New Proof for Plain OAEP: Post-Quantum Security without Parameter Restrictions or Collision-Resistance

Heming Liao[1,2], Jiangxia Ge[3], Shujiao Cao[1,2(✉)], and Rui Xue[1,2]

[1] State Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3] China Telecom Quantum Group Ltd., Hefei, China
{liaoheming,caoshujiao,xuerui}@iie.ac.cn,gejiangxia@chinatelecom.cn

Abstract. During NIST's post-quantum cryptography standardization process, two generic transforms, Fujisaki-Okamoto (FO) and OAEP, are widely used to achieve the IND-CCA security. For instance, the final winner Kyber has utilized FO, and a variant of the 3rd-round finalist NTRU has utilized OAEP. The FO and OAEP are both constructed in the random oracle model (ROM), so to evaluate their post-quantum security, a security proof in the quantum random oracle model (QROM) is required. So far, the QROM security proof of FO has been given and improved by a sequence of works, however, the QROM security proof of OAEP has not been fully explored: current proofs either introduced an extra plaintext-confirming hash to the ciphertext (TCC 2016), or required parameter restrictions and a quantum collision-resistant term (PKC 2022).
In this paper, by reorganizing the proof route, we give a new QROM security proof of plain OAEP. The key techniques used in our proof are the compressed oracle technique proposed by Zhandry (CRYPTO 2019) and the Fixed Permutation One-Way to Hiding recently proposed by Jaeger (Eprint 2024/797), and our proof has the following three advantages:

- Similar to Ebrahimi's proof (PKC 2022), our proof also achieves the stronger IND-qCCA security, where the decryption oracle can be accessed in superposition.
- The parameter restrictions "$n + k_1 \geq k_0$" and "$k_0 - n = \mathcal{O}(n)$" introduced in Ebrahimi's proof (PKC 2022) are removed.
- The reliance on the collision resistance of quantum random oracles required in Ebrahimi's proof (PKC 2022) is avoided, and hence our security bound does not introduce the quantum collision-resistant term "$\mathcal{O}(q^3/2^{n+k_1})$".

Keywords: quantum random oracle model · security proof · OAEP transform · IND-qCCA security.

1 Introduction

In [35], Shor designed an efficient quantum algorithm to solve the factoring and discrete logarithm problem. This sparked the cryptographic community to

study a new field named post-quantum cryptography (PQC), which aims at constructing classical systems secure against quantum adversaries. In December 2016 [29], National Institute of Standards and Technology (NIST) initiated a project to standardize public-key cryptographic algorithms which are secure against quantum adversaries, including public key encryption (PKE), digital signatures, and key encapsulation mechanisms (KEM). Among the PKE/KEM submissions, the indistinguishability against chosen-ciphertext attacks (IND-CCA) is a widely adopted security goal. To achieve this goal, the Fujisaki-Okamoto (FO) transform [14,15] and Optimal Asymmetric Encryption Padding (OAEP) transform [3], along with their variants, have been frequently used. In particular, the final winner Kyber [30] uses (a tweaked version of) $\mathsf{FO}^{\not\perp}_m$, which is a KEM variant of FO, and a variant of the 3rd-round finalist NTRU [31] uses Q-OAEP, which is a variant of OAEP by adding a plaintext-confirming hash.

Compared with FO, OAEP has a more concise structure, since its decryption does not require a re-encryption check. Meanwhile, different from FO, the security proof of OAEP does not require the underlying scheme to be γ-spread. However, a drawback of OAEP is that, unlike FO which only requires the underlying scheme to be one-way secure against chosen-plaintext attacks (OW-CPA), OAEP instead requires a stronger partial domain one-wayness (PD-OW) security. Nevertheless, we are not pessimistic about this situation. Firstly, one may construct PD-OW secure cryptographic primitives based on lattice or code theory. Moreover, OAEP indeed offers scheme designers another option to achieve IND-CCA security besides FO, particularly in scenarios where re-encryption check leads to side-channel vulnerabilities [38]. Overall, FO and OAEP have their own advantages and disadvantages, and their post-quantum security needs to be considered when assessing the post-quantum security of NIST's submissions.

Notice that FO and OAEP both are transforms in the random oracle model (ROM). Hence by [6], when analyzing their post-quantum security, we should prove their IND-CCA security in the quantum random oracle model (QROM). Up to now, a long sequence of works [37,24,4,27,20,34,16] have studied the IND-CCA security proofs of FO (or its variants) in the QROM. These works gradually optimized the security proof to get tighter security bounds, and [34,16] also considered a stronger security notion named indistinguishability against quantum chosen-ciphertext attacks (IND-qCCA), where the adversary can make decryption/decapsulation oracle queries in superposition.

On the other hand, however, research on the QROM security proof of OAEP is not yet in-depth. Ebrahimi and Unruh [37] provided the IND-CCA security proof of Q-OAEP in the QROM, where Q-OAEP is a variant of OAEP obtained by adding an extra plaintext-confirming hash to the ciphertext. Later, Ebrahimi [13] presented the first QROM security proof of plain OAEP, which also achieved the stronger IND-qCCA security. However, Ebrahimi's proof additionally requires the parameters n, k_0, k_1 used by OAEP to satisfy $n+k_1 \geq k_0$ and $k_0 - n = \mathcal{O}(n)$, and its security bound has a quantum collision-resistant term $\mathcal{O}(q^3/2^{n+k_1})$ since his proof relies on the collision resistance of quantum random oracles. This makes his proof looser than the existing proof of FO [24,20,34]. Based on this observa-

tion, in this paper we focus on the following natural question:

In the QROM, can we prove the IND-CCA security of plain OAEP, without parameter restrictions or the quantum collision-resistant term?

1.1 Our Contribution

We give an affirmative answer to the above question. In this paper, we present a new QROM security proof of plain OAEP, and similar to [13], we also achieve a stronger IND-qCCA security. The detailed security bound is shown in Table 1.

Table 1. Security bounds in the QROM. Here q is the total number of queries to random oracles, and q_{D} is the total number of queries to the decryption oracle. The "Requirement" shows the property needed to be satisfied by the underlying scheme. ϵ denotes the security bound of the underlying scheme. n, k_0, k_1 and γ denote the parameters for the corresponding transforms, and WLOG we assume $n = k_0 = k_1 = \gamma$.

