth [AHU19], $\mathcal{A}^O_{\mathrm{U}}(z)$ also called a unitary quantum oracle algorithm. As shown in the Definition 8 of [DHK+22], the detailed execution of a unitary quantum oracle algorithm can be described as follows:

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

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

Here $\mathrm{U}_1, \ldots, \mathrm{U}_d$ are the fixed unitary operations applied between queries, $|\psi\rangle$ is the initial pure state. $\mathcal{B}$ perform a projective measurement on its quantum register after applying U_d and output the measure outcome. For multiple oracles case, as explained in the Remark 8 of [DHK+22], if $\mathcal{B}$ have quantum access to all oracles, then the execution of $\mathcal{B}$ can be described analogously, .

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

As for the detailed representation of $\{0,1\}^n \cup \perp$, we use the extension method introduced in [CFHL21]. That is to say, we use a classical encoding function *enc* that $enc(\perp) = 1||0^n \in \{0,1\}^{n+1}$ and $enc(x) = 0||x \in \{0,1\}^{n+1}$ for any $x \in \{0,1\}^n$, then the set $\{0,1\}^n \cup \perp$ can be embedded into the set $\{0,1\}^{n+1}$. Under this representation, the binary operation $x \oplus y$ for $x, y \in \{0,1\}^n \cup \perp$ that used in this paper actually means $enc(x) \oplus enc(y)$, where operation $\oplus$ denotes bitwise addition modulo 2, a group operation on $\{0,1\}^{n+1}$. Overall, with this representation, the quantum register defined over set $\{0,1\}^n \cup \perp$ is implemented by a quantum register defined over set $\{0,1\}^{n+1}$.

B QROM Lemmas

Lemma 9 (Simulate the QROM [Zha12])**.** *Let O be a random oracle, and H be a function uniformly chosen from the set of $2q$-wise independent functions. For any algorithm $\mathcal{A}$ that has quantum access to its oracle and makes at most q queries, we have* $\Pr[1 \leftarrow \mathcal{A}^H(z)] = \Pr[1 \leftarrow \mathcal{A}^O(z)]$ *for any input z.*

Semi-classical oracle. For subset $S \subseteq \{0,1\}^m$, Let f_S be the function that $f_S(x) = 1$ if $x \in S$, and $f_S(x) = 0$ otherwise. The semi-classical oracle $\mathcal{O}_S^{SC}$ performs the following operation on input state $\sum_{x\in\mathcal{X}, z\in\{0,1\}^*} \alpha_{x,z}|x,z\rangle$:

1. Initialize a single qubit L with $|0\rangle_L$, transform state $\sum_{x\in\mathcal{X}, z\in\{0,1\}^*} \alpha_{x,z}|x,z\rangle|0\rangle_L$ into state $\sum_{x\in\mathcal{X}, z\in\{0,1\}^*} \alpha_{x,z}|x,z\rangle|f_S(x)\rangle_L$.
2. Measure L and output the measurement outcome.

In the execution of an quantum algorithm that has oracle access to $\mathcal{O}_S^{SC}$, Denote Find as the event that $\mathcal{O}_S^{SC}$ ever outputs 1.

Lemma 10 (Semi-classical O2H [AHU19])**.** *Let $H, G : \{0,1\}^m \to \{0,1\}^n$ be random functions such that $H(x) = G(x)$ for any $x \notin S$, where $S \subseteq \{0,1\}^m$. Let z be a random bitstring, suppose that H, G, S, z may have arbitrary joint distribution $\mathcal{D}$. Let $H\backslash S$ be an oracle that first queries $\mathcal{O}_S^{SC}$ and then queries H.*

Let $\mathcal{A}$ be an oracle algorithm (not necessarily unitary) with query depth d. Define

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

Then

$$|P_{\text{left}} - P_{\text{right}}| \leq \sqrt{(d+1)\cdot P_{\text{find}}}, \quad |P_{\text{left}} - P'_{\text{right}}| \leq 2\sqrt{(d+1)\cdot P_{\text{find}}}.$$

Lemma 11 (Search in the semi-classical oracle [AHU19])**.** *Let $\mathcal{A}$ be a quantum oracle algorithm making at most d queries to the semi-classical oracle with domain $\{0,1\}^m$. Let $S \subseteq \{0,1\}^m$ and $z \in \{0,1\}^*$, suppose that S, z may have arbitrary joint distribution $\mathcal{D}$. Let $\mathcal{B}$ be an algorithm that on input z chooses $i \xleftarrow{\$} \{1,\ldots,d\}$, runs $\mathcal{A}^{\mathcal{O}_\varnothing^{SC}}(z)$ until (just before) the i-th query, then measures all query input registers in the computational basis and outputs the set T of the measurement outcomes. Then*

$$\Pr[\mathsf{Find} \textit{ occurs in } \mathcal{A}^{\mathcal{O}_S^{SC}}(z) : (S,z) \leftarrow \mathcal{D}] \leq 4d \cdot \Pr[S \cap T \neq \varnothing \wedge T \leftarrow \mathcal{B}(z) : (S,z) \leftarrow \mathcal{D}].$$

C Proof of Lemma 2

Proof of Lemma 2. By Eq. (13), we have

$$\begin{aligned}\|[\mathsf{Ext}_f, \mathsf{StdDecomp}_x]\| &= \left\|\left[\sum_{t\in\mathcal{Y}} |t\rangle\langle t|_{\mathsf{T}} \otimes \mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]\right\| \\ &= \left\|\sum_{t\in\mathcal{Y}} |t\rangle\langle t|_{\mathsf{T}} \otimes \left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]\right\| \\ &\overset{(a)}{\leq} \max_{t\in\mathcal{Y}} \left\|\left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]\right\|,\end{aligned}$$

where (a) uses the following corollary:

Corollary 5 ([DFMS22], Corollary 2.2)**.** *If $A = \sum_x |x\rangle\langle x| \otimes A^x$, i.e., A is a controlled operator, then $\|A\| \leq \max_x \|A^x\|$.*

By the result of Appendix C.1,

$$\left\|\left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]\right\| \leq 16 \cdot \sqrt{\Gamma_{R_t^f}/2^n}.$$

Then

$$\|[\mathsf{Ext}_f, \mathsf{StdDecomp}_x]\| \leq \max_{t\in\mathcal{Y}} \left\|\left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]\right\| \leq 16 \cdot \sqrt{\max_{t\in\mathcal{Y}} \Gamma_{R_t^f}/2^n}.$$

By the definition of CStO in Definition 2, we have

$$\begin{aligned}\|[\mathsf{CStO}, \Sigma^\perp]\| &= \left\|\left[\sum_{x\in\{0,1\}^m} |x\rangle\langle x|_{\mathsf{X}} \otimes \mathsf{StdDecomp}_x \circ \mathsf{CNOT}^x_{\mathsf{YD}_q} \circ \mathsf{StdDecomp}_x, \Sigma^\perp\right]\right\| \\ &\overset{(b)}{\leq} \max_{x\in\{0,1\}^m} \|[\mathsf{StdDecomp}_x \circ \mathsf{CNOT}^x_{\mathsf{YD}_q} \circ \mathsf{StdDecomp}_x, \Sigma^\perp]\| \\ &\overset{(c)}{\leq} 2 \cdot \max_{x\in\{0,1\}^m} \|[\mathsf{StdDecomp}_x, \Sigma^\perp]\|.\end{aligned}$$

Here (b) uses Corollary 5 again, (c) uses the fact that $\mathsf{CNOT}^x_{\mathsf{YD}_q}$ is naturally commute with $\Sigma^\perp$ for any $x \in \{0,1\}^m$.

By the result of Appendix C.2,

$$\left\|\left[\mathsf{StdDecomp}_x, \Sigma^\perp\right]\right\| \leq 4 \cdot \sqrt{|\Gamma_x|/2^n},$$

where set $\Gamma_x := \{y \in \{0,1\}^n | f(x,y) = t\}$, then by Eq. (11)

$$\|[\mathsf{CStO}, \Sigma^\perp]\| \leq 8 \max_{x\in\{0,1\}^m} \sqrt{|\Gamma_x|/2^n} = 8 \cdot \sqrt{\Gamma_{R_t^f}/2^n}.$$

□

C.1 Bound on $\left\|\left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]\right\|$

For fixed function f, $t \in \mathcal{Y}$ and $x \in \{0,1\}^m$, define set $\Gamma_x := \{y \in \{0,1\}^n | f(x,y) = t\}$. As defined in Section 2.4 and Section 2.5, $\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}$ acts on registers $\mathsf{D}_q\mathsf{P}$ and $\mathsf{StdDecomp}_x$ acts on register D_q. Moreover, for a computational basis state $|D,p\rangle$ on registers $\mathsf{D}_q\mathsf{P}$, where $D \in \mathbf{D}_q$ and $p \in \{0,1\}^{m+1}$, it is straightforward to check that

$$\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}|D,p\rangle = \begin{cases} |D, p\oplus z\rangle & \text{if } (z, D(z)) \in R_t^f \wedge \nexists z' < z \text{ s.t. } (z', D(z')) \in R_t^f, \\ |D, p \oplus \perp\rangle & \text{otherwise.}\end{cases} \tag{55}$$

For any state $|\Phi\rangle$ on registers $\mathsf{D}_q\mathsf{P}$ with norm 1, we can denote

$$|\Phi\rangle = \sum_{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}} \alpha_{D,p}|D,p\rangle,$$

where $\sum_{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}} |\alpha_{D,p}|^2 = 1$. Next, by using x, we can separate state $|\Phi\rangle$ into eight mutual orthogonal parts $|\Phi_1\rangle$ to $|\Phi_8\rangle$ that

$$|\Phi\rangle = \sum_{i=1}^{8} |\Phi_i\rangle, \ \ \||\Phi\rangle\|^2 = \sum_{i=1}^{8} \||\Phi_i\rangle\|^2. \tag{56}$$

Here $|\Phi_1\rangle$ to $|\Phi_8\rangle$ are the following states:

$$\begin{aligned}
|\Phi_1\rangle &= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1} \\ D(x)=\perp, \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,p}|D,p\rangle, \\
|\Phi_2\rangle &= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1} \\ D(x)=\perp, \exists z_D<x \text{ s.t. } (z_D,D(z_D))\in R_t^f}} \beta_{D,p}|D,p\rangle, \\
|\Phi_3\rangle &= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1} \\ D(x)=\perp, \exists z_D>x \text{ s.t. } (z_D,D(z_D))\in R_t^f \\ \nexists z'<z_D \text{ s.t. } (z',D(z'))\in R_t^f}} \beta_{D,p}|D,p\rangle, \\
|\Phi_4\rangle &= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ r\in\{0,1\}^n, r\neq 0^n \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,p,r}|D\cup(x,\hat{r}),p\rangle, \\
|\Phi_5\rangle &= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ r=0^n \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,p,r}|D\cup(x,\hat{r}),p\rangle, \\
|\Phi_6\rangle &= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ r\in\{0,1\}^n \\ D(x)=\perp, \exists z_D<x \text{ s.t. } (z_D,D(z_D))\in R_t^f}} \beta_{D,p,r}|D\cup(x,\hat{r}),p\rangle, \\
|\Phi_7\rangle &= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ r\in\{0,1\}^n, r\neq 0^n \\ D(x)=\perp, \exists z_D>x \text{ s.t. } (z_D,D(z_D))\in R_t^f \\ \nexists z'<z_D, z'\neq x \text{ s.t. } (z',D(z'))\in R_t^f}} \beta_{D,p,r}|D\cup(x,\hat{r}),p\rangle, \\
|\Phi_8\rangle &= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ r=0^n \\ D(x)=\perp, \exists z_D>x \text{ s.t. } (z_D,D(z_D))\in R_t^f \\ \nexists z'<z_D, z'\neq x \text{ s.t. } (z',D(z'))\in R_t^f}} \beta_{D,p,r}|D\cup(x,\hat{r}),p\rangle.
\end{aligned}$$

Let $|\Psi_i\rangle := \left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]|\Phi_i\rangle$ for $i=1,\ldots,8$, by Eq. (55) and the definition of $\mathsf{StdDecomp}_x$

defined in Section 2.4, we compute:

$$|\Psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}\\ D(x)=\perp, \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,p}\left(|D\cup(x,y), p\oplus x\rangle - |D\cup(x,y), p\oplus\perp\rangle\right),$$

$$|\Psi_2\rangle = \mathbf{0},$$

$$|\Psi_3\rangle = \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}\\ D(x)=\perp, \exists z_D > x \text{ s.t. } (z_D,D(z_D))\in R_t^f\\ \nexists z' < z_D \text{ s.t. } (z',D(z'))\in R_t^f}} \frac{\beta_{D,p}}{\sqrt{2^n}}\left(|D\cup(x,y), p\oplus x\rangle - |D\cup(x,y), p\oplus z_D\rangle\right),$$

$$|\Psi_4\rangle = \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q\\ r\in\{0,1\}^n, r\neq 0^n\\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \frac{(-1)^{y\cdot r}}{2^n}\beta_{D,p,r}\begin{pmatrix} |D\cup(x,\hat{0^n}), p\oplus x\rangle - |D, p\oplus x\rangle\\ +|D, p\oplus\perp\rangle - |D\cup(x,\hat{0^n}), p\oplus\perp\rangle \end{pmatrix},$$

$$|\Psi_5\rangle = \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q\\ r=0^n\\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \frac{\beta_{D,p,r}}{\sqrt{2^n}}\cdot \mathsf{StdDecomp}_x\begin{pmatrix} |D\cup(x,y), p\oplus\perp\rangle\\ -|D\cup(x,y), p\oplus x\rangle \end{pmatrix},$$

$$|\Psi_6\rangle = \mathbf{0},$$

$$|\Psi_7\rangle = \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q\\ r\in\{0,1\}^n, r\neq 0^n\\ D(x)=\perp, \exists z_D > x \text{ s.t. } (z_D,D(z_D))\in R_t^f\\ \nexists z' < z_D, z'\neq x \text{ s.t. } (z',D(z'))\in R_t^f}} \frac{(-1)^{y\cdot r}}{2^n}\beta_{D,p,r}\begin{pmatrix} |D\cup(x,\hat{0^n}), p\oplus x\rangle - |D, p\oplus x\rangle\\ +|D, p\oplus z_D\rangle - |D\cup(x,\hat{0^n}), p\oplus z_D\rangle \end{pmatrix},$$

$$|\Psi_8\rangle = \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q\\ r=0^n\\ D(x)=\perp, \exists z_D > x \text{ s.t. } (z_D,D(z_D))\in R_t^f\\ \nexists z' < z_D, z'\neq x \text{ s.t. } (z',D(z'))\in R_t^f}} \frac{\beta_{D,p,r}}{\sqrt{2^n}}\cdot \mathsf{StdDecomp}_x\begin{pmatrix} |D\cup(x,y), p\oplus z_D\rangle\\ -|D\cup(x,y), p\oplus x\rangle \end{pmatrix}.$$

For state $|\Psi_1\rangle$, we compute

$$\begin{aligned}
\||\Psi_1\rangle\|^2 &= \left\|\frac{1}{\sqrt{2^n}} \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}\\ D(x)=\perp, \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,p}\left(|D\cup(x,y), p\oplus x\rangle - |D\cup(x,y), p\oplus\perp\rangle\right)\right\|^2\\
&\overset{(a)}{\leq} 2\left\|\frac{1}{\sqrt{2^n}} \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}\\ D(x)=\perp, \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,p}|D\cup(x,y), p\oplus x\rangle\right\|^2\\
&\qquad + 2\left\|\frac{1}{\sqrt{2^n}} \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}\\ D(x)=\perp, \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,p}|D\cup(x,y), p\oplus\perp\rangle\right\|^2 \qquad (57)\\
&= \frac{2}{2^n}\sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}\\ D(x)=\perp, \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} |\beta_{D,p}|^2 + \frac{2}{2^n}\sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}\\ D(x)=\perp, \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} |\beta_{D,p}|^2\\
&= \frac{4|\Gamma_x|}{2^n} \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}\\ D(x)=\perp, \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} |\beta_{D,p}|^2 = \frac{4|\Gamma_x|}{2^n}\||\Phi_1\rangle\|^2.
\end{aligned}$$

Here (a) uses the following corollary.

Corollary 6. *For any state $|\psi_1\rangle$ to $|\psi_q\rangle$, we have $\|\sum_{i=1}^{q}|\psi_i\rangle\|^2 \le q\cdot\sum_{i=1}^{q}\||\psi_i\rangle\|^2$.*

Proof of Corollary 6. The proof is simple:

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

Here (a) uses the triangle inequality, and (b) uses the AM-QM (or Jensen's) inequality. □

Similar with the computation of $\||\Psi_1\rangle\|^2$, we also have

$$\||\Psi_3\rangle\|^2 \le \frac{4|\Gamma_x|}{2^n}\||\Phi_3\rangle\|^2,\ \||\Psi_5\rangle\|^2 \le \frac{4|\Gamma_x|}{2^n}\||\Phi_5\rangle\|^2,\ \||\Psi_8\rangle\|^2 \le \frac{4|\Gamma_x|}{2^n}\||\Phi_8\rangle\|^2. \tag{58}$$

For state $|\Psi_4\rangle$, we compute

$$\begin{aligned}
\||\Psi_4\rangle\|^2 &= \left\|\sum_{y\in\Gamma_x}\sum_{\substack{D\in\mathbf{D}_q,p\in\{0,1\}^{m+1},n(D)<q\\ r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\frac{(-1)^{y\cdot r}}{2^n}\beta_{D,p,r}\begin{pmatrix}|D\cup(x,\hat{0^n}),p\oplus x\rangle-|D,p\oplus x\rangle\\ +|D,p\oplus\perp\rangle-|D\cup(x,\hat{0^n}),p\oplus\perp\rangle\end{pmatrix}\right\|^2\\
&\overset{(b)}{\le} 4\left\|\sum_{y\in\Gamma_x}\sum_{\substack{D\in\mathbf{D}_q,p\in\{0,1\}^{m+1},n(D)<q\\ r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\frac{(-1)^{y\cdot r}}{2^n}\beta_{D,p,r}|D\cup(x,\hat{0^n}),p\oplus x\rangle\right\|^2\\
&\quad+4\left\|\sum_{y\in\Gamma_x}\sum_{\substack{D\in\mathbf{D}_q,p\in\{0,1\}^{m+1},n(D)<q\\ r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\frac{(-1)^{y\cdot r}}{2^n}\beta_{D,p,r}|D,p\oplus x\rangle\right\|^2\\
&\quad+4\left\|\sum_{y\in\Gamma_x}\sum_{\substack{D\in\mathbf{D}_q,p\in\{0,1\}^{m+1},n(D)<q\\ r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\frac{(-1)^{y\cdot r}}{2^n}\beta_{D,p,r}|D,p\oplus\perp\rangle\right\|^2\\
&\quad+4\left\|\sum_{y\in\Gamma_x}\sum_{\substack{D\in\mathbf{D}_q,p\in\{0,1\}^{m+1},n(D)<q\\ r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\frac{(-1)^{y\cdot r}}{2^n}\beta_{D,p,r}|D\cup(x,\hat{0^n}),p\oplus\perp\rangle\right\|^2\\
&= \frac{16}{2^n}\sum_{\substack{D\in\mathbf{D}_q,p\in\{0,1\}^{m+1},n(D)<q\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\left|\sum_{y\in\Gamma_x,r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,p,r}\right|^2\\
&\overset{(c)}{\le}\frac{16|\Gamma_x|}{2^n}\sum_{\substack{D\in\mathbf{D}_q,p\in\{0,1\}^m\cup\perp,n(D)<q\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\sum_{y\in\Gamma_x}\left|\sum_{r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,p,r}\right|^2.
\end{aligned} \tag{59}$$

Here (b) uses Corollary 6 again, and (c) uses the Cauchy-Schwarz inequality.