Transform	Underlying security	Achieved security	Requirement	Security bound ($\approx$)
FO [34]	OW-CPA	IND-CCA	γ-spread	$(q_{\mathsf{D}} + q) \cdot \sqrt{\epsilon} + q \cdot \sqrt{2^{-\gamma}}$[a]
OAEP [13]	PD-OW	IND-qCCA	$n + k_1 \geq k_0$ $k_0 - n = \mathcal{O}(n)$	$q \cdot \sqrt{\epsilon} + q \cdot \sqrt{2^{-k_0}} + q_{\mathsf{D}} \cdot \sqrt{2^{-k_1}}$ $+q^3/2^{n+k_1}$
OAEP Thm.4	PD-OW	IND-qCCA	\	$q_{\mathsf{D}} \cdot q \cdot \sqrt{\epsilon} + q_{\mathsf{D}} \cdot q \cdot \sqrt{2^{-k_0}} + q_{\mathsf{D}} \cdot \sqrt{2^{-k_1}}$

[a] In [34], the security bound of FO also contains another parameter, ϵ^{sy}, which denotes the security bound of the underlying secret key scheme. Here we omit it WLOG.

As shown in Table 1, our bound avoids the quantum collision-resistant term $q^3/2^{n+k_1}$. So at least in the case where q_{D} is relatively small (e.g. a small constant[4]), our bound is comparable with that of FO in [34] and is tighter than that in [13]. However, when q_{D} is large, our bound is looser than that of FO in [34], and in this case it seems hard to compare our bound with that in [13].

Nevertheless, regardless of whether q_{D} is small or large, our proof does not impose additional parameter restrictions (i.e. $n + k_1 \geq k_0$ and $k_0 - n = \mathcal{O}(n)$) compared with [13]. Roughly speaking, setting $n = 256$, the parameter restrictions in [13] will make $k_0 = 3*256$, $k_1 = 2*256$. Consequently, to encrypt 256-bit plaintexts using OAEP, the input length of the underlying injective function f_{pk} should be $n + k_1 + k_0 = 6 * 256$, and hence the length of the ciphertext c will not be shorter than 6×256 since c is the output of f_{pk}. Fortunately, in our proof

[4] There do exist scenarios where the number of decryption/decapsulation queries q_{D} is small, see [11,25,40,18,21,23] for instance. These works focus on certain protocols in which IND-t-CCA secure PKEs/KEMs are sufficient, where IND-t-CCA is the same security notion as IND-CCA except that the adversary can only query the decryption/decapsulation oracles at most t times. Here t is a small constant independent of the security parameter.

$n = k_1 = k_0 = 256$ is enough to encrypt 256-bit plaintexts, which will result in a scheme with higher computational efficiency and smaller ciphertext size.

Remark 1. We would like to point out that the corresponding proof in [13], which leads to the highlighted term $q \cdot \sqrt{\epsilon} + q \cdot \sqrt{2^{-k_0}}$, may contain a potential gap. A detailed discussion is provided in Supplementary Material A.

1.2 Technical Overview

We begin by briefly reviewing the structure of OAEP. Let $\mathsf{TDI} = (\mathsf{Gen}, f, f^{-1})$ be a trapdoor injective function (Definition 6). Let $\mathsf{G} : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ and $\mathsf{H} : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ be two random oracles. The structure of the encryption and decryption process of OAEP is shown in Fig. 1.

The encryption process of OAEP first uses a 2-round Feistel network to transform the plaintext m into two components (s,t), and then computes the ciphertext $c = f_{\mathsf{pk}}(s,t)$. The structure of the decryption process of OAEP is symmetric to that of the encryption: after receiving c, first derive s and t by computing $f_{\mathsf{sk}}^{-1}(c)$, and then invert the 2-round Feistel network to recover M. Finally, check the redundancy 0^{k_1} added to m during encryption. If $[M]_{k_1} = 0^{k_1}$, return $[M]^n$; otherwise, return $\perp$.

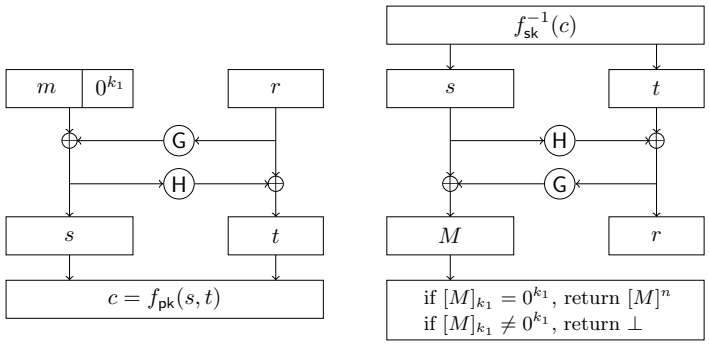


Fig. 1. The encryption and decryption structure of OAEP. Here $(\mathsf{pk}, \mathsf{sk})$ is generated by Gen, m denotes the plaintext, and r is the randomness of encrypting m.

We now briefly outline our new proof route. In the QROM, we complete the IND-qCCA security proof of OAEP through 17 games[5], among which $\mathbf{G_5}$, $\mathbf{G_{14}}$, and $\mathbf{G_{17}}$, shown in Fig. 2, serve as three core games. Based on these core games, our proof can be divided into the following three steps:

1. **Games $\mathbf{G_0}$-$\mathbf{G_5}$:** Game $\mathbf{G_0}$ is the IND-qCCA game of OAEP, where the simulation of decryption oracle qDec requires the secret key sk. In this step, by

[5] The number of games may seem a bit large, but this is reasonable, as some game hops are conceptual steps designed to improve the clarity of the proof.

simulating G and H as compressed oracles, we gradually modify qDec and finally obtain $\mathsf{qDec}_{\mathsf{sk}}$ in game $\mathbf{G_5}$, where $\mathsf{qDec}_{\mathsf{sk}}$ no longer depends on sk.

2. **Games $\mathbf{G_5}$-$\mathbf{G_{14}}$:** In this step, we reprogram the oracles G and H queried before $\mathcal{A}$ outputs (m_0^*, m_1^*). After this reprogramming, we get game $\mathbf{G_{14}}$, where $\mathsf{G}(r^*)$ and $\mathsf{H}(s^*)$ used to generate c^* and b' are replaced by fresh uniformly random values g^* and h^*. Moreover, since G and H have been simulated as compressed oracles, the unitary operations $\mathrm{U}_{g^*}^{r^*}$ (Eq. (15)) and $\mathrm{U}_{h^*}^{s^*}$ (Eq. (19)) must be applied in game $\mathbf{G_{14}}$ to maintain consistency.
3. **Games $\mathbf{G_{14}}$-$\mathbf{G_{17}}$**: In this step, based on game $\mathbf{G_{14}}$, we reprogram the oracles $\mathsf{G}[r^* \mapsto g^*]$ and $\mathsf{H}[s^* \mapsto h^*]$ queried during the generation of b'. After this reprogramming, we obtain game $\mathbf{G_{17}}$, where $\mathsf{G}[r^* \mapsto g^*]$ and $\mathsf{H}[s^* \mapsto h^*]$ are replaced by the original G and H, respectively, and the implementation of $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ is canceled. Besides, s^* (resp. t^*) in game $\mathbf{G_{17}}$ is no longer computed based on g^* (resp. h^*), but is instead uniformly sampled.

At this point, we have $\Pr[1 \leftarrow \mathbf{G_{17}}] = 1/2$, and by combining the bounds derived in each step, we obtain the $\mathsf{IND\text{-}qCCA}$ security bound of OAEP and complete the proof. In the following, we will provide a high-level explanation of our techniques for each step.

GAME $\mathbf{G_5}$	**GAME $\mathbf{G_{14}}$**	**GAME $\mathbf{G_{17}}$**
1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$	1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$	1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$
uniformly sample r^*	uniformly sample r^*, g^*, h^*	uniformly sample r^*, s^*, t^*
2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}_{\mathsf{sk}}}$	2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}_{\mathsf{sk}}}$	2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}_{\mathsf{sk}}}$
3. $b \leftarrow_{\$} \{0,1\}$	3. $b \leftarrow_{\$} \{0,1\}$	3. $b \leftarrow_{\$} \{0,1\}$
$s^* = (m_b^* \| 0^{k_1}) \oplus \mathsf{G}(r^*)$	$s^* = (m_b^* \| 0^{k_1}) \oplus g^*$	$c^* = f_{\mathsf{pk}}(s^*, t^*)$
$t^* = r^* \oplus \mathsf{H}(s^*)$	$t^* = r^* \oplus h^*$	4. $b' \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}_{\mathsf{sk}}}(c^*)$
$c^* = f_{\mathsf{pk}}(s^*, t^*)$	$c^* = f_{\mathsf{pk}}(s^*, t^*)$, perform $\mathrm{U}_{h^*}^{s^*} \circ \mathrm{U}_{g^*}^{r^*}$	5. **return** $[\![b' = b]\!]$
4. $b' \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}_{\mathsf{sk}}}(c^*)$	4. $b' \leftarrow \mathcal{A}^{\mathsf{G}[r^* \mapsto g^*],\mathsf{H}[s^* \mapsto h^*],\mathsf{qDec}_{\mathsf{sk}}}(c^*)$	
5. **return** $[\![b' = b]\!]$	5. **return** $[\![b' = b]\!]$	

Fig. 2. Three core games $\mathbf{G_5}$, $\mathbf{G_{14}}$ and $\mathbf{G_{17}}$. Here for simplicity and clarity we only present informal descriptions of the corresponding games defined in Section 4. Moreover, the $\mathsf{G}[r^* \mapsto g^*]$ (resp. $\mathsf{H}[s^* \mapsto h^*]$) used in game $\mathbf{G_{14}}$ is the same as G (resp. H) except that it directly replies g^* (resp. h^*) for the query r^* (resp. s^*).

Techniques in the game hop from $\mathbf{G_0}$ to $\mathbf{G_5}$: In this game hop, we introduce two variants of qDec: qDec_1 and qDec_2, as the bridge transforming qDec to $\mathsf{qDec}_{\mathsf{sk}}$[6]. The construction of qDec_1 and qDec_2 is shown in Fig. 3.

[6] In fact, the notations "qDec_1", "qDec_2", and "$\mathsf{qDec}_{\mathsf{sk}}$" were not used in our proof. They are introduced here only for the convenience of explaining the techniques.

– Compared with qDec, qDec_1 simulates G as a compressed oracle and adds a judgment for the query state $|c, y, D_{\mathsf{G}}\rangle$: if $D_{\mathsf{G}}(r) = \perp$, qDec_1 returns $\perp$, otherwise the next "$M = s \oplus \mathsf{G}(r)$" is replaced by "$M = s \oplus D_{\mathsf{G}}(r)$". Intuitively speaking, if $D_{\mathsf{G}}(r) = \perp$, the compressed oracle simulation of G on r will purify $\mathsf{G}(r)$ into a uniform superposition $\sqrt{2^{-n-k_1}} \sum_{l \in \{0,1\}^{n+k_1}} |l\rangle$. Obviously, for a fixed s, only 2^n out of 2^{n+k_1} l in this superposition satisfy $[s \oplus l]_{k_1} = 0^{k_1}$. Hence, even if qDec_1 returns $\perp$ when $D_{\mathsf{G}}(r) = \perp$, the probability that such change is detected cannot exceed $2^n/2^{n+k_1} = 2^{-k_1}$. Moreover, if $D_{\mathsf{G}}(r) \neq \perp$, we can indeed replace $\mathsf{G}(r)$ by $D_{\mathsf{G}}(r)$. The reason is that, roughly speaking, for a state $|c, y, D_{\mathsf{G}}\rangle$ with $D_{\mathsf{G}}(r) \neq \perp$, the probability that $\mathsf{G}(r)$ computed by the compressed oracle does not equal to $D_{\mathsf{G}}(r)$ is negligible. Thus with a negligible error we can replace qDec with qDec_1. Indeed, we compute the difference between qDec and qDec_1 in the proof of Lemma 1, and find that it can be bounded by $8 \cdot \sqrt{2^{-k_1}}$. The square root arises because both qDec and qDec_1 can be accessed in superposition.