Indeed, we can compute

$$\begin{aligned}
\||\Phi_4\rangle\|^2 &= \left\| \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ r\in\{0,1\}^n, r\neq 0^n \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,p,r}|D\cup(x,\hat{r}),p\rangle \right\|^2 \\
&= \left\| \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ r\in\{0,1\}^n, r\neq 0^n \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \sum_{y\in\{0,1\}^n} \beta_{D,p,r}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}|D\cup(x,y),p\rangle \right\|^2 \\
&= \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \sum_{y\in\{0,1\}^n} \left| \sum_{r\in\{0,1\}^n, r\neq 0^n} \beta_{D,p,r}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}} \right|^2 \\
&\geq \sum_{\substack{D\in\mathbf{D}_q, p\in\{0,1\}^{m+1}, n(D)<q \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \sum_{y\in\Gamma_x} \left| \sum_{r\in\{0,1\}^n, r\neq 0^n} \frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,p,r} \right|^2 .
\end{aligned}$$

Combine above inequality with Eq. (59), we get

$$\||\Psi_4\rangle\|^2 \leq \frac{16|\Gamma_x|}{2^n}\||\Phi_4\rangle\|^2. \tag{60}$$

Similar with the computation of $\||\Psi_4\rangle\|^2$, we also have

$$\||\Psi_7\rangle\|^2 \leq \frac{16|\Gamma_x|}{2^n}\||\Phi_7\rangle\|^2. \tag{61}$$

Combining Eq. (57), (58), (60) and (61), we have

$$\begin{aligned}
\left\|\left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]\right\| &= \max_{|\Phi\rangle, \||\Phi\rangle\|=1} \left\|\left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]|\Phi\rangle\right\| \\
&= \max_{|\Phi\rangle, \||\Phi\rangle\|=1} \left\|\left[\mathrm{M}^{R_t^f}_{\mathsf{D}_q\mathsf{P}}, \mathsf{StdDecomp}_x\right]\sum_{i=1}^{8}|\Phi_i\rangle\right\| \\
&\overset{(d)}{\leq} 4\cdot\frac{2\sqrt{|\Gamma_x|}}{\sqrt{2^n}} + 2\cdot\frac{4\sqrt{|\Gamma_x|}}{\sqrt{2^n}} \\
&= 16\sqrt{\frac{|\Gamma_x|}{2^n}} \overset{(e)}{\leq} 16\cdot\sqrt{\frac{\Gamma_{R_t^f}}{2^n}}.
\end{aligned}$$

Here (d) uses the triangle inequality and Eq. (56), (e) uses the fact that $\Gamma_{R_t^f} = \max_{x\in\{0,1\}^m}|\Gamma_x|$.

C.2 Bound on $\left\|\left[\mathsf{StdDecomp}_x, \Sigma^\perp\right]\right\|$

For fixed function f, $t\in\mathcal{Y}$ and $x\in\{0,1\}^m$, define set $\Gamma_x := \{y\in\{0,1\}^n | f(x,y)=t\}$. For any state $|\Phi\rangle = \sum_{D\in\mathbf{D}_q}\alpha_D|D\rangle$ on register D_q with norm 1 ($\sum_{D\in\mathbf{D}_q}|\alpha_D|^2=1$), we separate $|\Phi\rangle$ into four mutual orthogonal parts that

$$|\Phi\rangle = \sum_{i=1}^{4}|\Phi_i\rangle,\ \||\Phi\rangle\|^2 = \sum_{i=1}^{4}\||\Phi_i\rangle\|^2. \tag{62}$$

Here $|\Phi_1\rangle$ to $|\Phi_4\rangle$ are the following states:

$$
\begin{aligned}
|\Phi_1\rangle &= \sum_{D\in\mathbf{D}_q, \exists z\neq x \text{ s.t. } (z,D(z))\in R_t^f} \beta_D|D\rangle, \\
|\Phi_2\rangle &= \sum_{\substack{D\in\mathbf{D}_q, D(x)=\perp \\ \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \beta_D|D\rangle, \\
|\Phi_3\rangle &= \sum_{\substack{D\in\mathbf{D}_q, n(D)<q, r\in\{0,1\}^n, r\neq 0^n \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,r}|D\cup(x,\hat{r})\rangle, \\
|\Phi_4\rangle &= \sum_{\substack{D\in\mathbf{D}_q, n(D)<q, r=0^n \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,r}|D\cup(x,\hat{r})\rangle.
\end{aligned}
$$

Let $|\Psi_i\rangle := \left[\mathsf{StdDecomp}_x, \Sigma^\perp\right]|\Phi_i\rangle$ for $i = 1, \ldots, 4$, by the definition of $\mathsf{StdDecomp}_x$ and $\Sigma^\perp$ defined in Section 2.4 and Section 2.5, respectively, we compute:

$$
\begin{aligned}
|\Psi_1\rangle &= \mathbf{0}, \\
|\Psi_2\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, D(x)=\perp \\ \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \beta_D|D\cup(x,y)\rangle, \\
|\Psi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, n(D)<q, r\in\{0,1\}^n, r\neq 0^n \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,r}(|D\cup(x,\hat{0^n})\rangle - |D\rangle), \\
|\Psi_4\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, n(D)<q, r=0^n \\ D(x)=\perp, \nexists z\neq x \text{ s.t. } (z,D(z))\in R_t^f}} \beta_{D,r}|D\cup(x,y)\rangle.
\end{aligned}
$$

For state $|\Psi_2\rangle$, we compute

$$
\begin{aligned}
\||\Psi_2\rangle\|^2 &= \left\| \frac{1}{\sqrt{2^n}} \sum_{y\in\Gamma_x} \sum_{\substack{D\in\mathbf{D}_q, D(x)=\perp \\ \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \beta_D|D\cup(x,y)\rangle \right\|^2 \\
&= \frac{1}{2^n} \sum_{\substack{D\in\mathbf{D}_q, D(x)=\perp \\ \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} \sum_{y\in\Gamma_x} |\beta_D|^2 \qquad (63) \\
&= \frac{|\Gamma_x|}{2^n} \sum_{\substack{D\in\mathbf{D}_q, D(x)=\perp \\ \nexists z \text{ s.t. } (z,D(z))\in R_t^f}} |\beta_D|^2 = \frac{|\Gamma_x|}{2^n}\||\Phi_2\rangle\|^2.
\end{aligned}
$$

Similar with the computation of $\||\Psi_2\rangle\|^2$, we also have

$$
\||\Psi_4\rangle\|^2 \leq \frac{|\Gamma_x|}{2^n}\||\Phi_4\rangle\|^2. \qquad (64)
$$

For state $|\Psi_3\rangle$, we compute

$$\begin{aligned}
\||\Psi_3\rangle\|^2 &= \left\|\frac{1}{\sqrt{2^n}}\sum_{y\in\Gamma_x}\sum_{\substack{D\in\mathbf{D}_q,n(D)<q,r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,r}(|D\cup(x,\hat{0^n})\rangle-|D\rangle)\right\|^2\\
&\overset{(a)}{\leq} 2\left\|\frac{1}{\sqrt{2^n}}\sum_{y\in\Gamma_x}\sum_{\substack{D\in\mathbf{D}_q,n(D)<q,r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,r}|D\cup(x,\hat{0^n})\rangle\right\|^2\\
&\quad+2\left\|\frac{1}{\sqrt{2^n}}\sum_{y\in\Gamma_x}\sum_{\substack{D\in\mathbf{D}_q,n(D)<q,r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,r}|D\rangle\right\|^2 \qquad (65)\\
&= \frac{4}{2^n}\sum_{\substack{D\in\mathbf{D}_q,n(D)<q\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\left|\sum_{y\in\Gamma_x,r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,r}\right|^2\\
&\overset{(b)}{\leq}\frac{4|\Gamma_x|}{2^n}\sum_{\substack{D\in\mathbf{D}_q,n(D)<q\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\sum_{y\in\Gamma_x}\left|\sum_{r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,r}\right|^2.
\end{aligned}$$

Here (a) uses Corollary 6, (b) uses the Cauchy-Schwarz inequality. In addition, we have

$$\begin{aligned}
\||\Phi_3\rangle\|^2 &= \left\|\sum_{\substack{D\in\mathbf{D}_q,n(D)<q,r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\beta_{D,r}|D\cup(x,\hat{r})\rangle\right\|^2\\
&= \left\|\sum_{\substack{D\in\mathbf{D}_q,n(D)<q,r\in\{0,1\}^n,r\neq 0^n\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\sum_{y\in\{0,1\}^n}\beta_{D,r}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}|D\cup(x,y)\rangle\right\|^2\\
&= \sum_{\substack{D\in\mathbf{D}_q,n(D)<q\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\sum_{y\in\{0,1\}^n}\left|\sum_{r\in\{0,1\}^n,r\neq 0^n}\beta_{D,p,r}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\right|^2\\
&\geq \sum_{\substack{D\in\mathbf{D}_q,n(D)<q\\ D(x)=\perp,\nexists z\neq x\text{ s.t. }(z,D(z))\in R_t^f}}\sum_{y\in\Gamma_x}\left|\sum_{r\in\{0,1\}^n,r\neq 0^n}\frac{(-1)^{y\cdot r}}{\sqrt{2^n}}\beta_{D,p,r}\right|^2.
\end{aligned}$$

Combine above inequality with Eq. (65), we get

$$\||\Psi_3\rangle\|^2 \leq \frac{4|\Gamma_x|}{2^n}\||\Phi_3\rangle\|^2. \qquad (66)$$

Combining Eq. (63), (64) and (66), we have

$$\begin{aligned}\left\|\left[\mathsf{StdDecomp}_x, \Sigma^{\perp}\right]\right\| &= \max_{|\Phi\rangle, \||\Phi\rangle\|=1} \left\|\left[\mathsf{StdDecomp}_x, \Sigma^{\perp}\right]|\Phi\rangle\right\| \\ &= \max_{|\Phi\rangle, \||\Phi\rangle\|=1} \left\|\left[\mathsf{StdDecomp}_x, \Sigma^{\perp}\right]\sum_{i=1}^{4}|\Phi\rangle\right\| \\ &\overset{(c)}{\leq} \sqrt{\frac{|\Gamma_x|}{2^n}} + \sqrt{\frac{|\Gamma_x|}{2^n}} + 2\cdot\sqrt{\frac{|\Gamma_x|}{2^n}} = 4\cdot\sqrt{\frac{|\Gamma_x|}{2^n}}.\end{aligned}$$

Here (c) uses the triangle inequality and Eq. (62).

D Proof of Theorem 3

Proof of Theorem 3. Without loss of generality, we can assume that $\mathcal{A}$ is a unitary quantum oracle algorithm: If $\mathcal{A}$ is not a unitary quantum oracle algorithm, we can efficiently construct a unitary variant of $\mathcal{A}$ by the well-known fact mentioned in Appendix A. Then, we suppose that S and z are fixed. Denote Q as the quantum register of $\mathcal{A}$, let L be a "query log" register consisting of q_1 qubits. Define

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

Then

$$P_{\text{left}} = \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}} P_{\text{left}}^{S,z},\ P_{\text{right}} = \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}} P_{\text{right}}^{S,z},\ P_{\text{find}} = \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}} P_{\text{find}}^{S,z}.$$

Define a quantum algorithm $\mathcal{B}_1(S,z)$ executed on quantum registers Q, D_q and L as follows:

1. Initialize the register L with state $|0^{q_1}\rangle$.
2. $\mathcal{B}_1(S,z)$ implements the compressed standard oracle with database register D_q, the initial state on D_q is $|D^{\perp}\rangle$.
3. $\mathcal{B}_1(S,z)$ performs all operations that $\mathcal{A}^{H,\mathsf{oRead}_f}(z)$ does. Here $\mathcal{B}_1(S,z)$ can implement queries to H and oRead_f by unitary operation CStO and Read_f, respectively.
4. Measure register L to get outcome 0^{q_1}, then measure register Q to get the output of $\mathcal{A}^{H,\mathsf{oRead}_f}(z)$ and output it.

Obviously register L has no effect on the execution of $\mathcal{A}^{H,\mathsf{oRead}_f}(z)$, as it is always $|0^{q_1}\rangle$, hence we get

$$\Pr[1 \leftarrow \mathcal{B}_1(S,z) : (S,z)] = \Pr[1 \leftarrow \mathcal{A}^{H,\mathsf{oRead}_f}(z) : (S,z)] = P_{\text{left}}^{S,z}.$$

Next we define a new quantum algorithm $\mathcal{B}_2(S,z)$ executed on registers Q, D_q and L as follows:

1. Initialize the register L with state $|0^{q_1}\rangle$.
2. $\mathcal{B}_2(S,z)$ implements the compressed standard oracle with database register D_q, the initial state on D_q is $|D^{\perp}\rangle$.
3. $\mathcal{B}_2(S,z)$ performs all operations that $\mathcal{A}^{H,\mathsf{oRead}_f}(z)$ does. Here $\mathcal{B}_2(S,z)$ can implement queries to H and oRead_f by operation CStO and Read_f, respectively.
4. For all $1 \leq i \leq q_1$, just after $\mathcal{A}^{H,\mathsf{oRead}_f}(z)$ performs its i-th oracle query to H, $\mathcal{B}_2(S,z)$ applies the unitary operation U_S to registers D_q and L. Here U_S is defined as[23]:

$$U_S|D\rangle|l_1,l_2,\ldots,l_{q_1}\rangle := \begin{cases} |D\rangle|l_1,\ldots,l_{q_1}\rangle & (D\notin S) \\ |D\rangle|l_1,\ldots,l_i\oplus 1,\ldots,l_{q_1}\rangle & (D\in S). \end{cases}$$

[23]Note that the unitary operation U_S should be related to the query number i, however, we omit it for simplify.

5. Measure register L to get outcome l, then measure register Q to get the output of $\mathcal{A}^{H,\mathsf{oRead}_f}(z)$ and output it.

It is straightforward to check that

$$\Pr[1 \leftarrow \mathcal{B}_2(S,z) : (S,z)] = \Pr[1 \leftarrow \mathcal{A}^{H\setminus S,\mathsf{oRead}_f}(z) : (S,z)] = P_{\text{right}}^{S,z},$$
$$\Pr[l \neq 0^{q_1} \text{ occurs in } \mathcal{B}_2(S,z) : (S,z)] = \Pr[\mathsf{Find} \text{ occurs in } \mathcal{A}^{H\setminus S,\mathsf{oRead}_f}(z) : (S,z)] = P_{\text{find}}^{S,z}.$$

Since $\mathcal{A}$ is a unitary quantum oracle algorithm, the final state of $\mathcal{B}_1(S,z)$ (resp. $\mathcal{B}_2(S,z)$) before measure can be written as

$$|\Psi_1\rangle|0^{q_1}\rangle := \prod_{i=1}^{q_1+q_2}(U_2^i \circ \mathsf{Read}_f^{y_i} \circ U_1^i \circ \mathsf{CStO}^{x_i})|\psi\rangle|D^\perp\rangle|0^{q_1}\rangle = \prod_{i=1}^{q_1}(U_3^i \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle|0^{q_1}\rangle$$
$$(\textit{resp. } |\Psi_2\rangle := \prod_{i=1}^{q_1+q_2}(U_2^i \circ \mathsf{Read}_f^{y_i} \circ U_1^i \circ U_S^{x_i} \circ \mathsf{CStO}^{x_i})|\psi\rangle|D^\perp\rangle|0^{q_1}\rangle = \prod_{i=1}^{q_1}(U_3^i \circ U_S \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle|0^{q_1}\rangle). \tag{67}$$

Here $x_i, y_i \in \{0,1\}$ and $x_i + y_i = 1$ $(1 \leq i \leq q_1 + q_2)$, $|\psi\rangle|D^\perp\rangle|0^{q_1}\rangle$ is the initial state of algorithm $\mathcal{B}_1(S,z)$ and $\mathcal{B}_2(S,z)$ on registers $Q\mathsf{D}_qL$. $U_1^1, \ldots, U_1^{q_1+q_2}$ and $U_2^1, \ldots, U_2^{q_1+q_2}$ are the unitary operation act on register Q between oracle queries, $U_3^1, \ldots, U_3^{q_1}$ are the unitary operation that alternatingly applies a unitary operation on registers Q and applies Read_f.