$\mathsf{qDec}_1(\|c, y, D_{\mathsf{G}}\rangle)$	$\mathsf{qDec}_2(\|c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle)$
1. **if** $f_{\mathsf{sk}}^{-1}(c) = \perp$, **return** $\|c, y \oplus \perp, D_{\mathsf{G}}\rangle$	1. **if** $f_{\mathsf{sk}}^{-1}(c) = \perp$, **return** $\|c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$
2. **else** phase $f_{\mathsf{sk}}^{-1}(c)$ into $s\|\|t$	2. **else** phase $f_{\mathsf{sk}}^{-1}(c)$ into $s\|\|t$
$r = t \oplus \mathsf{H}(s)$	**if** $D_{\mathsf{H}}(s) = \perp$, **return** $\|c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{G}}\rangle$
if $D_{\mathsf{G}}(r) = \perp$, **return** $\|c, y \oplus \perp, D_{\mathsf{G}}\rangle$	**else** $r = t \oplus D_{\mathsf{H}}(s)$
else $M = s \oplus D_{\mathsf{G}}(r)$	**if** $D_{\mathsf{G}}(r) = \perp$, **return** $\|c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$
3. **if** $[M]_{k_1} = 0^{k_1}$, **return** $\|c, y \oplus [M]^n, D_{\mathsf{G}}\rangle$	**else** $M = s \oplus D_{\mathsf{G}}(r)$
else return $\|c, y \oplus \perp, D_{\mathsf{G}}\rangle$	3. **if** $[M]_{k_1} = 0^{k_1}$, **return** $\|c, y \oplus [M]^n, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$
	else return $\|c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$

Fig. 3. Decryption oracles qDec_1 and qDec_2. Since they are quantum accessible, we write their query state as $|c, y\rangle$. When G or H is simulated as a compressed oracle, we attach the corresponding database state $|D_{\mathsf{G}}\rangle$ or $|D_{\mathsf{H}}\rangle$ to $|c, y\rangle$. We also stress that we omit the judgment whether c equals to the challenge ciphertext c^* for simplicity.

– Based on qDec_1, we make a similar change in qDec_2, which simulates H as a compressed oracle and adds a judgment for the query state $|c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$: if $D_{\mathsf{H}}(s) = \perp$, qDec_2 returns $\perp$, otherwise the next "$r = t \oplus \mathsf{H}(s)$" is replaced by "$r = t \oplus D_{\mathsf{H}}(s)$". The difference between qDec_1 and qDec_2 is analogous to that between qDec and qDec_1. Roughly speaking, if $D_{\mathsf{H}}(s) = \perp$, $\mathsf{H}(s)$ computed by the compressed oracle is uniformly random, making $D_{\mathsf{G}}(r) \neq \perp$ ($r = t \oplus \mathsf{H}(s)$) hard to achieve since there are at most q_{G} different x satisfying $D_{\mathsf{G}}(x) \neq \perp$. So under a negligible error we can directly return $\perp$ when $D_{\mathsf{H}}(s) = \perp$ in qDec_2. For the case that $D_{\mathsf{H}}(s) \neq \perp$, similar to the change from $\mathsf{G}(r)$ to $D_{\mathsf{G}}(r)$ in qDec_1, we can change $\mathsf{H}(s)$ to $D_{\mathsf{H}}(s)$ in qDec_2 with a negligible error. In fact,

we compute the difference between qDec_1 and qDec_2 in the proof of Lemma 2, and find that it is bounded by $8\sqrt{q_{\mathsf{G}}} \cdot \sqrt{2^{-k_0}}$.

– If qDec_2 does not return $\perp$ for the query state $|c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$, we know that there must exist (r, s) s.t. $D_{\mathsf{G}}(r) \neq \perp$, $D_{\mathsf{H}}(s) \neq \perp$ and $f_{\mathsf{sk}}^{-1}(c) = (s, r \oplus D_{\mathsf{H}}(s))$. That is, the inverse (s, t) $(= f_{\mathsf{sk}}^{-1}(c))$ of c must already be stored in the databases D_{G} and D_{H} as it is satisfied that $D_{\mathsf{H}}(s) \neq \perp$ and $D_{\mathsf{G}}(t \oplus D_{\mathsf{H}}(s)) \neq \perp$. Based on this observation, by using the injectivity of f_{pk} and ignoring the decryption error of the underlying TDI[7], in Supplementary Material D.4 we show that we can equivalently transform qDec_2 into $\mathsf{qDec}_{\not\mathsf{sk}}$, which does not use the secret key sk, to compute $f_{\mathsf{sk}}^{-1}(c)$ and instead directly searches $(r, D_{\mathsf{G}}(r) \neq \perp)$, $(s, D_{\mathsf{H}}(s) \neq \perp)$ from D_{G}, D_{H} such that $f_{\mathsf{pk}}(s, r \oplus D_{\mathsf{H}}(s)) = c$ and $[s \oplus D_{\mathsf{G}}(r)]_{k_1} = 0^{k_1}$, and then returns $[s \oplus D_{\mathsf{G}}(r)]^n$ (resp. $\perp$) if this search succeeds (resp. fails).

At this point, the decryption oracle $\mathsf{qDec}_{\not\mathsf{sk}}$ used in game $\mathbf{G_5}$ no longer uses sk.

Remark 2 (How we avoid the parameter restrictions and the quantum collision-resistant term). In [13], the author noticed that the search over the databases (like qDec_1, qDec_2 and $\mathsf{qDec}_{\not\mathsf{sk}}$ we described above) can disturb the simulation of the compressed oracle, which may be detected by the adversary. Hence [13] used the online extraction technique in [12] to bound the potential disturbances, which introduces a special parameter $\Gamma_{R_{\mathsf{H}}^c}$. In order to bound $\Gamma_{R_{\mathsf{H}}^c}$, [13] transformed $\mathsf{G} : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ into an injective function, which inevitably introduced the restriction $n + k_1 \geq k_0$ and the quantum collision-resistant term. Further, in order to make $\Gamma_{R_{\mathsf{H}}^c}$ negligible, the restriction $k_0 - n = \mathcal{O}(n)$ is also introduced. In our game hop $\mathbf{G_0}$ to $\mathbf{G_5}$ explained above, we simulate G and H both as compressed oracles, and then similar to the proof of [8,16], we compute the differences between qDec and qDec_1, qDec_1 and qDec_2, qDec_2 and $\mathsf{qDec}_{\not\mathsf{sk}}$ directly by the basic rules of compressed oracle. Essentially, the computation of these differences has already covered the estimation of the disturbances mentioned above, therefore the online extraction technique is no longer needed, thus avoiding the parameter restrictions and the quantum collision-resistant term needed in [13].

Techniques in the game hop from $\mathbf{G_5}$ to $\mathbf{G_{14}}$: In this game hop, we use three middle games $\mathbf{G_5^a}$ to $\mathbf{G_5^c}$ as shown in Fig. 4 to reprogram the oracles G and H queried before $\mathcal{A}$ outputs (m_0^*, m_1^*).

– Compared with game $\mathbf{G_5}$ shown in Fig. 2, game $\mathbf{G_5^a}$ introduces the semi-classical oracle $\mathcal{O}^{SC}_{\{r^*\}}$ to perform a binary projective measurement on $\mathcal{A}$'s quantum query to G, and adds a flag that if Find_{r^*} occurs, game finally returns $\perp$. Here Find_{r^*} is the event that $\mathcal{O}^{SC}_{\{r^*\}}$ ever has measurement result 1. Now by directly using the Semi-Classical One-way to Hiding (SC-O2H) theorem [1], we can bound the difference between game $\mathbf{G_5}$ and $\mathbf{G_5^a}$.

[7] That is, in our paper we assume that $c = f_{\mathsf{pk}}(s, t)$ must imply $f_{\mathsf{sk}}^{-1}(c) = (s, t)$.

– The flag added in game $\mathbf{G_5^a}$ actually indicates that if game $\mathbf{G_5^a}$ returns 1, then the event Find_{r^*} must do not occur. Hence, since we only focus on the probability $\Pr[1 \leftarrow \mathbf{G_5^a}]$, we can default to the assumption that Find_{r^*} does not occur. Under this assumption, and by the definition of $\mathcal{O}^{SC}_{\{r^*\}}$ (Definition 1), we can conclude that $\mathcal{A}$ only queries G with a query state in the form $\sum_{x \neq r^*, y} \alpha_{x,y}|x, y\rangle$ before $\mathcal{A}$ outputs (m_0^*, m_1^*). This means that the $\mathsf{G}(r^*)$ used to compute s^* in the next step is actually the first r^* query to G. Recall that G is simulated as a compressed oracle, so by its basic rules, we can equivalently change the compressed oracle simulation of $\mathsf{G}(r^*)$ in game $\mathbf{G_5^a}$ to the particular simulation of $\mathsf{G}(r^*)$ in game $\mathbf{G_5^b}$. This simulation first chooses g^* uniformly and views it as $\mathsf{G}(r^*)$, then inserts (r^*, g^*) into the database state by sequentially applying the unitary operation $\mathrm{U}^{r^*}_{g^*}$ (Eq. (15)) and the local decompression procedure $\mathsf{StdDecomp}_{r^*}$ (Section 2.4).

  Moreover, in order to maintain consistency, during the generation of b', game $\mathbf{G_5^b}$ replaces oracle G as $\mathsf{G}[r^* \mapsto g^*]$, which is simulated the same as G except that it directly replies g^* to the query r^*. In fact, by basic rules of compressed oracle, such replacement cannot be detected by the adversary if $\mathsf{StdDecomp}_{r^*}$ commutes with $\mathsf{qDec}_{\mathsf{sk}}$. To address this, we prove our Lemma 4, which shows that $\mathsf{StdDecomp}_{r^*}$ almost commutes with $\mathsf{qDec}_{\mathsf{sk}}$, and hence we can transform game $\mathbf{G_5^a}$ into game $\mathbf{G_5^b}$ under a negligible error.

GAME $\mathbf{G_5^a}$	**GAME $\mathbf{G_5^b}$**	**GAME $\mathbf{G_5^c}$**
1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$	1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$	1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$
r^* uniformly sample	r^*, g^* uniformly sample	r^*, s^*, h^* uniformly sample
2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G} \circ \mathcal{O}^{SC}_{\{r^*\}}, \mathsf{H}, \mathsf{qDec}_{\mathsf{sk}}}$	2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G} \circ \mathcal{O}^{SC}_{\{r^*\}}, \mathsf{H}, \mathsf{qDec}_{\mathsf{sk}}}$	2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G} \circ \mathcal{O}^{SC}_{\{r^*\}}, \mathsf{H} \circ \mathcal{O}^{SC}_{\{s^*\}}, \mathsf{qDec}_{\mathsf{sk}}}$
if Find_{r^*} occurs, **return** $\perp$	**if** Find_{r^*} occurs, **return** $\perp$	**if** $\mathsf{Find}_{r^*} \vee \mathsf{Find}_{s^*}$ occurs, **return** $\perp$
3. $b \leftarrow_{\$} \{0,1\}$	3. $b \leftarrow_{\$} \{0,1\}$	3. $b \leftarrow_{\$} \{0,1\}$
$s^* = (m_b^*\|\|0^{k_1}) \oplus \mathsf{G}(r^*)$	$s^* = (m_b^*\|\|0^{k_1}) \oplus g^*$	$g^* = (m_b^*\|\|0^{k_1}) \oplus s^*$
$t^* = r^* \oplus \mathsf{H}(s^*)$	$t^* = r^* \oplus \mathsf{H}(s^*)$	$t^* = r^* \oplus h^*$
$c^* = f_{\mathsf{pk}}(s^*, t^*)$	$c^* = f_{\mathsf{pk}}(s^*, t^*)$, perform $\mathrm{U}^{r^*}_{g^*}$	$c^* = f_{\mathsf{pk}}(s^*, t^*)$, perform $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}$
4. $b' \leftarrow \mathcal{A}^{\mathsf{G}, \mathsf{H}, \mathsf{qDec}_{\mathsf{sk}}}(c^*)$	perform $\mathsf{StdDecomp}_{r^*}$	perform $\mathsf{StdDecomp}_{s^*} \circ \mathsf{StdDecomp}_{r^*}$
5. **return** $[\![b' = b]\!]$	4. $b' \leftarrow \mathcal{A}^{\mathsf{G}[r^* \mapsto g^*], \mathsf{H}, \mathsf{qDec}_{\mathsf{sk}}}(c^*)$	4. $b' \leftarrow \mathcal{A}^{\mathsf{G}[r^* \mapsto g^*], \mathsf{H}[s^* \mapsto h^*], \mathsf{qDec}_{\mathsf{sk}}}(c^*)$
	5. **return** $[\![b' = b]\!]$	5. **return** $[\![b' = b]\!]$