By the definition of unitary operation U_S, the state $|\Psi_2\rangle$ can be rewritten as

$$|\Psi_2\rangle = \sum_{l_1,\ldots,l_{q_1}\in\{0,1\}^{q_1}} \prod_{i=1}^{q_1}(U_3^i \circ \chi_{l_i} \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle|l_1,\ldots,l_{q_1}\rangle,$$

where $\chi_1 := \mathrm{J}_S$, $\chi_0 := \mathbf{I} - \mathrm{J}_S$. For a fixed q_1 bits string $l_1,\ldots,l_{q_1}$, define state

$$|\Phi\rangle_{l_1,\ldots,l_{q_1}} := \prod_{i=1}^{q_1}(U_3^i \circ \chi_{l_i} \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle,$$

we then have

$$|\Psi_2\rangle = \sum_{l_1,\ldots,l_{q_1}\in\{0,1\}^{q_1}} |\Phi\rangle_{l_1,\ldots,l_{q_1}}|l_1,\ldots,l_{q_1}\rangle$$

and

$$P_{\text{find}}^{S,z} = \sum_{l_1,\ldots,l_{q_1}\in\{0,1\}^{q_1}}^{l_1,\ldots,l_{q_1}\neq 0^{q_1}} \left\||\Phi\rangle_{l_1,\ldots,l_{q_1}}\right\|^2 = 1 - \||\Phi\rangle_{0^{q_1}}\|^2 = 1 - \left\|\prod_{i=1}^{q_1}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2. \tag{68}$$

Define values $a_1,\ldots,a_{q_1}$ and $b_1,\ldots,b_{q_1}$ as:

$$a_j := \left\|\prod_{i=1}^{j}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 \quad (j = 1,\ldots,q_1),$$

$$b_1 := \left\|U_3^1 \circ \chi_1 \circ \mathsf{CStO}|\psi\rangle|D^\perp\rangle\right\|^2, \; b_j := \left\|U_3^j \circ \chi_1 \circ \mathsf{CStO} \circ \prod_{i=1}^{j-1}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 \quad (j = 2,\ldots,q_1).$$

For $k = 2,\ldots,q_1$, we then have

$$\begin{aligned}
1 - a_k &= 1 - \left\|\prod_{i=1}^{k}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 = 1 - \left\|U_3^k \circ \chi_0 \circ \mathsf{CStO} \circ \prod_{i=1}^{k-1}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 \\
&\overset{(a)}{=} 1 - \left\|U_3^k \circ \mathbf{I} \circ \mathsf{CStO} \circ \prod_{i=1}^{k-1}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 \\
&\quad + \left\|U_3^k \circ \chi_1 \circ \mathsf{CStO} \circ \prod_{i=1}^{k-1}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 \\
&= 1 - a_{k-1} + b_k
\end{aligned}$$

Here (a) uses the fact that $\||\phi_1\rangle + |\phi_2\rangle\|^2 = \||\phi_1\rangle\|^2 + \||\phi_2\rangle\|^2$ if $|\phi_1\rangle$ and $|\phi_2\rangle$ are orthogonal. Note that $1 - a_1 = b_1$ by the definition of a_1 and b_1, then by Eq. (68), it is easily to obtain that

$$P_{\text{find}}^{S,z} = 1 - a_{q_1} = \sum_{j=1}^{q_1} b_j.$$

Define states $|A_1\rangle, \ldots, |A_{q_1}\rangle$ and $|B_1\rangle, \ldots, |B_{q_1}\rangle$ as:

$$\begin{aligned}
|A_j\rangle &:= \prod_{i=j+1}^{q_1} (U_3^i \circ \mathsf{CStO}) \circ \prod_{i=1}^{j} (U_3^i \circ \chi_0 \circ \mathsf{CStO}) |\psi\rangle |D^\perp\rangle \ (j = 1, \ldots, q_1 - 1), \\
|A_{q_1}\rangle &:= \prod_{i=1}^{q_1} (U_3^i \circ \chi_0 \circ \mathsf{CStO}) |\psi\rangle |D^\perp\rangle = |\Phi\rangle_{0^{q_1}}, \\
|B_1\rangle &:= \prod_{i=2}^{q_1} (U_3^i \circ \mathsf{CStO}) \circ U_3^1 \circ \chi_1 \circ \mathsf{CStO} |\psi\rangle |D^\perp\rangle, \\
|B_j\rangle &:= \prod_{i=j+1}^{q_1} (U_3^i \circ \mathsf{CStO}) \circ U_3^j \circ \chi_1 \circ \mathsf{CStO} \circ \prod_{i=1}^{j-1} (U_3^i \circ \chi_0 \circ \mathsf{CStO}) |\psi\rangle |D^\perp\rangle \ (j = 2, \ldots, q_1 - 1), \\
|B_{q_1}\rangle &:= U_3^{q_1} \circ \chi_1 \circ \mathsf{CStO} \circ \prod_{i=1}^{q_1 - 1} (U_3^i \circ \chi_0 \circ \mathsf{CStO}) |\psi\rangle |D^\perp\rangle.
\end{aligned}$$

For $k = 1, \ldots, q_1 - 2$, we then have

$$\begin{aligned}
|A_k\rangle &= \prod_{i=k+1}^{q_1} (U_3^i \circ \mathsf{CStO}) \circ \prod_{i=1}^{k} (U_3^i \circ \chi_0 \circ \mathsf{CStO}) |\psi\rangle |D^\perp\rangle \\
&= \prod_{i=k+2}^{q_1} (U_3^i \circ \mathsf{CStO}) \circ U_3^{k+1} \circ \mathsf{CStO} \circ \prod_{i=1}^{k} (U_3^i \circ \chi_0 \circ \mathsf{CStO}) |\psi\rangle |D^\perp\rangle \\
&= \prod_{i=k+2}^{q_1} (U_3^i \circ \mathsf{CStO}) \circ U_3^{k+1} \circ \chi_0 \circ \mathsf{CStO} \circ \prod_{i=1}^{k} (U_3^i \circ \chi_0 \circ \mathsf{CStO}) |\psi\rangle |D^\perp\rangle \\
&\quad + \prod_{i=k+2}^{q_1} (U_3^i \circ \mathsf{CStO}) \circ U_3^{k+1} \circ \chi_1 \circ \mathsf{CStO} \circ \prod_{i=1}^{k} (U_3^i \circ \chi_0 \circ \mathsf{CStO}) |\psi\rangle |D^\perp\rangle \\
&= |A_{k+1}\rangle + |B_{k+1}\rangle.
\end{aligned}$$

Note that $|A_{q_1-1}\rangle = |A_{q_1}\rangle + |B_{q_1}\rangle$ by the definition of $|A_{q_1-1}\rangle$, $|A_{q_1}\rangle$ and $|B_{q_1}\rangle$, $|\Psi_1\rangle = |A_1\rangle + |B_1\rangle$ by Eq. (67) and the definition of $|A_1\rangle$ and $|B_1\rangle$, then it is easily to obtain that

$$|\Psi_1\rangle = \sum_{j=1}^{q_1} |B_j\rangle + |A_{q_1}\rangle = \sum_{j=1}^{q_1} |B_j\rangle + |\Phi\rangle_{0^{q_1}}.$$

Thus

$$\begin{aligned}
\||\Psi_1\rangle|0^{q_1}\rangle - |\Psi_2\rangle\|^2 &= \left\| \sum_{j=1}^{q_1} |B_j\rangle |0^{q_1}\rangle + |\Phi\rangle_{0^{q_1}} |0^{q_1}\rangle - \sum_{l_1, \ldots, l_{q_1} \in \{0,1\}^{q_1}} |\Phi\rangle_{l_1, \ldots, l_{q_1}} |l_1, \ldots, l_{q_1}\rangle \right\|^2 \\
&= \left\| \sum_{j=1}^{q_1} |B_j\rangle |0^{q_1}\rangle - \sum_{l_1, \ldots, l_{q_1} \in \{0,1\}^{q_1}}^{l_1, \ldots, l_{q_1} \neq 0^{q_1}} |\Phi\rangle_{l_1, \ldots, l_{q_1}} |l_1, \ldots, l_{q_1}\rangle \right\|^2 \\
&\overset{(b)}{=} \left\| \sum_{j=1}^{q_1} |B_j\rangle \right\|^2 + P_{\text{find}}^{S,z} \overset{(c)}{\leq} q_1 \cdot \sum_{j=1}^{q_1} \||B_j\rangle\|^2 + P_{\text{find}}^{S,z} \\
&= q_1 \cdot \sum_{j=1}^{q_1} b_j + P_{\text{find}}^{S,z} = (q_1 + 1) P_{\text{find}}^{S,z}.
\end{aligned}$$

Here (b) uses the fact that $\||\phi_1\rangle + |\phi_2\rangle\|^2 = \||\phi_1\rangle\|^2 + \||\phi_2\rangle\|^2$ if $|\phi_1\rangle$ and $|\phi_2\rangle$ are orthogonal, (c) uses the Corollary 6.

By [AHU19] Lemma 3 and 4,

$$\begin{aligned}\left|P_{\text{left}}^{S,z} - P_{\text{right}}^{S,z}\right| &= |\Pr[1 \leftarrow \mathcal{B}_1(S,z) : (S,z)] - \Pr[1 \leftarrow \mathcal{B}_2(S,z) : (S,z)]| \\ &\leq \||\Psi_1\rangle|0^{q_1}\rangle - |\Psi_2\rangle\| \leq \sqrt{(q_1+1)P_{\text{find}}^{S,z}}\end{aligned}$$

and

$$\begin{aligned}\left|\sqrt{P_{\text{left}}^{S,z}} - \sqrt{P_{\text{right}}^{S,z}}\right| &= \left|\sqrt{\Pr[1 \leftarrow \mathcal{B}_1(S,z) : (S,z)]} - \sqrt{\Pr[1 \leftarrow \mathcal{B}_2(S,z) : (S,z)]}\right| \\ &\leq \||\Psi_1\rangle|0^{q_1}\rangle - |\Psi_2\rangle\| \leq \sqrt{(q_1+1)P_{\text{find}}^{S,z}}\end{aligned}$$

Note that we only consider a fixed (S,z) in above proof, for random distribution $\mathcal{D}$ of (S,z), the final state of algorithm $\mathcal{B}_1$ (resp. $\mathcal{B}_2$) before measure is a mixed state

$$\rho_1 = \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}[|\Psi_1^{Sz}\rangle|0^{q_1}\rangle\langle\Psi_1^{Sz}|\langle 0^{q_1}|] \ (resp.\ \rho_2 = \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}[|\Psi_2^{Sz}\rangle\langle\Psi_2^{Sz}|]).$$

Here $|\Psi_1^{Sz}\rangle|0^{q_1}\rangle$ is the state $|\Psi_1\rangle|0^{q_1}\rangle$ from Eq. (67) for specific values of S, z, and analogously for $|\Psi_2^{Sz}\rangle$. Then by monotonicity and joint concavity of fidelity (exactly as in [AHU19] Lemma 6 and 9), we have

$$|P_{\text{left}} - P_{\text{right}}| \leq B(\rho_1,\rho_2) \leq \sqrt{(q_1+1)P_{\text{find}}}$$

and

$$\left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq B(\rho_1,\rho_2) \leq \sqrt{(q_1+1)P_{\text{find}}}$$

Here $B(\rho_1,\rho_2)$ is the Bures distance [NC16] between the mixed state ρ_1 and ρ_2.

For the value P_{find}, we compute

$$\begin{aligned}P_{\text{find}} &= \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}} P_{\text{find}}^{S,z} = \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}} \sum_{j=1}^{q_1} b_j \\ &= \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}\left(\sum_{j=2}^{q_1}\left\|U_3^j \circ \chi_1 \circ \mathsf{CStO} \circ \prod_{i=1}^{j-1}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 + \left\|U_3^1 \circ \chi_1 \circ \mathsf{CStO}|\psi\rangle|D^\perp\rangle\right\|^2\right) \\ &= \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}\left(\sum_{j=2}^{q_1}\left\|\chi_1 \circ \mathsf{CStO} \circ \prod_{i=1}^{j-1}(U_3^i \circ \chi_0 \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 + \left\|\chi_1 \circ \mathsf{CStO}|\psi\rangle|D^\perp\rangle\right\|^2\right) \\ &\overset{(c)}{=} \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}\left(\sum_{j=2}^{q_1}\left\|\chi_1 \circ \mathsf{CStO} \circ \prod_{i=1}^{j-1}(\chi_0 \circ U_3^i \circ \mathsf{CStO})|\psi\rangle|D^\perp\rangle\right\|^2 + \left\|\chi_1 \circ \mathsf{CStO} \circ \chi_0|\psi\rangle|D^\perp\rangle\right\|^2\right) \\ &\leq \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}\left(\sum_{j=2}^{q_1}\|\chi_1 \circ \mathsf{CStO} \circ \chi_0\|^2 + \|\chi_1 \circ \mathsf{CStO} \circ \chi_0\|^2\right) = q_1 \cdot \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}\|\chi_1 \circ \mathsf{CStO} \circ \chi_0\|^2 \\ &= q_1 \cdot \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}\|\mathrm{J}_S \circ \mathsf{CStO} \circ (\mathbf{I} - \mathrm{J}_S)\|^2 \overset{(d)}{=} q_1 \cdot \mathop{\mathbb{E}}_{(S,z)\leftarrow\mathcal{D}}\|[\mathrm{J}_S, \mathsf{CStO}]\|^2 .\end{aligned}$$

Here (c) uses the fact that $D^\perp \notin S$ and $U_3^1, \ldots, U_3^{q_1}$ are naturally commute with χ_0[24]. (d) uses the fact that

$$\mathrm{J}_S \circ (\mathbf{I} - \mathrm{J}_S)|\phi\rangle = (\mathbf{I} - \mathrm{J}_S) \circ \mathrm{J}_S|\phi\rangle = \mathbf{0}$$

for any state $|\phi\rangle$. □

[24] Note that $U_3^1, \ldots, U_3^{q_1}$ are the unitary operation that alternatingly applies a unitary operation on registers Q and applies database read operation Read_f, which are both commute with χ_0.

E The Quantum Circuit Implementation of $\mathrm{U}_{\mathsf{test}}$ and $\mathrm{U}_{\mathsf{comp}}$

By the Definition 4, $\mathsf{ota}_1(\mathsf{sk},\cdot)$, $\mathsf{ota}_2(\mathsf{pk},\cdot)$, $\mathsf{ota}_3(\mathsf{pk},\cdot)$ and $\mathsf{ota}_4(\mathsf{pk},\cdot)$ are deterministic algorithm that efficiently computed. Thus, the unitary operation $\mathrm{U}_{\mathsf{ota}_1}$, $\mathrm{U}_{\mathsf{ota}_2}$, $\mathrm{U}_{\mathsf{ota}_3}$ and $\mathrm{U}_{\mathsf{ota}_4}$ defined as follows can be efficiently implemented with quantum circuit by the basic theory of quantum computation.

$$\begin{aligned}
&\mathrm{U}_{\mathsf{f}_{\mathsf{ota}}}|\alpha, y_1\rangle := |\alpha, y_1 \oplus \mathsf{f}_{\mathsf{ota}}(\alpha)\rangle,\ \mathrm{U}_{\mathsf{ota}_1}|\alpha, y_1\rangle := |\alpha, y_1 \oplus \mathsf{ota}_1(\mathsf{sk},\alpha)\rangle,\\
&\mathrm{U}_{\mathsf{ota}_2}|y_1, y_2, y_3\rangle := |y_1, y_2, y_3 \oplus \mathsf{ota}_2(\mathsf{pk}, y_1, y_2)\rangle,\\
&\mathrm{U}_{\mathsf{ota}_3}|\alpha, y_2, y_3\rangle := |y_1, y_2, y_3 \oplus \mathsf{ota}_3(\mathsf{pk}, \alpha, y_2)\rangle,\\
&\mathrm{U}_{\mathsf{ota}_4}|\alpha, y_1, y_2, y_3\rangle := |\alpha, y_1, y_2, y_3 \oplus \mathsf{ota}_4(\mathsf{pk}, \alpha, y_1, y_2)\rangle.
\end{aligned}$$

Then by using unitary operation $\mathrm{U}_{\mathsf{ota}_1}$ and $\mathrm{U}_{\mathsf{ota}_2}$ above, $\mathrm{U}_{\mathsf{test}}$ defined in Eq. (30) with initial state $|\alpha\rangle|0^m\rangle$ on registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}$ can be implemented by the following procedure:

- Initialize register R_1, R_2, R_3 and R_4 to 0, where R_4 is a 1 qubit register.
- Apply $\mathrm{U}_{\mathsf{ota}_1}$ to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{R}_1$, where R_1 is the output register. Then apply the following two conditional operations with controlling register R_1:
  - If the value on register R_1 is $\perp$, apply $\mathrm{U}^{\perp}$ to registers Y, where $\mathrm{U}^{\perp}|0^m\rangle = |\perp\rangle$ and $\mathrm{U}^{\perp}|\perp\rangle = |0^m\rangle$.
  - If the value on register R_1 is not $\perp$:
    * Query random oracle O_1 by registers $\mathsf{R}_1\mathsf{R}_2$, where R_2 is the output register.
    * Apply $\mathrm{U}_{\mathsf{ota}_2}$ to registers $\mathsf{R}_1\mathsf{R}_2\mathsf{R}_3$, where R_3 is the output register.
    * Apply U_1 to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{R}_3\mathsf{R}_4$, where $\mathrm{U}_1|\alpha,\alpha',b\rangle = |\alpha,\alpha',b\oplus 1\rangle$ if $\alpha=\alpha'$, $\mathrm{U}_1|\alpha,\alpha',b\rangle = |\alpha,\alpha',b\rangle$ if $\alpha \neq \alpha'$. Then apply the following two conditional operations with controlling register R_4:
      · If the value on register R_4 is 1, apply CNOT to registers $\mathsf{R}_1\mathsf{Y}$.
      · If the value on register R_4 is 0, apply $\mathrm{U}^{\perp}$ to registers Y, where $\mathrm{U}^{\perp}|0^m\rangle = |\perp\rangle$ and $\mathrm{U}^{\perp}|\perp\rangle = |0^m\rangle$.
    * Apply U_1 to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{R}_3\mathsf{R}_4$ again, where R_4 is the output register.
    * Apply $\mathrm{U}_{\mathsf{ota}_2}$ to registers $\mathsf{R}_1\mathsf{R}_2\mathsf{R}_3$ again, where R_3 is the output register.
    * Query random oracle O_1 by registers $\mathsf{R}_1\mathsf{R}_2$ again, where R_2 is the output register.
- Apply $\mathrm{U}_{\mathsf{ota}_1}$ to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{R}_1$ again, where R_1 is the output register. Now the registers R_1 to R_4 are guaranteed to contain 0, so they can be discarded.