Fig. 4. Games $\mathbf{G_5^a}$ to $\mathbf{G_5^c}$. We stress that, these three games were not used in our proof, and they are introduced here for the convenience of explaining our techniques.

– The s^* in game $\mathbf{G_5^b}$ is computed as $(m_b^*||0^{k_1}) \oplus g^*$, where g^* is uniformly sampled. In game $\mathbf{G_5^c}$, based on the basic property of XOR, we make a conceptual change, i.e., we first uniformly sample s^* at the beginning of the game and then compute $g^* = (m_b^*||0^{k_1}) \oplus s^*$. Therefore the generation of s^* in game $\mathbf{G_5^c}$

is independent of the adversary. Based on this fact, we make another three changes in game $\mathbf{G_5^c}$ as follows:

- We introduce the semi-classical oracle $\mathcal{O}^{SC}_{\{s^*\}}$ to perform a binary projective measurement on $\mathcal{A}$'s quantum query to H, and add a flag that if Find_{s^*} occurs, game directly returns $\perp$.
- The compressed oracle simulation of $\mathsf{H}(s^*)$ is changed to a particular simulation, which first chooses h^* uniformly and views it as $\mathsf{H}(s^*)$, then inserts (s^*, h^*) into the database state by sequentially applying the unitary operation $\mathrm{U}^{s^*}_{h^*}$ (Eq. (19)) and the local decompression procedure $\mathsf{StdDecomp}_{s^*}$.
- The oracle H queried during the generation of b' is replaced by $\mathsf{H}[s^* \mapsto h^*]$, which is the same as H except that it directly replies h^* to the query s^*.

Obviously the above changes are the same as those made in game $\mathbf{G_5^a}$ and $\mathbf{G_5^b}$, except that the perspective is shifted from oracle G to oracle H. Hence similar to the explanation of game hop $\mathbf{G_5}$ to $\mathbf{G_5^a}$ and game hop $\mathbf{G_5^a}$ to $\mathbf{G_5^b}$, the difference between game hop $\mathbf{G_5^b}$ to $\mathbf{G_5^c}$ is also negligible.

– Based on game $\mathbf{G_5^c}$, we make three changes to get game $\mathbf{G_{14}}$ shown in Fig. 2. First, we remove $\mathcal{O}^{SC}_{\{r^*\}}$, $\mathcal{O}^{SC}_{\{s^*\}}$, and the flags related to Find_{r^*} and Find_{s^*}. The difference caused by this change can be bounded by using the SC-O2H theorem in reverse. Second, instead of uniformly sampling s^* and then computing $g^* = (m_b^*||0^{k_1}) \oplus s^*$, game $\mathbf{G_{14}}$ instead first uniformly samples g^* and then computes $s^* = (m_b^*||0^{k_1}) \oplus g^*$. Obviously, this is just a conceptual change.
Third, we remove the performing of $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$. In fact, $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ must commute with $\mathsf{G}[r^* \mapsto g^*]$ and $\mathsf{H}[s^* \mapsto h^*]$, and the reason is that, roughly speaking, the query to $\mathsf{G}[r^* \mapsto g^*]$ and $\mathsf{H}[s^* \mapsto h^*]$ at point r^* and s^*, respectively, is no longer simulated by the compressed oracle. Furthermore, as proved in our Lemma 4, $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ almost commute with $\mathsf{qDec}_{\mathsf{sk}}$. According to these two facts, under a negligible error, we move the performing of $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ to the end of the game and then delete them.

At this point, we complete the game hop from $\mathbf{G_5}$ to $\mathbf{G_{14}}$, during which we successfully reprogram the oracles G and H queried before $\mathcal{A}$ outputs (m_0^*, m_1^*).
Techniques in the game hop $\mathbf{G_{14}}$ to $\mathbf{G_{17}}$: In this game hop, we use two middle games $\mathbf{G_{14}^a}$ and $\mathbf{G_{14}^b}$ as shown in Fig. 5 to reprogram the oracles $\mathsf{G}[r^* \mapsto g^*]$ and $\mathsf{H}[s^* \mapsto h^*]$ queried during the generation of b'.

– Compared with game $\mathbf{G_{14}}$ shown in Fig. 2 which first uniformly samples g^*/h^* and then computes $s^* = (m_b^*||0^{k_1}) \oplus g^*/t^* = r^* \oplus h^*$, game $\mathbf{G_{14}^a}$ first uniformly samples s^*/t^* and then computes $g^* = (m_b^*||0^{k_1}) \oplus s^*/h^* = r^* \oplus t^*$. Obviously this is just a conceptual change.

– Compared with game $\mathbf{G_{14}^a}$, game $\mathbf{G_{14}^b}$ removes the performing of $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}$. In fact, by the definition of $\mathrm{U}^{r^*}_{g^*}$ and $\mathrm{U}^{s^*}_{h^*}$ given in Eq. (15) and Eq. (19), they both quantumly compute a permutation on the corresponding database state. Now a key observation is that, roughly speaking, the (composite) permutation

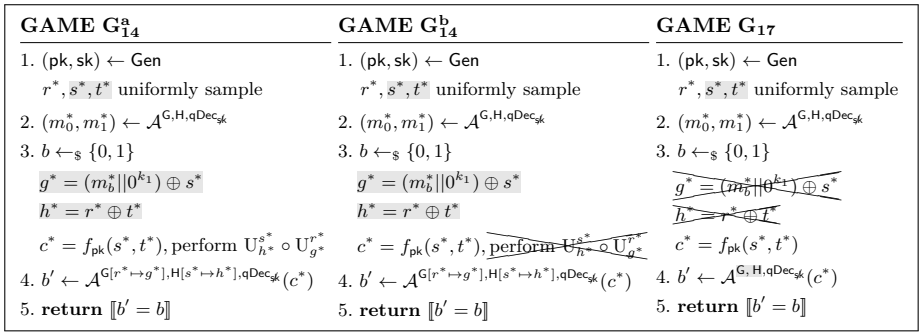


Fig. 5. Games $\mathbf{G^a_{14}}$ and $\mathbf{G^b_{14}}$. We stress that, these two games were not used in our proof, and they are introduced here for the convenience of explaining our techniques.

computed by $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}$ will disturb the database state and hence affect the search result of $\mathsf{qDec}_{\not\mathsf{sk}}$ performed during the generation of b'. This means that, if we remove the performing of $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}$, the simulation of $\mathsf{qDec}_{\not\mathsf{sk}}$ during the generation of b' will return affected search results. Based on this observation we design a quantum oracle algorithm $\mathcal{A}_1$ and abstract two fixed and different permutations P_0 and P_1 such that

$$\Pr[1 \leftarrow \mathbf{G^a_{14}}] = \Pr[1 \leftarrow \mathcal{A}_1^{\mathsf{P}_0}],\ \Pr[1 \leftarrow \mathbf{G^b_{14}}] = \Pr[1 \leftarrow \mathcal{A}_1^{\mathsf{P}_1}].$$

Here oracle P_0 (resp. P_1) maps the query state $|c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$ of $\mathsf{qDec}_{\not\mathsf{sk}}$ into $|\mathsf{P}_0(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})\rangle$ (resp. $|\mathsf{P}_1(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})\rangle$), and permutations P_0 and P_1 satisfy that $\mathsf{P}_0(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \neq \mathsf{P}_1(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})$ if and only if [8] $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R_1 \cup R_2$, where

$$R_1 := \{(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) : \exists (s, D_{\mathsf{H}}(s) \neq \perp) \text{ s.t. } c = f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s))\},$$
$$R_2 := \{(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) : \exists (r, D_{\mathsf{G}}(r) \neq \perp) \text{ s.t. } c = f_{\mathsf{pk}}(s^*, r \oplus r^* \oplus t^*)\}.$$

Thanks to the elegant structure of the Fixed Permutation One-Way to Hiding theorem [22], we are now able to bound the difference between game $\mathbf{G^a_{14}}$ and $\mathbf{G^b_{14}}$ by the success probability of another adversary aiming to guess one tuple $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R_1 \cup R_2$. In fact, the success probability of such guessing adversary must be negligible. The reason is that, roughly speaking, if $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R_1$ we can easily extract r^* from D_{H} and the inverse of c, even if r^* is uniformly sampled and independent of the guessing adversary. Else if $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R_2$, we can also extract r^* from D_{G} and the inverse of c with a noticeable probability since there are at most q_{G} different r satisfying $D_{\mathsf{G}}(r) \neq \perp$. So with a negligible error we can transform game $\mathbf{G^a_{14}}$ into $\mathbf{G^b_{14}}$.

[8] In fact the "if and only if" here is not a rigorously accurate expression, since several preprocessing is needed to derive this equivalence relation. Here we just use this expression for convenience, and more details can be seen in the proof of Lemma 6.

– Based on game $\mathbf{G}_{\mathbf{14}}^{\mathbf{b}}$, we get game $\mathbf{G}_{\mathbf{17}}$ shown in Fig. 2 by modifying the oracles $\mathsf{G}[r^* \mapsto g^*]$ and $\mathsf{H}[s^* \mapsto h^*]$ queried during the generation of b' back to G and H, and removing the computation of g^* and h^*. First, we note that it is hard for the adversary $\mathcal{A}$ to detect the change of the oracles. The intuition is that, $\mathcal{A}$ can only detect such change by querying G on r^* or querying H on s^*. However, in game $\mathbf{G}_{\mathbf{17}}$, r^* is independent in the view of $\mathcal{A}$, and $\mathcal{A}$ can only "touch" s^* through $c^*(= f_{\mathsf{pk}}(s^*, t^*))$. As a result, the probability that $\mathcal{A}$ queries G on r^* must be negligible, meanwhile it is also hard for $\mathcal{A}$ to query H on s^*, otherwise we can use $\mathcal{A}$ to attack the PD-OW security of the underlying TDI. Naturally, after the modification of the oracles, g^* and h^* are no longer used by the game, hence we can directly remove their computations.

Now we complete the game hop from $\mathbf{G}_{\mathbf{14}}$ to $\mathbf{G}_{\mathbf{17}}$, meaning that we successfully reprogram $\mathsf{G}[r^* \mapsto g^*]$ and $\mathsf{H}[s^* \mapsto h^*]$ queried during the generation of b'.

1.3 Related Work

As a widely accepted generic transform in the ROM, OAEP has been extensively explored from various perspectives in the classical setting. Examples are [5] for designing more efficient variants of OAEP, [36,33] for weakening the underlying security assumptions used by the proof, and [7,2] for proving particular security goals going beyond IND-CCA. However, to the best of our knowledge, there are fewer works exploring OAEP in the quantum setting: in the QROM, except for the previously mentioned [37,13] which both focused on the fundamental IND-CCA/IND-qCCA security of OAEP, only one work [26] studied a particular "qIND-qCPA" security of OAEP. Therefore, there is still significant research potential for OAEP in the quantum setting.