By using unitary operation $\mathrm{U}_{\mathsf{ota}_3}$ and $\mathrm{U}_{\mathsf{ota}_4}$ above, $\mathrm{U}_{\mathsf{comp}}$ defined in Eq. (31) with initial state $|\alpha\rangle|y\rangle|\beta\rangle$ on registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}_{\mathsf{ota}}\mathsf{Y}$ can be implemented by the following procedure:

- Initialize register R_5 and R_6 to 0.
- Apply the following two conditional operations with controlling register Y:
  - If the value on register Y is $\perp$, apply $\mathrm{U}_{\mathsf{f}_{\mathsf{ota}}}$ to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}_{\mathsf{ota}}$, where $\mathsf{Y}_{\mathsf{ota}}$ is the output register.
  - If the value on register Y is not $\perp$:
    * Apply $\mathrm{U}_{\mathsf{ota}_3}$ to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}\mathsf{R}_5$, where R_5 is the output register.
    * Query random oracle O_0 by registers $\mathsf{R}_5\mathsf{R}_6$, where R_6 is the output register.
    * Apply $\mathrm{U}_{\mathsf{ota}_4}$ to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}_{\mathsf{ota}}\mathsf{Y}\mathsf{R}_6$, where $\mathsf{Y}_{\mathsf{ota}}$ is the output register.
    * Query random oracle O_0 by registers $\mathsf{R}_5\mathsf{R}_6$ again, where R_6 is the output register.
    * Apply $\mathrm{U}_{\mathsf{ota}_3}$ to registers $\mathsf{X}_{\mathsf{ota}}\mathsf{Y}\mathsf{R}_5$ again, where R_5 is the output register.
- Now the register R_5 and R_6 is guaranteed to contain 0, so it can be discarded.

We note that the quantum circuit implementation of $\mathrm{U}_{\mathsf{test}}$ and $\mathrm{U}_{\mathsf{comp}}$ need to query random oracle O_1 and random oracle O_0 two times, respectively. Moreover, the quantum circuit implementation of $\mathrm{U}_{\mathsf{comp}}$ does not need the secret key sk.

F Missing Proofs of Section 4

Here we give the detailed proof of some lemmas introduced in Section 4.

F.1 Proof of Lemma 4

Proof. In this proof we first consider a fixed $(\mathsf{pk},\mathsf{sk})$ sampled from KGen. For the adversary $\mathcal{B}$ in game $\mathbf{G_1^q}$ and game $\mathbf{G_2^q}$, the random oracles O_0 and O_1, secret oracle O_{ota} in game $\mathbf{G_1^q}$ and game $\mathbf{G_2^q}$ both are quantum accessed. In addition, the process that the challenger $\mathcal{C}$ get $\mathsf{OHG.A}$ and then return $\mathsf{OHG.B}$ can also be viewed as that the adversary queries a "classical challenge oracle" with input $\mathsf{OHG.A}$ and then get an output $\mathsf{OHG.B}$. Indeed, the "classical challenge oracle" can be easily simulated on quantum superposition since this oracle is implemented by O_0 and O_1 that are quantum simulated. Hence as explained in Appendix A, the game $\mathbf{G_1^q}$ and game $\mathbf{G_2^q}$ can be rewritten as a unitary quantum oracle algorithm and its execution before finally binary measurement can be described as:

$$\mathbf{G_1^q}: |\psi_1\rangle|0^{m'}\rangle_{\mathsf{Y}} := \mathrm{U}_{q_{\mathsf{ota}}}\cdot \mathrm{U}^{1,*}_{\mathsf{ota}}\cdot \mathrm{U}_{q_{\mathsf{ota}}-1}\cdot \mathrm{U}^{1,*}_{\mathsf{ota}}\cdots \mathrm{U}_{\mathsf{OHG.B}}\cdots \mathrm{U}_2\cdot \mathrm{U}^{1}_{\mathsf{ota}}\cdot \mathrm{U}_1\cdot \mathrm{U}^{1}_{\mathsf{ota}}|\psi\rangle|0^{m'}\rangle_{\mathsf{Y}},$$
$$\mathbf{G_2^q}: |\psi_2\rangle|0^{m'}\rangle_{\mathsf{Y}} := \mathrm{U}_{q_{\mathsf{ota}}}\cdot \mathrm{U}^{2,*}_{\mathsf{ota}}\cdot \mathrm{U}_{q_{\mathsf{ota}}-1}\cdot \mathrm{U}^{2,*}_{\mathsf{ota}}\cdots \mathrm{U}_{\mathsf{OHG.B}}\cdots \mathrm{U}_2\cdot \mathrm{U}^{2}_{\mathsf{ota}}\cdot \mathrm{U}_1\cdot \mathrm{U}^{2}_{\mathsf{ota}}|\psi\rangle|0^{m'}\rangle_{\mathsf{Y}},$$

Here $|\psi_1\rangle|0^{m'}\rangle_{\mathsf{Y}}$ and $|\psi_2\rangle|0^{m'}\rangle_{\mathsf{Y}}$ are final states of game $\mathbf{G_1^q}$ and game $\mathbf{G_2^q}$, respectively, $|\psi\rangle|0^{m'}\rangle_{\mathsf{Y}}$ is the initial state of these two games. Register Y is the internal register used by $\mathrm{U}^1_{\mathsf{ota}}$, $\mathrm{U}^{1,*}_{\mathsf{ota}}$, $\mathrm{U}^2_{\mathsf{ota}}$ and $\mathrm{U}^{1,*}_{\mathsf{ota}}$, it always in state $|0^{m'}\rangle$ before and after once application of these unitary operations. $\mathrm{U}_1,\ldots,\mathrm{U}_{q_{\mathsf{ota}}}$ are the unitary operations applied between the queries to secret oracle O_{ota}. $\mathrm{U}_{\mathsf{OHG.B}}$ is the unitary that simulates the "classical challenge oracle", and the $\mathrm{U}^1_{\mathsf{ota}}$ (resp, $\mathrm{U}^2_{\mathsf{ota}}$) is replaced to $\mathrm{U}^{1,*}_{\mathsf{ota}}$ (resp, $\mathrm{U}^{2,*}_{\mathsf{ota}}$) after the application of $\mathrm{U}_{\mathsf{OHG.B}}$.

For any state $|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}$ on the whole quantum register of game $\mathbf{G_1^q}$ and game $\mathbf{G_2^q}$ before the application of $\mathrm{U}^1_{\mathsf{ota}}$, $\mathrm{U}^{1,*}_{\mathsf{ota}}$, $\mathrm{U}^2_{\mathsf{ota}}$ and $\mathrm{U}^{1,*}_{\mathsf{ota}}$ as

$$|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}} := \sum_{z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1}} \alpha_{z,D,x,y}|z,D,x,y\rangle_{\mathsf{Z}\mathsf{D}_{q_1}\mathsf{X}_{\mathsf{ota}}\mathsf{Y}_{\mathsf{ota}}}|0^{m'}\rangle_{\mathsf{Y}},$$

where $\mathsf{X}_{\mathsf{ota}}/\mathsf{Y}_{\mathsf{ota}}$ is the input/output register of secret oracle O_{ota}, D_{q_1} is the database register and the other registers are abbreviated into register Z, by the analysis of F.1.1, we have

$$\max\left\{\|(\mathrm{U}^1_{\mathsf{ota}}-\mathrm{U}^2_{\mathsf{ota}})|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\|, \|(\mathrm{U}^{1,*}_{\mathsf{ota}}-\mathrm{U}^{2,*}_{\mathsf{ota}})|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\|\right\} \le 8\cdot\sqrt{\max_{\alpha\in\mathcal{X},\beta\in\{0,1\}^{m'}}\frac{\left|\mathsf{ota.sub}^{x,y'}_{\mathsf{pk}}\right|}{2^{n'}}}.$$

By the hybrid argument, the final state $|\psi_1\rangle|0^{m'}\rangle_{\mathsf{Y}}$ and $|\psi_2\rangle|0^{m'}\rangle_{\mathsf{Y}}$ satisfy

$$\||\psi_1\rangle|0^{m'}\rangle_{\mathsf{Y}} - |\psi_2\rangle|0^{m'}\rangle_{\mathsf{Y}}\| \le 8q_{\mathsf{ota}}\cdot\sqrt{\max_{\alpha\in\mathcal{X},\beta\in\{0,1\}^{m'}}\frac{\left|\mathsf{ota.sub}^{x,y'}_{\mathsf{pk}}\right|}{2^{n'}}}.$$

Then by [AHU19] Lemma 3 and 4,

$$|\Pr[1\leftarrow\mathbf{G_1^q}:(\mathsf{pk},\mathsf{sk})] - \Pr[1\leftarrow\mathbf{G_2^q}:(\mathsf{pk},\mathsf{sk})]| \le 8q_{\mathsf{ota}}\cdot\sqrt{\max_{\alpha\in\mathcal{X},\beta\in\{0,1\}^{m'}}\frac{\left|\mathsf{ota.sub}^{x,y'}_{\mathsf{pk}}\right|}{2^{n'}}}.$$

Averaging over $(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KGen}(1^\lambda)$ and using the Jensen's inequality, we finally obtain

$$\begin{aligned}|\Pr[1\leftarrow\mathbf{G_1^q}] - \Pr[1\leftarrow\mathbf{G_2^q}]| &\le 8q_{\mathsf{ota}}\cdot\sqrt{\mathop{\mathbb{E}}_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KGen}(1^\lambda)}\max_{\alpha\in\mathcal{X},\beta\in\{0,1\}^{m'}}\frac{\left|\mathsf{ota.sub}^{x,y'}_{\mathsf{pk}}\right|}{2^{n'}}}\\ &\overset{(a)}{=} 8q_{\mathsf{ota}}\cdot\sqrt{\mathsf{ota.max}}.\end{aligned}$$

Here (a) uses Eq. (15). $\square$

F.1.1 Bound on $\|(\mathrm{U}^1_{\mathsf{ota}} - \mathrm{U}^2_{\mathsf{ota}})|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\|$ and $\|(\mathrm{U}^{1,*}_{\mathsf{ota}} - \mathrm{U}^{2,*}_{\mathsf{ota}})|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\|$

For the sake of convenience, we abbreviate $|z, D, x, y\rangle_{\mathsf{ZD}_{q_1}\mathsf{X}_{\mathsf{ota}}\mathsf{Y}_{\mathsf{ota}}}|0^{m'}\rangle_{\mathsf{Y}}$ into $|z, D, x, y, 0^{m'}\rangle$ in the following. Now we separate $|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}$ into four mutual orthogonal parts $|\phi_1\rangle$ to $|\phi_4\rangle$ that $|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}} = \sum_{i=1}^{4}|\phi_i\rangle$, where $|\phi_1\rangle$ to $|\phi_4\rangle$ are the following states:

$$
\begin{aligned}
|\phi_1\rangle &= \sum_{\substack{z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}\\ x\in\mathcal{X}, y\in\{0,1\}^{l+1}, \mathsf{ota}_1(\mathsf{sk},x)=\perp}} \alpha_{z,D,x,y}|z, D, x, y, 0^{m'}\rangle,\\
|\phi_2\rangle &= \sum_{\substack{z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1}\\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp}} \alpha_{z,D,x,y}|z, D, x, y, 0^{m'}\rangle,\\
|\phi_3\rangle &= \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n\\ z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1}\\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \alpha_{z,D,x,y,r}|z, D\cup(y',\hat{r}), x, y, 0^{m'}\rangle,\\
|\phi_4\rangle &= \sum_{\substack{r=0^n\\ z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1}\\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \alpha_{z,D,x,y,r}|z, D\cup(y',\hat{r}), x, y, 0^{m'}\rangle.
\end{aligned}
$$

Here we default the database D in each basis state of $|\phi_2\rangle$ also satisfies $n(D) < q_1$, which is unproblematic since the query times of random oracle O_1 in game $\mathbf{G^q_1}$ and $\mathbf{G^q_2}$ both is at most q_1 times.

Denote $\Delta := \mathrm{U}^1_{\mathsf{ota}} - \mathrm{U}^2_{\mathsf{ota}}$, by the definition of $\mathrm{U}^1_{\mathsf{ota}}$ and $\mathrm{U}^2_{\mathsf{ota}}$ and the quantum circuit implementation of $\mathrm{U}_{\mathsf{test}}$ and $\mathrm{U}_{\mathsf{comp}}$ given in Appendix E, we compute[25]:

$$
\begin{aligned}
&\Delta|\phi_1\rangle = \mathbf{0},\\
&\Delta|\phi_2\rangle = \sum_{\substack{z'\in\mathsf{ota.sub}^{x,y'}_{\mathsf{pk}}\\ z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1}\\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp}} \frac{\alpha_{z,D,x,y}}{\sqrt{2^{n'}}}\cdot\mathsf{StdDecomp}_x\begin{pmatrix} |z, D\cup(y',z'), x, y\oplus\mathsf{ota}_4(\mathsf{pk},x,y',O_0(y'')), 0^{m'}\rangle\\ -|z, D\cup(y',z'), x, y\oplus\perp, 0^{m'}\rangle\end{pmatrix},\\
&\Delta|\phi_3\rangle = \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n, z'\in\mathsf{ota.sub}^{x,y'}_{\mathsf{pk}}\\ z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1}\\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{2^{n'}}\begin{pmatrix} |z, D, x, y\oplus\mathsf{ota}_3(\mathsf{pk},x,y',O_0(y')), 0^{m'}\rangle\\ -|z, D\cup(y',\hat{0^n}), x, y\oplus\mathsf{ota}_4(\mathsf{pk},x,y',O_0(y'')), 0^{m'}\rangle\\ -|z, D, x, y\oplus\perp, 0^{m'}\rangle\\ +|z, D\cup(y',\hat{0^n}), x, y\oplus\perp, 0^{m'}\rangle\end{pmatrix}.
\end{aligned}
\tag{69}
$$

Here $y'' := \mathsf{ota}_3(pk, x, y')$.

As for the $\Delta|\phi_4\rangle$, we find that the state with the form of $|z, D\cup(y',\hat{0^n}), x, y, 0^{m'}\rangle$ is illegal [Zha19] and it can not appear just before the application of $\mathrm{U}^1_{\mathsf{ota}}$ and $\mathrm{U}^{1,*}_{\mathsf{ota}}$ in game $\mathbf{G^q_1}$[26]. Hence we add a complement of the operation of $\mathrm{U}^1_{\mathsf{ota}}$ as

$$
\mathrm{U}^1_{\mathsf{ota}}|z, D\cup(y',\hat{0^n}), x, y, 0^{m'}\rangle := |z, D\cup(y',\hat{0^n}), x, y\oplus\perp, 0^{m'}\rangle,
$$

which is easily to implement since the state $|z, D\cup(y',\hat{0^n}), x, y, 0^{m'}\rangle$ must be orthogonal with $|\phi_1\rangle$, $|\phi_2\rangle$ and $|\phi_3\rangle$. With this complement, we have

$$
\Delta|\phi_4\rangle = \sum_{\substack{r=0^n, z'\in\mathsf{ota.sub}^{x,y'}_{\mathsf{pk}}\\ z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1}\\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \frac{\alpha_{z,D,x,y,r}}{\sqrt{2^{n'}}}\begin{pmatrix} |z, D\cup(y',z'), x, y\oplus\perp, 0^{m'}\rangle\\ -|z, D\cup(y',z'), x, y\oplus\mathsf{ota}_4(\mathsf{pk},x,y',O_0(y'')), 0^{m'}\rangle\end{pmatrix}.
\tag{70}
$$

[25] Since the quantum circuit implementation of $\mathrm{U}_{\mathsf{test}}$ and $\mathrm{U}_{\mathsf{comp}}$ given in Appendix E is not very simple, the detailed computational process of $\Delta|\phi_1\rangle$ to $\Delta|\phi_4\rangle$ are complicated and we omit it. Nevertheless, we stress that, following the quantum circuit implementation of $\mathrm{U}_{\mathsf{test}}$ and $\mathrm{U}_{\mathsf{comp}}$, one can get $\Delta|\phi_1\rangle$ to $\Delta|\phi_4\rangle$ shown in Eq. (69) and Eq. (70) by directly compute.

[26] However, the state with the form of $|z, D\cup(y',\hat{0^n}), x, y, 0^{m'}\rangle$ can appear in game $\mathbf{G^q_2}$ since the extraction-interface $\mathsf{eCO.E}_{f_1}$ is applied.

Here $y'' := \mathsf{ota}_3(pk, x, y')$.

Then we can compute

$$
\begin{aligned}
&\| \Delta | \phi_2 \rangle \|^2 \\
&= \left\| \sum_{\substack{z' \in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z \in \{0,1\}^*, D \in \mathbf{D}_{q_1}, x \in \mathcal{X}, y \in \{0,1\}^{l+1} \\ y' := \mathsf{ota}_1(\mathsf{sk}, x) \neq \perp, D(y') = \perp}} \frac{\alpha_{z,D,x,y}}{\sqrt{2^{n'}}} \cdot \mathsf{StdDecomp}_x \begin{pmatrix} |z, D \cup (y', z'), x, y \oplus \mathsf{ota}_3(\mathsf{pk}, x, y', O_0(y')), 0^{m'} \rangle \\ -|z, D \cup (y', z'), x, y \oplus \perp, 0^{m'} \rangle \end{pmatrix} \right\|^2 \\
&\overset{(a)}{=} \left\| \sum_{\substack{z' \in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z \in \{0,1\}^*, D \in \mathbf{D}_{q_1}, x \in \mathcal{X}, y \in \{0,1\}^{l+1} \\ y' := \mathsf{ota}_1(\mathsf{sk}, x) \neq \perp, D(y') = \perp}} \frac{\alpha_{z,D,x,y}}{\sqrt{2^{n'}}} \begin{pmatrix} |z, D \cup (y', z'), x, y \oplus \mathsf{ota}_3(\mathsf{pk}, x, y', O_0(y')), 0^{m'} \rangle \\ -|z, D \cup (y', z'), x, y \oplus \perp, 0^{m'} \rangle \end{pmatrix} \right\|^2 \\
&\overset{(b)}{\leq} 2 \cdot \left\| \sum_{\substack{z' \in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z \in \{0,1\}^*, D \in \mathbf{D}_{q_1}, x \in \mathcal{X}, y \in \{0,1\}^{l+1} \\ y' := \mathsf{ota}_1(\mathsf{sk}, x) \neq \perp, D(y') = \perp}} \frac{\alpha_{z,D,x,y}}{\sqrt{2^{n'}}} |z, D \cup (y', z'), x, y \oplus \mathsf{ota}_3(\mathsf{pk}, x, y', O_0(y')), 0^{m'} \rangle) \right\|^2 \\
&\quad + 2 \cdot \left\| \sum_{\substack{z' \in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z \in \{0,1\}^*, D \in \mathbf{D}_{q_1}, x \in \mathcal{X}, y \in \{0,1\}^{l+1} \\ y' := \mathsf{ota}_1(\mathsf{sk}, x) \neq \perp, D(y') = \perp}} \frac{\alpha_{z,D,x,y}}{\sqrt{2^{n'}}} |z, D \cup (y', z'), x, y \oplus \perp, 0^{m'} \rangle \right\|^2 \\
&= 4 \cdot \sum_{\substack{z' \in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z \in \{0,1\}^*, D \in \mathbf{D}_{q_1}, x \in \mathcal{X}, y \in \{0,1\}^{l+1} \\ y' := \mathsf{ota}_1(\mathsf{sk}, x) \neq \perp, D(y') = \perp}} \left| \frac{\alpha_{z,D,x,y}}{\sqrt{2^{n'}}} \right|^2 \\
&\leq 4 \cdot \max_{x \in \mathcal{X}, y' \in \{0,1\}^{m'}} \frac{\left| \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \right|}{2^{n'}} \cdot \sum_{\substack{z \in \{0,1\}^*, D \in \mathbf{D}_{q_1}, x \in \mathcal{X}, y \in \{0,1\}^{l+1} \\ y' := \mathsf{ota}_1(\mathsf{sk}, x) \neq \perp, D(y') = \perp}} |\alpha_{z,D,x,y}|^2 \\
&= 4 \cdot \max_{x \in \mathcal{X}, y' \in \{0,1\}^{m'}} \frac{\left| \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \right|}{2^{n'}} \cdot \| | \phi_2 \rangle \|^2.
\end{aligned}
\tag{71}
$$

Here (a) uses the fact that $\mathsf{StdDecomp}_x$ is a unitary operation, (b) uses Corollary 6. Similar with the computation of $\| \Delta | \phi_2 \rangle \|^2$, we also have

$$
\| \Delta | \phi_4 \rangle \|^2 \leq 4 \cdot \max_{x \in \mathcal{X}, y' \in \{0,1\}^{m'}} \frac{\left| \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \right|}{2^{n'}} \cdot \| | \phi_4 \rangle \|^2. \tag{72}
$$

For the $\Delta|\phi_3\rangle$, we can compute

$$
\begin{aligned}
&\|\Delta|\phi_3\rangle\|^2 \\
&= \left\| \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n, z'\in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z\in\{0,1\}^*, D\in \mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{2^{n'}} \begin{pmatrix} |z,D,x,y\oplus \mathsf{ota}_3(\mathsf{pk},x,y',O_0(y')),0^{m'}\rangle \\ -|z,D\cup(y',\hat{0^n}),x,y\oplus \mathsf{ota}_3(\mathsf{pk},x,y',O_0(y')),0^{m'}\rangle \\ -|z,D,x,y\oplus\perp,0^{m'}\rangle \\ +|z,D\cup(y',\hat{0^n}),x,y\oplus\perp,0^{m'}\rangle \end{pmatrix} \right\|^2 \\
&\overset{(c)}{\leq} 4\cdot \left\| \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n, z'\in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z\in\{0,1\}^*, D\in \mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{2^{n'}} |z,D,x,y\oplus \mathsf{ota}_3(\mathsf{pk},x,y',O_0(y')),0^{m'}\rangle \right\|^2 \\
&\quad + 4\cdot \left\| \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n, z'\in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z\in\{0,1\}^*, D\in \mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{2^{n'}} |z,D\cup(y',\hat{0^n}),x,y\oplus \mathsf{ota}_3(\mathsf{pk},x,y',O_0(y')),0^{m'}\rangle \right\|^2 \\
&\quad + 4\cdot \left\| \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n, z'\in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z\in\{0,1\}^*, D\in \mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{2^{n'}} |z,D,x,y\oplus\perp,0^{m'}\rangle \right\|^2 \\
&\quad + 4\cdot \left\| \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n, z'\in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'} \\ z\in\{0,1\}^*, D\in \mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{2^{n'}} |z,D\cup(y',\hat{0^n}),x,y\oplus\perp,0^{m'}\rangle \right\|^2 \\
&= 16\cdot \sum_{\substack{z\in\{0,1\}^*, D\in \mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \left| \sum_{r\in\{0,1\}^n, r\neq 0^n, z'\in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{2^{n'}} \right|^2 \\
&\overset{(d)}{\leq} 16\cdot \sum_{\substack{z\in\{0,1\}^*, D\in \mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \sum_{z'\in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}} \left| \sum_{r\in\{0,1\}^n, r\neq 0^n} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{\sqrt{2^{n'}}} \right|^2 \\
&\leq 16\cdot \max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}} \sum_{\substack{z\in\{0,1\}^*, D\in \mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\perp, D(y')=\perp, n(D)<q_1}} \sum_{z'\in \mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}} \left| \sum_{r\in\{0,1\}^n, r\neq 0^n} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{\sqrt{2^{n'}}} \right|^2 .
\end{aligned}
\tag{73}
$$

Here (c) uses Corollary 6 again, (d) uses the Cauchy-Schwarz inequality.

In addition, we have

$$\begin{aligned}
\||\phi_3\rangle\|^2 &= \left\| \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n \\ z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\bot, D(y')=\bot, n(D)<q_1}} \alpha_{z,D,x,y,r}|z, D\cup(y',\hat{r}), x, y, 0^{m'}\rangle \right\|^2 \\
&= \left\| \sum_{\substack{r\in\{0,1\}^n, r\neq 0^n \\ z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\bot, D(y')=\bot, n(D)<q_1}} \sum_{z'\in\{0,1\}^n} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{\sqrt{2^{n'}}}|z, D\cup(y',z'), x, y, 0^{m'}\rangle \right\|^2 \\
&= \sum_{\substack{z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\bot, D(y')=\bot, n(D)<q_1}} \sum_{z'\in\{0,1\}^n} \left| \sum_{r\in\{0,1\}^n, r\neq 0^n} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{\sqrt{2^{n'}}} \right|^2 \\
&\geq \sum_{\substack{z\in\{0,1\}^*, D\in\mathbf{D}_{q_1}, x\in\mathcal{X}, y\in\{0,1\}^{l+1} \\ y':=\mathsf{ota}_1(\mathsf{sk},x)\neq\bot, D(y')=\bot, n(D)<q_1}} \sum_{z'\in\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}} \left| \sum_{r\in\{0,1\}^n, r\neq 0^n} \frac{(-1)^{z'\cdot r}\alpha_{z,D,x,y,r}}{\sqrt{2^{n'}}} \right|^2.
\end{aligned}$$

Combining above inequality with Eq. (73), we get

$$\|\Delta|\phi_3\rangle\|^2 \leq 16 \cdot \max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}} \cdot \||\phi_3\rangle\|^2. \tag{74}$$

Combining Eq. (71), (72) and (74), we obtain

$$\begin{aligned}
&\|(\mathrm{U}_{\mathsf{ota}}^1 - \mathrm{U}_{\mathsf{ota}}^2)|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\| \\
&\overset{(e)}{\leq} 2\sqrt{\max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}}}(\||\phi_2\rangle\| + \||\phi_4\rangle\|) + 4\sqrt{\max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}}}\||\phi_3\rangle\| \\
&\overset{(f)}{\leq} 8\sqrt{\max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}}}\||\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\rangle\| = 8\sqrt{\max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}}}.
\end{aligned} \tag{75}$$

Here (e) uses the fact that $|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}} = \sum_{i=1}^4 |\phi_i\rangle$, (f) uses the fact that $|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}} = \sum_{i=1}^4 |\phi_i\rangle$ and $|\phi_1\rangle$ to $|\phi_4\rangle$ are mutual orthogonal.

As for $\|(\mathrm{U}_{\mathsf{ota}}^{1,*} - \mathrm{U}_{\mathsf{ota}}^{2,*})|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\|$, note that $\mathrm{U}_{\mathsf{ota}}^{1,*} := \mathrm{U}_{\perp} \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}_{\mathsf{ota}}^1 \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})$ and $\mathrm{U}_{\mathsf{ota}}^{2,*} := \mathrm{U}_{\perp} \circ \mathrm{P}_{\mathsf{hide}} + \mathrm{U}_{\mathsf{ota}}^2 \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})$, thus

$$\begin{aligned}
\|(\mathrm{U}_{\mathsf{ota}}^{1,*} - \mathrm{U}_{\mathsf{ota}}^{2,*})|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\| &= \|(\mathrm{U}_{\mathsf{ota}}^1 - \mathrm{U}_{\mathsf{ota}}^2) \circ (\mathbf{I} - \mathrm{P}_{\mathsf{hide}})|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\| \\
&\overset{(g)}{\leq} 8\sqrt{\max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}}}\|(\mathbf{I} - \mathrm{P}_{\mathsf{hide}})|\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\rangle\| \\
&\leq 8\sqrt{\max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}}}.
\end{aligned}$$

Here (g) uses the fact that

$$\|(\mathrm{U}_{\mathsf{ota}}^1 - \mathrm{U}_{\mathsf{ota}}^2)|\phi\rangle|0^{m}\rangle_{\mathsf{Y}}\| \leq 8\sqrt{\max_{x\in\mathcal{X}, y'\in\{0,1\}^{m'}} \frac{\left|\mathsf{ota.sub}_{\mathsf{pk}}^{x,y'}\right|}{2^{n'}}}\||\phi\rangle|0^{m'}\rangle_{\mathsf{Y}}\rangle\|,$$

which is implied by the (e) and (f) of Eq. (75).

F.2 Proof of Lemma 5

Proof. Based on game $\mathbf{G_2^q}$ and game $\mathbf{G_3^q}$, we introduce two new games as follows:

Game $\mathbf{G_{2a}^q}$: This game is identical with game $\mathbf{G_2^q}$ except that the compressed semi-classical oracle $\mathcal{O}_S^{CSC}$ is queried just after each invoking of the RO-interface $\mathsf{eCO.RO}$.

Game $\mathbf{G_{3a}^q}$: This game is identical with game $\mathbf{G_3^q}$ except that the compressed semi-classical oracle $\mathcal{O}_S^{CSC}$ is queried just after each invoking of the RO-interface $\mathsf{eCO.RO}$.

In game $\mathbf{G_2^q}$, the random oracle O_1 is simulated by invoking the RO-interface $\mathsf{eCO.RO}$ directly, and the simulation of secret oracle O_{ota} uses the extraction-interface $\mathsf{eCO.E}_{f_1}$. Hence, we can rewrite game $\mathbf{G_2^q}$ as a quantum oracle algorithm $\mathcal{B}^{O_1,\mathsf{eCO.E}_{f_1}}$ with input $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$ that makes at most q_1 times queries to random oracle O_1. Then

$$\begin{aligned}
&\Pr[1 \leftarrow \mathbf{G_2^q}] = \Pr[1 \leftarrow \mathcal{B}^{O_1,\mathsf{eCO.E}_{f_1}}(\mathsf{pk},\mathsf{sk}) : (S,\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{D}],\\
&\Pr[1 \leftarrow \mathbf{G_{2a}^q}] = \Pr[1 \leftarrow \mathcal{B}^{O_1\backslash S,\mathsf{eCO.E}_{f_1}}(\mathsf{pk},\mathsf{sk}) : (S,\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{D}],\\
&\Pr[1 \leftarrow \mathbf{G_3^q}] = \Pr[1 \leftarrow \mathcal{B}^{O_1,\mathsf{eCO.E}_{f_2}}(\mathsf{pk},\mathsf{sk}) : (S,\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{D}],\\
&\Pr[1 \leftarrow \mathbf{G_{3a}^q}] = \Pr[1 \leftarrow \mathcal{B}^{O_1\backslash S,\mathsf{eCO.E}_{f_2}}(\mathsf{pk},\mathsf{sk}) : (S,\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{D}].
\end{aligned}$$

Here $\mathcal{D}$ is a joint distribution that $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$, set $S \subseteq \mathbf{D}_{q_1}$ defined in Eq. (34) is determined by $(\mathsf{pk},\mathsf{sk})$ since $\mathsf{ota}_1(\mathsf{sk},\cdot)$ and $\mathsf{ota}_2(\mathsf{pk},\cdot)$ are deterministic algorithms.

As explained in Section 2.5, the extraction-interface $\mathsf{eCO.E}_f$ for any function f is processed by a database read operation Ext_f. Thus, by using Theorem 3, we have

$$|\Pr[1 \leftarrow \mathbf{G_2^q}] - \Pr[1 \leftarrow \mathbf{G_{2a}^q}]| \le \sqrt{q_1(q_1+1) \cdot \mathop{\mathbb{E}}_{(S,\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{D}} \|[\mathrm{J}_S, \mathsf{CStO}]\|^2}, \tag{76}$$

and

$$|\Pr[1 \leftarrow \mathbf{G_3^q}] - \Pr[1 \leftarrow \mathbf{G_{3a}^q}]| \le \sqrt{q_1(q_1+1) \cdot \mathop{\mathbb{E}}_{(S,\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{D}} \|[\mathrm{J}_S, \mathsf{CStO}]\|^2}. \tag{77}$$

Note that $\mathsf{eCO.E}_{f_1}$ and $\mathsf{eCO.E}_{f_2}$ proceed identically for any input state $|\alpha, 0^{m'}, D\rangle$ if $D \notin S$, hence algorithm $\mathcal{B}^{O_1\backslash S,\mathsf{eCO.E}_{f_1}}(\mathsf{pk},\mathsf{sk})$ and $\mathcal{B}^{O_1\backslash S,\mathsf{eCO.E}_{f_2}}(\mathsf{pk},\mathsf{sk})$ proceed identically if the compressed semi-classical oracle $\mathcal{O}_S^{CSC}$ never returns 1. This implies that for

$$\begin{aligned}
&\Pr[\mathsf{Find} \text{ occurs in } \mathcal{B}^{O_1\backslash S,\mathsf{eCO.E}_{f_1}}(\mathsf{pk},\mathsf{sk}) : (S,\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{D}]\\
&\qquad = \Pr[\mathsf{Find} \text{ occurs in } \mathcal{B}^{O_1\backslash S,\mathsf{eCO.E}_{f_2}}(\mathsf{pk},\mathsf{sk}) : (S,\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{D}],
\end{aligned}$$

$$\begin{aligned}
|\Pr[1 \leftarrow \mathbf{G_{2a}^q}] - \Pr[1 \leftarrow \mathbf{G_{3a}^q}]| &\le \Pr[\mathsf{Find} \text{ occurs in } \mathcal{B}^{O_1\backslash S,\mathsf{eCO.E}_{f_2}}(\mathsf{pk},\mathsf{sk}) : (S,\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{D}]\\
&\overset{(a)}{\le} q_1 \cdot \mathop{\mathbb{E}}_{(S,\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{D}} \|[\mathrm{J}_S, \mathsf{CStO}]\|^2
\end{aligned} \tag{78}$$

Here (a) uses Theorem 3. Then by combining Eq. (76), (77) and (78), we obtain

$$|\Pr[1 \leftarrow \mathbf{G_2^q}] - \Pr[1 \leftarrow \mathbf{G_3^q}]| \le \sqrt{q_1(q_1+1) \cdot \mathop{\mathbb{E}}_{(S,\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{D}} \|[\mathrm{J}_S, \mathsf{CStO}]\|^2} + q_1 \cdot \mathop{\mathbb{E}}_{(S,\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{D}} \|[\mathrm{J}_S, \mathsf{CStO}]\|^2. \tag{79}$$

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

$$g(x,y) = \begin{cases} 1 & \text{if } \mathsf{ota}_2(\mathsf{pk},x,y) = z \wedge \mathsf{ota}_1(\mathsf{sk},z) \neq x \\ 0 & \text{otherwise.} \end{cases}$$

For function g, the corresponding relation R_1^g and parameter $\Gamma_{R_1^g}$ defined in Eq. (11) is

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

$$\Gamma_{R_1^g} := \max_{x\in\{0,1\}^{m'}} |\{y \in \{0,1\}^{n'} : \mathsf{ota}_2(\mathsf{pk}, x, y) = z \wedge \mathsf{ota}_1(\mathsf{sk}, z) \neq x\}| \overset{(b)}{\leq} \max_{x\in\{0,1\}^{m'}} \left| \bigcup_{z\in\mathsf{Set}.x} \mathsf{ota.sub}_{\mathsf{pk}}^{z,x} \right|. \tag{80}$$

Here (b) is hold since one can easily check that if $y \in \{y \in \{0,1\}^{n'} : \mathsf{ota}_2(\mathsf{pk}, x, y) = z \wedge \mathsf{ota}_1(\mathsf{sk}, z) \neq x\}$ then y must belong to $\bigcup_{z\in\mathsf{Set}.x} \mathsf{ota.sub}_{\mathsf{pk}}^{z,x}$ by the definition of $\mathsf{Set}.x$ and $\mathsf{ota.sub}_{\mathsf{pk}}^{z,x}$ defined in Definition 4.

For the relation R_1^g, define following projectors act on database register D_{q_1}:

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

By the definition of set $S \subseteq \mathbf{D}_{q_1}$ defined in (34), it is obvious that $\mathrm{J}_S = \sum_{x\in\{0,1\}^{m'}} \Sigma^x$, and then $\Sigma^\perp = \mathbf{I} - \mathrm{J}_S$. Hence we have

$$\|[\mathrm{J}_S, \mathsf{CStO}]\| \overset{(c)}{=} \|[\mathbf{I} - \mathrm{J}_S, \mathsf{CStO}]\| = \left\|\left[\Sigma^\perp, \mathsf{CStO}\right]\right\| \overset{(d)}{\leq} 8 \cdot \sqrt{\Gamma_{R_1^g}/2^n}. \tag{81}$$

Here (c) uses the basic property of the commutator, (d) uses the Lemma 2.

Combining Eq. (79), (80) and (81), we finally obtain

$$\begin{aligned}|\Pr[1 \leftarrow \mathbf{G_2^q}] - \Pr[1 \leftarrow \mathbf{G_3^q}]| \leq 8 \cdot & \sqrt{q_1(q_1+1) \cdot \mathop{\mathbb{E}}_{(S,\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{D}} \frac{1}{2^{n'}} \max_{x\in\{0,1\}^{m'}} \left| \bigcup_{z\in\mathsf{Set}.x} \mathsf{ota.sub}_{\mathsf{pk}}^{z,x} \right|} \\ & + 64q_1 \cdot \mathop{\mathbb{E}}_{(S,\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{D}} \frac{1}{2^{n'}} \max_{x\in\{0,1\}^{m'}} \left| \bigcup_{z\in\mathsf{Set}.x} \mathsf{ota.sub}_{\mathsf{pk}}^{z,x} \right| \\ \overset{(e)}{=} & \ 8 \cdot \sqrt{q_1(q_1+1) \cdot \mathsf{ota.union}} + 64q_1 \cdot \mathsf{ota.union}\end{aligned}$$

Here (e) uses Eq. (15). □

G Cryptographic Primitives

Definition 6 (Public key encryption)**.** *A public key encryption (PKE) scheme consist of a finite message space $\mathcal{M}$ and three polynomial algorithm* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *according to security parameter λ.*

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

Definition 7 (Correctness [HHK17])**.** *A PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct if*

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

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}$. We call a pair (m, c) is "error" pair if $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m$. Denote

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

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

Definition 8 (weakly γ-spread [DFMS22])**.** *A PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is weakly* γ*-spread if*

$$-\log \underset{(sk,pk)\leftarrow Gen}{\mathbb{E}} \left[\max_{m \in \mathcal{M}, c \in \mathcal{C}} \Pr[c = \mathrm{Enc}_{pk}(m)] \right] \geq \gamma,$$

where the probability is over the randomness of the encryption.

Definition 9 (Security notions for PKE)**.** *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme. For any adversary* $\mathcal{A}$ *and* $\mathsf{GOAL\text{-}ATK} \in \{\mathsf{IND\text{-}qCCA}, \mathsf{WPR\text{-}qCCA}, \mathsf{ANO\text{-}qCCA}, \mathsf{SDS\text{-}IND}\}$*, we define its* $\mathsf{GOAL\text{-}ATK}$ *advantage against* PKE *as follows:*

$$\mathsf{Adv}^{\mathsf{GOAL\text{-}ATK}}_{\mathcal{A},(\mathcal{S}),\mathsf{PKE}}(1^\lambda) := \left| \Pr[1 \leftarrow \mathsf{Game}^{\mathsf{GOAL\text{-}ATK}}_{\mathcal{A},(\mathcal{S}),\mathsf{PKE}}(1^\lambda)] - \frac{1}{2} \right|,$$

where $\mathsf{Game}^{\mathsf{GOAL\text{-}ATK}}_{\mathcal{A},\mathsf{PKE}}(1^\lambda)$ *is a game described in Fig. 11.*

For any adversary $\mathcal{A}$*, we define its* $\mathsf{OW\text{-}CPA}$ *advantage against* PKE *as follows:*

$$\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(1^\lambda) := \Pr[1 \leftarrow \mathsf{Game}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(1^\lambda)],$$

where $\mathsf{Game}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(1^\lambda)$ *is a game described in Fig. 11. For*

$$\mathsf{GOAL\text{-}ATK} \in \{\mathsf{IND\text{-}qCCA}, \mathsf{WPR\text{-}qCCA}, \mathsf{ANO\text{-}qCCA}, \mathsf{SDS\text{-}IND}, \mathsf{OW\text{-}CPA}\},$$

we say that PKE *is* $\mathsf{GOAL\text{-}ATK}$*-secure if* $\mathsf{Adv}^{\mathsf{GOAL\text{-}ATK}}_{\mathcal{A},(\mathcal{S}),\mathsf{PKE}}(1^\lambda)$ *is negligible for any QPT adversary* $\mathcal{A}$*.*

Definition 10 (Key-encapsulation mechanism)**.** *A key-encapsulation mechanism (KEM) consists of three algorithms* Gen*,* Enca *and* Deca*. The key generation algorithm* Gen *outputs a key pair* (pk, sk)*. The encapsulation algorithm* Enca*, on input* pk*, outputs a tuple* (K, c) *where* c *is said to be an encapsulation of the key* K *which is contained in key space* $\mathcal{K}$*. The deterministic decapsulation algorithm* Deca*, on input* sk *and an encapsulation* c*, outputs either a key* $K := \mathsf{Deca}(sk, c) \in \mathcal{K}$ *or a special symbol* $\perp \notin \mathcal{K}$ *to indicate that* c *is not a valid encapsulation.*

Definition 11 (Security notions for KEM)**.** *Let* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Enca}, \mathsf{Deca})$ *be a KEM scheme. For any adversary* $\mathcal{A}$ *and* $\mathsf{GOAL\text{-}ATK} \in \{\mathsf{IND\text{-}qCCA}, \mathsf{SPR\text{-}qCCA}, \mathsf{ANO\text{-}qCCA}\}$*, we define its* $\mathsf{GOAL\text{-}ATK}$ *advantage against* KEM *as follows:*

$$\mathsf{Adv}^{\mathsf{GOAL\text{-}ATK}}_{\mathcal{A},(\mathcal{S}),\mathsf{KEM}}(1^\lambda) := \left| \Pr[1 \leftarrow \mathsf{Game}^{\mathsf{GOAL\text{-}ATK}}_{\mathcal{A},(\mathcal{S}),\mathsf{KEM}}(1^\lambda)] - \frac{1}{2} \right|,$$

where $\mathsf{Game}^{\mathsf{GOAL\text{-}ATK}}_{\mathcal{A},\mathsf{KEM}}(1^\lambda)$ *is a game described in Fig. 11. For* $\mathsf{GOAL\text{-}ATK} \in \{\mathsf{IND\text{-}qCCA}, \mathsf{SPR\text{-}qCCA}, \mathsf{ANO\text{-}qCCA}\}$*, we say that* KEM *is* $\mathsf{GOAL\text{-}ATK}$*-secure if* $\mathsf{Adv}^{\mathsf{GOAL\text{-}ATK}}_{\mathcal{A},(\mathcal{S}),\mathsf{KEM}}(1^\lambda)$ *is negligible for any QPT adversary* $\mathcal{A}$*.*

Definition 12 (Data-encapsulation mechanism)**.** *A data-encapsulation mechanism (DEM) consist of a finite message space* $\mathcal{M}$ *and two polynomial algorithm* E, D *according to security parameter* λ*.*

1. E*: a encapsulation algorithm with input a message* $m \in \mathcal{M}$ *and key* $k \leftarrow \mathcal{K}$*(*$\mathcal{K}$ *is the key space), computes* $c := E(k, m)$ *and output ciphertext* c*.*
2. D*: a decapsulation algorithm with input a ciphertext* c *and key* k*, computes* $m := D(k, c)$ *and output* m *or a rejection symbol* $\perp \notin \mathcal{M}$*.*

Definition 13 (OT secure DEM)**.** *A DEM scheme* $\mathsf{DEM} = (E, D)$ *is* OT *secure if for any quantum polynomial adversary* $\mathcal{A}$*, the probability of* $\mathcal{A}$ *wins in game* $\mathrm{Game}^{\mathsf{OT}}_{\mathcal{A},\mathsf{DEM}}(1^\lambda)$ *is* $1/2 + negl$*, where* $negl$ *is negligible.*

$\underline{\mathrm{Game}^{\mathsf{OT}}_{\mathcal{A},\mathsf{DEM}}(1^\lambda)}$*:*

1. *Query: The adversary* $\mathcal{A}$ *choose two message* m_0, m_1 *of same length on it's input* 1^λ*, then send* m_0, m_1 *to challenger. The challenger choose* $b \xleftarrow{\$} \{0,1\}$ *and respond with* $c = \mathsf{E}(k, m_b)$
2. *Guess:* $\mathcal{A}$ *produce a guess* b'*, if* $b' = b$*,* $\mathcal{A}$ *wins.*

$$\begin{array}{l}\underline{\text{Game } \mathsf{Game}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(1^\lambda)}\\ 1,\ (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)\\ 2,\ m^* \leftarrow \mathcal{M}\\ \quad c^* := \mathsf{Enc}(pk, m^*)\\ 3,\ m' \leftarrow \mathcal{A}(pk, c^*)\\ 4,\ \textbf{Return } \mathsf{boole}[m^* = m']\end{array}$$

$$\begin{array}{l}\underline{\mathsf{oDec}(c)}\\ 1,\ \textbf{Return } \mathsf{Dec}(sk, c)\end{array}$$

$$\begin{array}{l}\underline{\mathsf{oDec}_a(c)}\\ 1,\ \textbf{If } c = a, \textbf{ return } \bot\\ \quad \textbf{Else return } \mathsf{Dec}(sk, c)\end{array}$$

$$\begin{array}{l}\underline{\text{Game } \mathsf{Game}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{PKE}}(1^\lambda)}\\ 1,\ (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)\\ 2,\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{oDec}}(pk, c^*)\\ 3,\ b \xleftarrow{\$} \{0,1\}\\ \quad c^* := \mathsf{Enc}(pk, m_b)\\ 4,\ b' \leftarrow \mathcal{A}^{\mathsf{oDec}_{c^*}}(pk, c^*)\\ 5,\ \textbf{Return } \mathsf{boole}[b = b']\end{array}$$

$$\begin{array}{l}\underline{\text{Game } \mathsf{Game}^{\mathsf{ANO\text{-}qCCA}}_{\mathcal{A},\mathsf{PKE}}(1^\lambda)}\\ 1,\ (pk_0, sk_0) \leftarrow \mathsf{Gen}(1^\lambda)\\ \quad (pk_1, sk_1) \leftarrow \mathsf{Gen}(1^\lambda)\\ 2,\ m^* \leftarrow \mathcal{A}^{\mathsf{oDec}'(\cdot,\cdot)}(pk_0, pk_1)\\ 3,\ b \xleftarrow{\$} \{0,1\}\\ \quad c^* := \mathsf{Enc}(pk_b, m^*)\\ 4,\ b' \leftarrow \mathcal{A}^{\mathsf{oDec}'_{c^*}(\cdot,\cdot)}(pk_0, pk_1, c^*)\\ 5,\ \textbf{Return } \mathsf{boole}[b = b']\end{array}$$

$$\begin{array}{l}\underline{\mathsf{oDec}'(b,\cdot)}\\ 1,\ \textbf{Return } \mathsf{Dec}(sk_b, c)\\ \underline{\mathsf{oDec}'_a(b,\cdot)}\\ 1,\ \textbf{If } c = a, \textbf{ return } \bot\\ \quad \textbf{Else return } \mathsf{Dec}(sk_b, c)\end{array}$$

$$\begin{array}{l}\underline{\text{Game } \mathsf{Game}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S},\mathsf{PKE}}(1^\lambda)}\\ 1,\ (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)\\ 2,\ m^* \leftarrow \mathcal{A}^{\mathsf{oDec}}(pk)\\ 3,\ b \xleftarrow{\$} \{0,1\}\\ \quad c_0^* := \mathsf{Enc}(pk, m^*)\\ \quad c_1^* \leftarrow \mathcal{S}(1^\lambda, m^*)\\ 4,\ b' \leftarrow \mathcal{A}^{\mathsf{oDec}_{c_b^*}}(pk, c_b^*)\\ 5,\ \textbf{Return } \mathsf{boole}[b = b']\end{array}$$

$$\begin{array}{l}\underline{\text{Game } \mathsf{Game}^{\mathsf{SDS\text{-}IND}}_{\mathcal{A},\mathcal{S},\mathsf{PKE}}(1^\lambda)}\\ 1,\ (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)\\ 2,\ m^* \leftarrow \mathcal{M},\ b \xleftarrow{\$} \{0,1\}\\ \quad c_0^* := \mathsf{Enc}(pk, m^*)\\ \quad c_1^* \leftarrow \mathcal{S}(1^\lambda)\\ 3,\ b' \leftarrow \mathcal{A}(pk, c_b^*)\\ 4,\ \textbf{Return } \mathsf{boole}[b = b']\end{array}$$

$$\begin{array}{l}\underline{\text{Game } \mathsf{Game}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}}(1^\lambda)}\\ 1,\ (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)\\ 2,\ b \xleftarrow{\$} \{0,1\}\\ \quad (c^*, K_0^*) := \mathsf{Enca}(pk)\\ \quad K_1^* \xleftarrow{\$} \mathcal{K}\\ 3,\ b' \leftarrow \mathcal{A}^{\mathsf{oDeca}_{c^*}}(pk, c^*, K_b^*)\\ 4,\ \textbf{Return } \mathsf{boole}[b = b']\end{array}$$

$$\begin{array}{l}\underline{\text{Game } \mathsf{Game}^{\mathsf{SPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S},\mathsf{KEM}}(1^\lambda)}\\ 1,\ (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)\\ 2,\ b \xleftarrow{\$} \{0,1\}\\ \quad (c_0^*, K_0^*) := \mathsf{Enca}(pk)\\ \quad (c_1^*, K_1^*) \leftarrow \mathcal{S}(1^\lambda) \times \mathcal{K}\\ 3,\ b' \leftarrow \mathcal{A}^{\mathsf{oDeca}_{c_b^*}}(pk, c_b^*, K_b^*)\\ 4,\ \textbf{Return } \mathsf{boole}[b = b']\end{array}$$

$$\begin{array}{l}\underline{\text{Game } \mathsf{Game}^{\mathsf{ANO\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}}(1^\lambda)}\\ 1,\ (pk_0, sk_0) \leftarrow \mathsf{Gen}(1^\lambda)\\ \quad (pk_1, sk_1) \leftarrow \mathsf{Gen}(1^\lambda)\\ 2,\ b \xleftarrow{\$} \{0,1\}\\ \quad (c^*, K^*) := \mathsf{Enca}(pk_b)\\ 3,\ b' \leftarrow \mathcal{A}^{\mathsf{oDeca}'_{c^*}(\cdot,\cdot)}(pk_0, pk_1, c^*)\\ 4,\ \textbf{Return } \mathsf{boole}[b = b']\end{array}$$

$$\begin{array}{l}\underline{\mathsf{oDeca}_a(c)}\\ 1,\ \textbf{If } c = a, \textbf{ return } \bot\\ \quad \textbf{Else return } \mathsf{Deca}(c)\\ \underline{\mathsf{oDeca}'_a(b,\cdot)}\\ 1,\ \textbf{If } c = a, \textbf{ return } \bot\\ \quad \textbf{Else return } \mathsf{Deca}(sk_b, c)\end{array}$$

Figure 11: Games for PKE and KEM schemes. In game $\mathsf{Game}^{\mathsf{GOAL\text{-}qCCA}}_{\mathcal{A},(\mathcal{S}),\mathsf{PKE}}(1^\lambda)$ and $\mathsf{Game}^{\mathsf{GOAL\text{-}qCCA}}_{\mathcal{A},(\mathcal{S}),\mathsf{KEM}}(1^\lambda)$ the adversary $\mathcal{A}$ can query its oracles in superposition.

H Missing proofs of Section 5

H.1 Proof of Theorem 5

Proof. Denote $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Let us define four games as shown in Fig. 12, according to the definition of ANO-qCCA security given in Appendix G, it is obvious that

$$|\Pr[1 \leftarrow \mathbf{G_1}] - \Pr[1 \leftarrow \mathbf{G_2}]| = 2 \cdot \mathsf{Adv}^{\mathsf{ANO\text{-}qCCA}}_{\mathcal{A},\Pi},\ \Pr[1 \leftarrow \mathbf{G_3}] = \Pr[1 \leftarrow \mathbf{G_4}]. \tag{82}$$

Game $\mathbf{G_1}$	Game $\mathbf{G_2}$	Game $\mathbf{G_3}$	Game $\mathbf{G_4}$
1: $(pk_0, sk_0) \leftarrow \mathsf{Gen}$	1: $(pk_0, sk_0) \leftarrow \mathsf{Gen}$	1: $(pk_0, sk_0) \leftarrow \mathsf{Gen}$	1: $(pk_0, sk_0) \leftarrow \mathsf{Gen}$
$(pk_1, sk_1) \leftarrow \mathsf{Gen}$	$(pk_1, sk_1) \leftarrow \mathsf{Gen}$	$(pk_1, sk_1) \leftarrow \mathsf{Gen}$	$(pk_1, sk_1) \leftarrow \mathsf{Gen}$
2: $m^* \leftarrow \mathcal{A}^{\mathsf{Dec}(\cdot,\cdot)}(pk_0, pk_1)$	2: $m^* \leftarrow \mathcal{A}^{\mathsf{Dec}(\cdot,\cdot)}(pk_0, pk_1)$	2: $m^* \leftarrow \mathcal{A}^{\mathsf{Dec}(\cdot,\cdot)}(pk_0, pk_1)$	2: $m^* \leftarrow \mathcal{A}^{\mathsf{Dec}(\cdot,\cdot)}(pk_0, pk_1)$
3: $b = 0$	3: $b = 1$	3: $b = 0$	3: $b = 1$
$c_0^* := \mathsf{Enc}(pk_0, m^*)$	$c_1^* := \mathsf{Enc}(pk_1, m^*)$	$c_0^* := \mathcal{S}(1^\lambda, m^*)$	$c_1^* := \mathcal{S}(1^\lambda, m^*)$
4: $b' \leftarrow \mathcal{A}^{\mathsf{Dec}_{c_0^*}(\cdot,\cdot)}(pk_0, pk_1, c_0^*)$	4: $b' \leftarrow \mathcal{A}^{\mathsf{Dec}_{c_1^*}(\cdot,\cdot)}(pk_0, pk_1, c_1^*)$	4: $b' \leftarrow \mathcal{A}^{\mathsf{Dec}_{c_0^*}(\cdot,\cdot)}(pk_0, pk_1, c_0^*)$	4: $b' \leftarrow \mathcal{A}^{\mathsf{Dec}_{c_1^*}(\cdot,\cdot)}(pk_0, pk_1, c_1^*)$
5: **Return** b'	5: **Return** b'	5: **Return** b'	5: **Return** b'

Figure 12: Game $\mathbf{G_1}$ to $\mathbf{G_4}$. Here $\mathsf{Dec}(\cdot,\cdot)$ return $\mathsf{Dec}(sk_b, c)$ for input (b, c), $\mathsf{Dec}_a(\cdot,\cdot)$ is identical with $\mathsf{Dec}(\cdot,\cdot)$ except that Dec_a output $\perp$ for input $(0, a)$ and $(1, a)$. The adversary in these four games both can query its oracles in superposition.

Then we define an adversary $\mathcal{B}_1$ against the WPR-qCCA security of PKE as follows:

1. After get the pk from the challenger, sample a new (pk', sk') pair by using Gen, then runs adversary $\mathcal{A}(pk, pk')$ to get m^* and send it to the challenger. The decryption oracle query $\sum_{t\in\{0,1\},c\in\mathcal{C},y\in\{0,1\}^*} |t, c, y\rangle$ performed by $\mathcal{A}$ is answered as:
   - For each basis state $|t, c, y\rangle$, query decryption oracle $\mathsf{Dec}(sk, \cdot)$ if $t = 0$. Else, compute and return $|t, c, y \oplus \mathsf{Dec}(sk', c)\rangle$. Here decryption oracle $\mathsf{Dec}(sk, \cdot)$ is the oracle $\mathcal{B}$ can access in the WPR-qCCA game.
2. After get the c_b^* from the challenger, runs $\mathcal{A}(pk, pk', c_b^*)$ to get output b' and send b' to the challenger. The decryption oracle query $\sum_{t\in\{0,1\},c\in\mathcal{C},y\in\{0,1\}^*} |t, c, y\rangle$ performed by $\mathcal{A}$ is answered as:
   - For each basis state $|t, c, y\rangle$, if $c = c_b^*$, return $|t, c, y \oplus \perp\rangle$. Else if $t = 0$, query decryption oracle $\mathsf{Dec}(sk, \cdot)$. Else, compute and return $|t, c, y \oplus \mathsf{Dec}(sk', c)$.

We also define an adversary $\mathcal{B}_2$, which is identical with $\mathcal{B}_1$ except that the decryption oracle query $\sum_{t\in\{0,1\},c\in\mathcal{C},y\in\{0,1\}^*} |t, c, y\rangle$ performed by $\mathcal{A}$ is instead answered as:

- c_b^* has not yet been obtained: For each basis state $|t, c, y\rangle$, query decryption oracle $\mathsf{Dec}(sk, \cdot)$ if $t = 1$. Otherwise, compute and return $|t, c, y \oplus \mathsf{Dec}(sk', c)\rangle$.
- c_b^* has been obtained: For each basis state $|t, c, y\rangle$, if $c = c_b^*$, return $|t, c, y \oplus \perp\rangle$. Else if $t = 1$, query decryption oracle $\mathsf{Dec}(sk, \cdot)$. Else, compute and return $|t, c, y \oplus \mathsf{Dec}(sk', c)$.

One can easily check that

$$|\Pr[1 \leftarrow \mathbf{G_1}] - \Pr[1 \leftarrow \mathbf{G_4}]| = 2 \cdot \mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{B}_1,\mathcal{S},\Pi},\ \Pr[1 \leftarrow \mathbf{G_2}] - \Pr[1 \leftarrow \mathbf{G_3}] = 2 \cdot \mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{B}_2,\mathcal{S},\Pi}. \tag{83}$$

Combing Eq. (82) and (83), we have

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ANO\text{-}qCCA}}_{\Pi,\mathcal{A}} &= |\Pr[1 \leftarrow \mathbf{G_1}] - \Pr[1 \leftarrow \mathbf{G_2}]|/2 \\
&\leq |\Pr[1 \leftarrow \mathbf{G_1}] - \Pr[1 \leftarrow \mathbf{G_4}]|/2 + |\Pr[1 \leftarrow \mathbf{G_2}] - \Pr[1 \leftarrow \mathbf{G_3}]|/2 \\
&= \mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{B}_1,\mathcal{S},\Pi} + \mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{B}_2,\mathcal{S},\Pi} \overset{(a)}{\leq} 2 \cdot \mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{B},\mathcal{S},\Pi}.
\end{aligned}$$

Here (a) is obtained by folding $\mathcal{B}_1$ and $\mathcal{B}_2$ into one single adversary $\mathcal{B}$. $\square$

H.2 The IND-qCCA security of $\mathsf{KEM}^{\perp}$, $\mathsf{KEM}_m^{\not\perp}$ and $\mathsf{KEM}^{\not\perp}$ in the QROM

Theorem 9. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct and weakly γ-spread. Let $\mathcal{A}$ be an* IND-qCCA *adversary against* $\mathsf{KEM}^{\perp}$ *in the QROM, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist an* OW-CPA *adversary $\mathcal{A}_1$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}^{\perp}} \leq 40q_D \cdot \sqrt{\gamma} + 8(q_H+1) \cdot \sqrt{\delta} + 64q_H \cdot \delta + 4(q_H + q_G + 1) \cdot \sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}}}.$$

The running time of $\mathcal{A}_1$ can be bounded as $\mathrm{Time}[\mathcal{A}_1] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_C \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2)$.

Proof. Compared with $\mathsf{KEM}_m^{\perp}$, the only difference in $\mathsf{KEM}^{\perp}$ is that the key K in $\mathsf{KEM}^{\perp}$ is derived from message m and ciphertext c, not just from the message m like $\mathsf{KEM}_m^{\perp}$. Therefore, the decapsulation algorithm $\mathsf{Deca}^{\perp}(sk,\cdot)$ of $\mathsf{KEM}^{\perp}$ can also be written as an oracle-testing algorithm like the decapsulation algorithm $\mathsf{Deca}_m^{\perp}(sk,\cdot)$ of $\mathsf{KEM}_m^{\perp}$, and thus the proof of Theorem 6 is also valid for Theorem 9, as long as we correspondingly modify the definition of algorithm $\mathsf{dec}_1(pk,\cdot)$ and challenger $\mathcal{C}_{\mathsf{dec}}$ in the proof of Theorem 6. □

Indeed, the IND-qCCA security reductions of $\mathsf{KEM}_m^{\not\perp}$ and $\mathsf{KEM}^{\not\perp}$ in the QROM are similar to that of $\mathsf{KEM}_m^{\perp}$ and $\mathsf{KEM}^{\perp}$, respectively. It should be noted that the reductions of $\mathsf{KEM}_m^{\not\perp}$ and $\mathsf{KEM}^{\not\perp}$ need to first transform the pseudorandom functions used in the decapsulation algorithm into uniform random functions. The security loss generated after above transition can be bounded by using the Lemma 2 in [JZC+18]. Here, we directly give the theorem states that $\mathsf{KEM}_m^{\not\perp}$ and $\mathsf{KEM}^{\not\perp}$ are IND-qCCA security in the QROM and omit the proofs.

Theorem 10. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a randomized PKE that is δ-correct and weakly γ-spread. Let $\mathcal{A}$ be an* IND-qCCA *adversary (in the QROM) against* $\mathsf{KEM}_m^{\not\perp}$*, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist an adversary $\mathcal{A}'$ against the security of* PRF *with at most q_D (quantum) queries and an* OW-CPA *adversary $\mathcal{A}_1$ against the* PKE *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}_m^{\not\perp}} \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}'} &+ 40q_D \cdot \sqrt{\gamma} + 8(q_H+1) \cdot \sqrt{\delta} + 64q_H \cdot \delta \\ &+ 4(q_H + q_G + 1) \cdot \sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}}}.\end{aligned}$$

Then the running time of $\mathcal{A}'$ and $\mathcal{A}_1$ can be bounded as

$$\mathrm{Time}[\mathcal{A}'] \approx \mathrm{Time}[\mathcal{A}],\ \mathrm{Time}[\mathcal{A}_1] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_C \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Theorem 11. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a randomized PKE that is δ-correct and weakly γ-spread. Let $\mathcal{A}$ be an* IND-qCCA *adversary (in the QROM) against* $\mathsf{KEM}^{\not\perp}$*, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist an* OW-CPA *adversary $\mathcal{A}_1$ against the* PKE *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathcal{A},\mathsf{KEM}_m^{\not\perp}} \leq 2q_H \cdot \frac{1}{\sqrt{2^u}} &+ 40q_D \cdot \sqrt{\gamma} + 8(q_H+1) \cdot \sqrt{\delta} + 64q_H \cdot \delta \\ &+ 4(q_H + q_G + 1) \cdot \sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}}}.\end{aligned}$$

Then the running time of $\mathcal{A}_1$ can be bounded as

$$\mathrm{Time}[\mathcal{A}_1] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_C \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

H.3 Proof of Lemma 7

Proof. Our proof idea is simple, first rewrite game $\mathbf{G}_{\mathcal{A}}^{b=0}$ and game $\mathbf{G}_{\mathcal{A}}^{b=1}$ as an oracle-hiding game in the QROM, then apply the Theorem 4 to obtain the adversary $\mathcal{B}$ and adversary $\mathcal{A}_1$.

Based on the (pk, sk) generated by the Gen, define four deterministic algorithms $\mathsf{dec}_1(sk,\cdot)$ to $\mathsf{dec}_4(pk,\cdot)$ as follows (Here we omit the input space for simplify.):

Game $\mathbf{G}_{\mathcal{A}}^{b=0}$

1: $(pk, sk) \leftarrow \mathsf{Gen}$, $b = 0$
2: $m^* \xleftarrow{\$} \{0,1\}^u$
$c_0^* := \mathsf{Enc}(pk, m^*, H(m^*))$
$K_0^* := G(m^*)$
3: $b' \leftarrow \mathcal{A}^{H,G,O_{\mathsf{dec}}^{c_0^*}}(pk, c_0^*, K_0^*)$
4: **Return** b'

$O_{\mathsf{dec}}^{c_0^*}(c)$

1: **If** $c = c_0^*$, **return** $\perp$
Else return $\mathsf{Deca}_m^{\perp}(c)$

$O_{\mathsf{dec}}^{c_1^*}(c)$

1: **If** $c = c_1^*$, **return** $\perp$
Else return $\mathsf{Deca}_m^{\perp}(c)$

Game $\mathbf{G}_{\mathcal{A}}^{b=1}$

1: $(pk, sk) \leftarrow \mathsf{Gen}$, $b = 1$
2: $m^* \xleftarrow{\$} \{0,1\}^u$
$c_1^* := \mathcal{S}(1^\lambda)$
$K_1^* \xleftarrow{\$} \{0,1\}^k$
3: $b' \leftarrow \mathcal{A}^{H,G,O_{\mathsf{dec}}^{c_1^*}}(pk, c_1^*, K_1^*)$
4: **Return** b'

Figure 13: Game $\mathbf{G}_{\mathcal{A}}^{b=0}$ and game $\mathbf{G}_{\mathcal{A}}^{b=1}$. Here adversary $\mathcal{A}$ can query its oracles in superposition.

- $\mathsf{dec}_1(sk, \cdot)$: For input x, return $\perp$ if $\mathsf{Dec}(sk, x) = \perp$. Otherwise, return $\mathsf{Dec}(sk, x)$.
- $\mathsf{dec}_2(pk, \cdot)$: For input (x, y), return $\mathsf{Enc}(pk, x, y)$.
- $\mathsf{dec}_3(pk, \cdot)$: For input (x, y), return y.
- $\mathsf{dec}_4(pk, \cdot)$: For input (x, y, z), return z.

Define $\mathsf{f}_{\mathsf{dec}}$ be a function that $\mathsf{f}_{\mathsf{dec}}(x) = \perp$ for any x, then the decapsulation algorithm $\mathsf{Deca}_m^{\perp}$ shown in Fig. 5 can be rewritten as the following oracle algorithm $\mathsf{dec}^{G,H}(sk, \cdot)$:

1. For the input c, compute $\beta := \mathsf{dec}_1(sk, c)$. If $\beta := \perp$, return $\mathsf{f}_{\mathsf{dec}}(c)$.
2. Else comute $\mathsf{dec}_2(pk, \beta, H(\beta))$. If $\mathsf{dec}_2(pk, \beta, H(\beta)) \neq c$, return $\mathsf{f}_{\mathsf{dec}}(c)$.
   - Else compute $\gamma := \mathsf{dec}_3(pk, c, \beta)$, return $\mathsf{dec}_4(pk, c, \beta, G(\beta))$.

According to the definition of the oracle-testing algorithm in Definition 4, it is obvious that oracle algorithm $\mathsf{dec}^{G,H}(sk, \cdot)$ is an oracle-testing algorithm. In Table 4, we provide a detailed correspondence between the basic components (e.g. the internal algorithms) of oracle algorithm $\mathsf{dec}^{G,H}(sk, \cdot)$ and oracle-testing algorithm $\mathsf{ota}^{O_0,O_1}(\mathsf{sk}, \cdot)$ introduced in Definition 4.

Table 4: The correspondence between the basic components of $\mathsf{ota}^{O_0,O_1}(\mathsf{sk}, \cdot)$ and $\mathsf{dec}^{G,H}(sk, \cdot)$.

	Key generator	Random oracle	function	Internal algorithms
$\mathsf{ota}^{O_0,O_1}(\mathsf{sk}, \cdot)$	$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$	O_0/O_1	$\mathsf{f}_{\mathsf{ota}}$	$\mathsf{ota}_1(\mathsf{sk}, \cdot)/\mathsf{ota}_2(\mathsf{pk}, \cdot)/\mathsf{ota}_3(\mathsf{pk}, \cdot)/\mathsf{ota}_4(\mathsf{pk}, \cdot)$
$\mathsf{dec}^{G,H}(sk, \cdot)$	$(pk, sk) \leftarrow \mathsf{Gen}$	G/H	$\mathsf{f}_{\mathsf{dec}}$	$\mathsf{dec}_1(sk, \cdot)/\mathsf{dec}_2(pk, \cdot)/\mathsf{dec}_3(pk, \cdot)/\mathsf{dec}_4(pk, \cdot)$

The corresponding parameters $\mathsf{dec.time}$, $\mathsf{dec.max}$ and $\mathsf{dec.union}$ of oracle-testing algorithm $\mathsf{dec}^{G,H}(sk, \cdot)$ defined in Eq. (15) can be written as:

$$\begin{aligned}
&\mathsf{dec.time} = \mathrm{Time}[\mathsf{dec}_2] + \mathrm{Time}[\mathsf{dec}_3] + \mathrm{Time}[\mathsf{dec}_4] \approx \mathrm{Time}[\mathsf{Enc}],\\
&\mathsf{dec.max} = \frac{1}{2^v} \mathop{\mathbb{E}}_{(pk,sk) \leftarrow \mathsf{Gen}} \max_{c \in \mathcal{C}, m \in \mathcal{M}} \left|\{r \in \{0,1\}^v : \mathsf{Enc}(pk, m, r) = c\}\right|,\\
&\mathsf{dec.union} = \frac{1}{2^v} \mathop{\mathbb{E}}_{(pk,sk) \leftarrow \mathsf{Gen}} \max_{m \in \mathcal{M}} \left| \bigcup_{c \in \{c \in \mathcal{C} : \mathsf{Dec}(sk, c) \neq m\}} \{r \in \{0,1\}^v : \mathsf{Enc}(pk, m, r) = c\} \right|.
\end{aligned} \tag{84}$$

Since the PKE scheme PKE is δ-correct and weakly γ-spread, one can obtain the following inequality immediately by combing Eq. (84) with the definition of δ-correct and weakly γ-spread given in Appendix G.

$$\mathsf{dec.max} \leq \gamma, \ \mathsf{dec.union} \leq \delta. \tag{85}$$

Based on the oracle-testing algorithm $\mathsf{dec}^{G,H}(sk, \cdot)$, we define an oracle-hiding game $\mathsf{OHG}_{\mathcal{A}_{\mathsf{dec}}, \mathcal{C}_{\mathsf{dec}}}^{G,H,O_{\mathsf{dec}}}$ in the QROM as shown in Fig. 14, where $\mathcal{A}_{\mathsf{dec}}$ and $\mathcal{C}_{\mathsf{dec}}$ satisfies following properties:

- Without any computations, $\mathcal{A}_{\mathsf{dec}}$ directly generates $\mathsf{OHG.A}$ as $\perp$.

- $\mathsf{cha}_1(pk,\cdot)$ and $\mathsf{cha}_2(pk,\cdot)$ performed by $\mathcal{C}_{\mathsf{dec}}$ return $\varnothing$ for any input, where $\varnothing$ satisfies $x||\varnothing := x$ for any x.
- $\mathsf{cha}_3(pk,\cdot)$ performed by $\mathcal{C}_{\mathsf{dec}}$ generates $\mathsf{OHG.B}$ as $\mathsf{Enc}(pk, m^*, y_1)$.
- $\mathcal{A}_{\mathsf{dec}}$ then runs $\mathcal{A}$ in game $\mathbf{G}_{\mathcal{A}}^{b=0}$[27], return the output b' of $\mathcal{A}$ as $\mathsf{OHG.C}$.
- The algorithm $\mathsf{verify}(pk, sk, \cdot)$ performed by $\mathcal{C}_{\mathsf{dec}}$ directly return b'.

Game $\mathsf{OHG}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}^{G,H,O_{\mathsf{dec}}}$

1, $(pk, sk) \leftarrow \mathsf{Gen}$

2, $\perp \leftarrow \mathcal{A}_{\mathsf{dec}}(pk)$

3, $\mathcal{C}_{\mathsf{dec}}$ **perform following operation**

  $m^* \xleftarrow{\$} \{0,1\}^u$, $r \xleftarrow{\$} \{0,1\}^u$, $s = 0$

  $\varnothing \leftarrow \mathsf{cha}_1(pk, \perp, m^*, r)$

  $y_0 = G(m^*)$

  $\varnothing \leftarrow \mathsf{cha}_2(pk, \perp, y_0, m^*, r)$

  $y_1 = H(m^*)$

  $\mathsf{Enc}(pk, m^*, y_1) \leftarrow \mathsf{cha}_3(pk, \perp, y_0, y_1, m^*, r)$

4, $b' \leftarrow \mathcal{A}^{H,G,O_{\mathsf{dec}}}(pk, \mathsf{Enc}(pk, m^*, y_1))$

5, $b' \leftarrow \mathsf{verify}(pk, sk, \perp, m^*, r, s, b')$

  $\mathcal{C}_{\mathsf{dec}}$ **output** b' **as game's output**

$G(x)$

1, $O \xleftarrow{\$} \mathcal{F}_{*,k}$, **return** $O(x)$

$H(x)$

1, $O' \xleftarrow{\$} \mathcal{F}_{u,v}$, **return** $O'(x)$

$O_{\mathsf{dec}}(c)$

1, **If** $\mathsf{Enc}(pk, m^*, y_1)$ **is defined**

  and $c = \mathsf{Enc}(pk, m^*, y_1)$

   return $\perp$

  Else return $\mathsf{dec}^{G,H}(sk, c)$

Figure 14: The oracle-hiding game $\mathsf{OHG}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}^{G,H,O_{\mathsf{dec}}}$ in the QROM.

It is easy to see that

$$\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] = \mathsf{Adv}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}^{\mathsf{OHG}}(1^\lambda). \tag{86}$$

Then by using Theorem 4 and Eq. (84) and (85), there exists adversaries $\mathcal{A}_{\mathsf{dec}}^1$ and $\mathcal{A}_{\mathsf{dec}}^2$ do not query the oracle it can access that satisfy

$$\left|\mathsf{Adv}_{\mathcal{A}_{\mathsf{dec}},\mathcal{C}_{\mathsf{dec}}}^{\mathsf{OHG}}(1^\lambda) - \mathsf{Adv}_{\mathcal{A}_{\mathsf{dec}}^1,\mathcal{C}_{\mathsf{dec}}}^{\mathsf{OHG}}(1^\lambda)\right| \le 40q_D \cdot \sqrt{\gamma} + 8(q_H + q_G + 1)\cdot\sqrt{\delta} + 64q_H \cdot \delta + 4(q_H + q_G + 1)\cdot\sqrt{\mathsf{Adv}_{\mathcal{A}_{\mathsf{dec}}^2,\mathcal{C}_{\mathsf{dec}}^{\mathsf{find}}}^{\mathsf{OHG}}(1^\lambda)}, \tag{87}$$

and

$$\mathrm{Time}[\mathcal{A}_{\mathsf{dec}}^1] \approx \mathrm{Time}[\mathcal{A}_{\mathsf{dec}}^2] \le \mathrm{Time}[\mathcal{A}_{\mathsf{dec}}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2). \tag{88}$$

Here $\mathcal{C}_{\mathsf{dec}}^{\mathsf{find}}$ is the same as $\mathcal{C}_{\mathsf{dec}}$ except that the algorithm verify used by $\mathcal{C}_{\mathsf{dec}}^{\mathsf{find}}$ output $\mathsf{boole}[m^* = \mathsf{OHG.C}]$.

For the $\mathsf{Adv}_{\mathcal{A}_{\mathsf{dec}}^1,\mathcal{C}_{\mathsf{dec}}}^{\mathsf{OHG}}(1^\lambda)$, since $\mathcal{A}_{\mathsf{dec}}^1$ only invokes adversary $\mathcal{A}_{\mathsf{dec}}$ in a black-box manner, it is obvious that there exists an adversary $\mathcal{B}$ does not query the oracle it can access satisfy

$$\Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}] = \mathsf{Adv}_{\mathcal{A}_{\mathsf{dec}}^1,\mathcal{C}_{\mathsf{dec}}}^{\mathsf{OHG}}(1^\lambda),\ \mathrm{Time}[\mathcal{B}] = \mathrm{Time}[\mathcal{A}_{\mathsf{dec}}^1]. \tag{89}$$

As for the $\mathsf{Adv}_{\mathcal{A}_{\mathsf{dec}}^2,\mathcal{C}_{\mathsf{dec}}^{\mathsf{find}}}^{\mathsf{OHG}}(1^\lambda)$, since the adversary $\mathcal{A}_{\mathsf{dec}}^2$ do not query any oracle it can access, the value y_1 used by challenger $\mathcal{C}_{\mathsf{dec}}^{\mathsf{find}}$ is uniformly random in the views of $\mathcal{A}_{\mathsf{dec}}^2$ in oracle-hiding game $\mathsf{OHG}_{\mathcal{A}_{\mathsf{dec}}^2,\mathcal{C}_{\mathsf{dec}}}^{G,H,O_{\mathsf{dec}}}$. Hence, it is easy to see that there exist an $\mathsf{OW\text{-}CPA}$ adversaries $\mathcal{A}_1$ against the underlying PKE scheme PKE such that

$$\mathsf{Adv}_{\mathcal{A}_{\mathsf{dec}}^2,\mathcal{C}_{\mathsf{dec}}^{\mathsf{find}}}^{\mathsf{OHG}}(1^\lambda) = \mathsf{Adv}_{\mathcal{A}_1,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}},\ \mathrm{Time}[\mathcal{A}_1] = \mathrm{Time}[\mathcal{A}_{\mathsf{dec}}^2]. \tag{90}$$

Combining Eq. (86) to (90), we finally obtain the upper bound claimed for $|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}]|$ shown in Lemma 7. The upper bound of $|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]|$ shown in Lemma 7

[27] When the random oracle H, G and decapsulation oracle $O_{\mathsf{dec}}^{c_0^*}$ is queried by $\mathcal{A}$, $\mathcal{A}_{\mathsf{dec}}$ answers it by querying random oracle H, G and secret oracle O_{dec}, respectively. Note that the first check performed by O_{dec} is exactly the check that $c = \mathsf{Enc}(pk, m^*, y_1)$ by the definition of dec_2, hence $\mathcal{A}_{\mathsf{dec}}$ perfectly simulate $\mathcal{A}$'s view in game $\mathbf{G}_{\mathcal{A}}^{b=0}$.

can be obtained by the similar way with $|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}]|$, and we omit it. Note that compared to $|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=0}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=0}]|$, the upper bound of $|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]|$ shown in Lemma 7 does not have the term "$4(q_H + q_G + 1) \cdot \sqrt{\mathsf{Adv}_{\mathcal{A}_1,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}}$". Roughly speaking, the reason is that the operation in line 2 of game $\mathbf{G}_{\mathcal{A}}^{b=1}$ shown in Fig. 13 is already irreverent with the random oracle, hence, the game $\mathbf{G}_{\mathbf{4}}^{\mathbf{q}}$ to game $\mathbf{G}_{\mathbf{5}}^{\mathbf{q}}$ in the proof of Theorem 4 that are used to reprogram the challenger's random oracle query into fresh random value is redundant. This means that the upper bounds given by Eq. (32) and (38) to (40) of the proof of Theorem 4 can be removed from the final upper bound, and thus we obtain the bound we claim in Lemma 7 for $|\Pr[1 \leftarrow \mathbf{G}_{\mathcal{A}}^{b=1}] - \Pr[1 \leftarrow \mathbf{G}_{\mathcal{B}}^{b=1}]|$. □

H.4 SPR-qCCA security of $\mathsf{KEM}^{\perp}$, $\mathsf{KEM}_m^{\not\perp}$ and $\mathsf{KEM}^{\not\perp}$ in the QROM

Theorem 12. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, weakly γ-spread and* SDS-IND*-secure w.r.t. QPT simulator $\mathcal{S}$. Let $\mathcal{A}$ be a* SPR-qCCA *adversary against* $\mathsf{KEM}^{\perp}$ *in the QROM, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist an* OW-CPA *adversary $\mathcal{A}_1$ against the* PKE *and a* SDS-IND *adversary $\mathcal{A}_2$ against the* PKE *such that*

$$\mathsf{Adv}_{\mathcal{A},\mathcal{S},\mathsf{KEM}^{\perp}}^{\mathsf{SPR\text{-}qCCA}} \leq 24q_D \cdot \sqrt{\gamma} + 8(q_H + 1) \cdot \sqrt{\delta} + 64q_H \cdot \delta + 2(q_H + q_G + 1) \cdot \sqrt{\mathsf{Adv}_{\mathcal{A}_1,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}} + \mathsf{Adv}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}^{\mathsf{SDS\text{-}IND}}.$$

The running time of adversary $\mathcal{A}_1$ and $\mathcal{A}_2$ can be bounded as

$$\mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}_2] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Proof. The proof of this theorem is similar to Theorem 7 and we omit it. □

Theorem 13. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, weakly γ-spread and* SDS-IND*-secure w.r.t. QPT simulator $\mathcal{S}$. Let $\mathcal{A}$ be a* SPR-qCCA *adversary against* $\mathsf{KEM}_m^{\not\perp}$ *in the QROM, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist an adversary $\mathcal{A}'$ against the security of* PRF *with at most q_D queries, an* OW-CPA *adversary $\mathcal{A}_1$ against the* PKE *and a* SDS-IND *adversary $\mathcal{A}_2$ against the* PKE *such that*

$$\mathsf{Adv}_{\mathcal{A},\mathcal{S},\mathsf{KEM}_m^{\not\perp}}^{\mathsf{SPR\text{-}qCCA}} \leq \mathsf{Adv}_{\mathcal{A}'}^{\mathsf{PRF}} + 24q_D \cdot \sqrt{\gamma} + 8(q_H + 1) \cdot \sqrt{\delta} + 64q_H \cdot \delta + 2(q_H + q_G + 1) \cdot \sqrt{\mathsf{Adv}_{\mathcal{A}_1,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}} + \mathsf{Adv}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}^{\mathsf{SDS\text{-}IND}}.$$

The running time of adversary $\mathcal{A}'$, $\mathcal{A}_1$ and $\mathcal{A}_2$ can be bounded as

$$\mathrm{Time}[\mathcal{A}'] \approx \mathrm{Time}[\mathcal{A}],\ \mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}_2] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Proof. As shown in Fig. 5, compared with $\mathsf{KEM}_m^{\perp}$, the $\mathsf{KEM}_m^{\not\perp}$'s decapsulation algorithm returns $f(s, c)$ instead when c is an invalid encapsulation, where f is a pseudorandom function and $s \in \mathcal{K}^{prf}$ is randomly selected and part of the secret key.

Define a new game $\mathbf{G}$, which is identical with the SPR-qCCA game of $\mathsf{KEM}_m^{\not\perp}$ except that $R(c)$ is returned instead of $f(k, c)$ for an invalid encapsulation c, where R is an uniformly random function. Then via a straightforward reduction, there exists an adversary $\mathcal{A}'$ against the security of PRF with at most q_D queries such that

$$\left|\mathsf{Adv}_{\mathcal{A},\mathcal{S},\mathsf{KEM}_m^{\not\perp}}^{\mathsf{SPR\text{-}qCCA}} - \Pr[1 \leftarrow \mathbf{G}]\right| \leq \mathsf{Adv}_{\mathcal{A}'}^{\mathsf{PRF}},\ \mathrm{Time}[\mathcal{A}'] \approx \mathrm{Time}[\mathcal{A}].$$

Then similar with the proof of Theorem 7, we have

$$\Pr[1 \leftarrow \mathbf{G}] \leq 24q_D \cdot \sqrt{\gamma} + 8(q_H + 1) \cdot \sqrt{\delta} + 64q_H \cdot \delta + 2(q_H + q_G + 1) \cdot \sqrt{\mathsf{Adv}_{\mathcal{A}_1,\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}} + \mathsf{Adv}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}^{\mathsf{SDS\text{-}IND}}.$$

Combing above two equations we obtain our result. □

Theorem 14. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, weakly γ-spread and* SDS-IND*-secure w.r.t. QPT simulator $\mathcal{S}$. Let $\mathcal{A}$ be a* SPR-qCCA *adversary against* $\mathsf{KEM}^{\not\perp}$ *in the QROM, making at most q_H,*

q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist an $\mathsf{OW\text{-}CPA}$ *adversary $\mathcal{A}_1$ against the* PKE *and a* $\mathsf{SDS\text{-}IND}$ *adversary $\mathcal{A}_2$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{SPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S},\mathsf{KEM}^{\not\perp}} \leq 2q_H \cdot \frac{1}{\sqrt{2^u}} + 24q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 2(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}} + \mathsf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}}.$$

The running time of adversary $\mathcal{A}_1$ and $\mathcal{A}_2$ can be bounded as

$$\mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}_2] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Proof. This proof is similar with the proof of Theorem 13 except that we need to replace the $G(s,c)$ used by $\mathsf{Decaps}^{\not\perp}$ into $R(c)$, where R is an uniformly random function. By using the Lemma 2 of [JZC+18], the addition security loss is $2q_H \cdot \frac{1}{\sqrt{2^u}}$. □

H.5 $\mathsf{WPR\text{-}qCCA}$ security of $\mathsf{PKE}^{\perp}$, $\mathsf{PKE}^{\not\perp}_m$ and $\mathsf{PKE}^{\not\perp}$ in the QROM

Indeed, the $\mathsf{WPR\text{-}qCCA}$ security reductions of $\mathsf{PKE}^{\perp}$, $\mathsf{PKE}^{\not\perp}_m$ and $\mathsf{PKE}^{\not\perp}$ in the QROM are similar to that of $\mathsf{PKE}^{\perp}_m$. However, similar to Theorem 13 and Theorem 14 in Appendix H.4, the reductions of $\mathsf{PKE}^{\not\perp}_m$ and $\mathsf{PKE}^{\not\perp}$ need to first transform the pseudorandom functions used in the decryption algorithm into uniform random functions. Here, we directly give the theorems state that $\mathsf{PKE}^{\perp}$, $\mathsf{PKE}^{\not\perp}_m$ and $\mathsf{PKE}^{\not\perp}$ are $\mathsf{WPR\text{-}qCCA}$ security in the QROM and omit the proofs.

Theorem 15. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, weakly γ-spread and* $\mathsf{SDS\text{-}IND}$*-secure w.r.t. QPT simulator $\mathcal{S}$. Let $\mathcal{A}$ be a* $\mathsf{WPR\text{-}qCCA}$ *adversary against* $\mathsf{PKE}^{\perp}$ *in the QROM, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist a QPT simulator $\mathcal{S}'$ of* $\mathsf{PKE}^{\perp}$*, an* $\mathsf{OW\text{-}CPA}$ *adversary $\mathcal{A}_1$ against the* PKE *and a* $\mathsf{SDS\text{-}IND}$ *adversary $\mathcal{A}_2$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S}',\mathsf{PKE}^{\perp}} \leq 24q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 2(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}} + \mathsf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}}.$$

The running time of adversary $\mathcal{A}_1$ and $\mathcal{A}_2$ can be bounded as

$$\mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}_2] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Theorem 16. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, weakly γ-spread and* $\mathsf{SDS\text{-}IND}$*-secure w.r.t. QPT simulator $\mathcal{S}$. Let $\mathcal{A}$ be a* $\mathsf{WPR\text{-}qCCA}$ *adversary against* $\mathsf{PKE}^{\not\perp}_m$ *in the QROM, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist a QPT simulator $\mathcal{S}'$ of* $\mathsf{PKE}^{\not\perp}_m$*, an adversary $\mathcal{A}'$ against the security of* PRF *with at most q_D queries, an* $\mathsf{OW\text{-}CPA}$ *adversary $\mathcal{A}_1$ against the* PKE *and a* $\mathsf{SDS\text{-}IND}$ *adversary $\mathcal{A}_2$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S}',\mathsf{PKE}^{\not\perp}_m} \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}'} + 24q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 2(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}} + \mathsf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}}.$$

The running time of adversary $\mathcal{A}'$, $\mathcal{A}_1$ and $\mathcal{A}_2$ can be bounded as

$$\mathrm{Time}[\mathcal{A}'] \approx \mathrm{Time}[\mathcal{A}],\ \mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}_2] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$

Theorem 17. *Suppose* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, weakly γ-spread and* $\mathsf{SDS\text{-}IND}$*-secure w.r.t. QPT simulator $\mathcal{S}$. Let $\mathcal{A}$ be a* $\mathsf{WPR\text{-}qCCA}$ *adversary against* $\mathsf{PKE}^{\not\perp}$ *in the QROM, making at most q_H, q_G and q_D queries to random oracle H, G and decapsulation oracle, respectively. Then there exist a QPT simulator $\mathcal{S}'$ of* $\mathsf{PKE}^{\not\perp}$*, an* $\mathsf{OW\text{-}CPA}$ *adversary $\mathcal{A}_1$ against the* PKE *and a* $\mathsf{SDS\text{-}IND}$ *adversary $\mathcal{A}_2$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{WPR\text{-}qCCA}}_{\mathcal{A},\mathcal{S}',\mathsf{PKE}^{\not\perp}} \leq 2q_H \cdot \frac{1}{\sqrt{2^u}} + 24q_D \cdot \sqrt{\gamma} + 8(q_H+1)\cdot\sqrt{\delta} + 64q_H\cdot\delta + 2(q_H+q_G+1)\cdot\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{A}_1,\mathsf{PKE}} + \mathsf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathcal{A}_2,\mathcal{S},\mathsf{PKE}}}.$$

The running time of adversary $\mathcal{A}_1$ and $\mathcal{A}_2$ can be bounded as

$$\mathrm{Time}[\mathcal{A}_1] \approx \mathrm{Time}[\mathcal{A}_2] \leq \mathrm{Time}[\mathcal{A}] + O(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q_H^2).$$