In the QROM, different from [24,20,16], which proved the IND-CCA/IND-qCCA security of FO transform (or its variants) with the consideration of decryption error, all IND-CCA/IND-qCCA security proofs of OAEP in [37,13] and ours in this paper have not considered the decryption error. Roughly speaking, there are two main technical obstacles: first, the underlying scheme of OAEP is a trapdoor injective function, but as far as we know, there is currently no formal definition for the decryption error of trapdoor injective function; second, if the underlying trapdoor injective function is not perfectly correct, i.e., $c = f_{\mathsf{pk}}(s, t)$ may not imply $f_{\mathsf{sk}}^{-1}(c) = (s, t)$ for some $(\mathsf{pk}, \mathsf{sk}, s, t)$ tuples, the current simulation of the decryption oracle is no longer valid and it seems tricky to fix this up. Hence in this paper, we do not consider the decryption error and we leave it as an open question.

In addition, the Fixed Permutation One-Way to Hiding theorem applied in our proof is actually a very powerful and concise tool to handle the adaptive quantum reprogramming. In fact, [22] proved that this theorem not only implies the Pan-Zeng [32] and GHHM [17] adaptive reprogramming framework, but is also able to provide a comparable or essentially the same upper bound.

2 Preliminaries

2.1 Notation

Let n be a positive integer and then denote $[n]$ as the set $\{1, 2, \ldots, n\}$. Let s be a bitstring, denote $[s]_k$ as the k-least significant bits of s while $[s]^k$ denotes the k-most significant bits of s. $[\![P]\!]$ denotes a bit which is 1 if the predicate P is true and otherwise 0. $x \leftarrow_{\$} S$ denotes uniformly sampling an element x from a finite set S. $x \leftarrow D$ means that x is chosen according to distribution D. $\mathcal{T}_{\mathcal{A}}$ denotes the running time of algorithm or (computing) function $\mathcal{A}$. $\Pr[A : B]$ is the probability that predicate A keeps true where all variables in A are conditioned according to predicates B. During the security proof, we use $\Pr[1 \leftarrow \mathbf{G}]$ to denote the probability that the game $\mathbf{G}$ finally returns 1.

2.2 Quantum Background

In Supplementary Material B we present some fundamental quantum notions and principles utilized in this paper. For more detailed basics of quantum computation, we refer to a textbook [28].

2.3 Quantum Random Oracle Model

In the random oracle model (ROM), we assume that all parties have classical access to a random function. In practice, the random function is usually implemented by a suitable hash function. In the quantum setting, since the hash function can be locally computed in superposition, the ROM should be lifted to the quantum random oracle model (QROM) [6] where all parties have quantum access to the random oracle. In the QROM, the quantum query to a random oracle H is simulated by the unitary operation $\mathrm{O}_{\mathsf{H}} : |x, y\rangle \mapsto |x, y \oplus \mathsf{H}(x)\rangle$.

In the following, we present some definitions and techniques employed in the QROM focusing on security proof methodologies.

Definition 1 (Semi-Classical Oracle [1]). *For a subset $S \subseteq \mathcal{X}$, the indicator function f_S from $\mathcal{X}$ to $\{0,1\}$ equals 1 iff its input $x \in S$. The semi-classical oracle, $\mathcal{O}_S^{SC}$, is a quantum accessible oracle that performs the following two operations on its query state $\sum_{x\in\mathcal{X}} \alpha_x |x\rangle$:*

1. *Initialize a single qubit register L with the state $|0\rangle_{\mathsf{L}}$.*
2. *Transform $\sum_{x\in\mathcal{X}} \alpha_x |x\rangle|0\rangle_{\mathsf{L}}$ into $\sum_{x\in\mathcal{X}} \alpha_x |x\rangle|f_S(x)\rangle_{\mathsf{L}}$ and then measure L.*

During the execution of a quantum oracle algorithm $\mathcal{A}^{\mathcal{O}_S^{SC}}$, let Find be the classical event that the measurement of $\mathcal{O}_S^{SC}$ (on L) ever has outcome 1.

Theorem 1 (Semi-Classical One-Way to Hiding (O2H) [1, Theorem 1]). *Let $S \subseteq \mathcal{X}$ be random. Let $\mathsf{G}, \mathsf{H} : \mathcal{X} \to \mathcal{Y}$ be random functions satisfying $\forall x \notin S$, $\mathsf{G}(x) = \mathsf{H}(x)$. Let z be a random bitstring. ($\mathsf{G}, \mathsf{H}, S, z$ may have arbitrary joint distribution D.) Let $\mathsf{H}\backslash S$ be an oracle that first queries $\mathcal{O}_S^{SC}$ on the input register*

of H *and then queries* H*. Let* $\mathcal{A}$ *be a quantum oracle algorithm with query times* q[9] *(not necessarily unitary). Let*

$$P_{\text{left}} := \Pr[b = 1 : b \leftarrow \mathcal{A}^{\mathsf{H}}(z), (\mathsf{G}, \mathsf{H}, S, z) \leftarrow D],$$
$$P_{\text{right}} := \Pr[b = 1 \wedge \neg \mathsf{Find} : b \leftarrow \mathcal{A}^{\mathsf{H}\setminus S}(z), (\mathsf{G}, \mathsf{H}, S, z) \leftarrow D].$$

Let $P_{\text{find}} := \Pr[\mathsf{Find} : \mathcal{A}^{\mathsf{H}\setminus S}(z), (\mathsf{G}, \mathsf{H}, S, z) \leftarrow D]$*. Then*

$$|P_{\text{left}} - P_{\text{right}}| \leq \sqrt{(q+1) \cdot P_{\text{find}}} \text{ and } \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq \sqrt{(q+1) \cdot P_{\text{find}}}.$$

Remark 3. There are several other definitions of P_{right} in [1, Theorem 1]. In this paper, we only need the above definition in our security proofs.

Theorem 2 (Search in Semi-Classical Oracle [1, Theorem 2]). *Let* $S \subseteq \mathcal{X}$ *be random and* z *be a random bitstring.* (S, z) *may have an arbitrary joint distribution. Let* $\mathcal{A}$ *be any quantum oracle algorithm making at most* q *queries to a semi-classical oracle with domain* $\mathcal{X}$*.*

Let $\mathcal{B}$ *be an algorithm that on input* z*, chooses* $i \leftarrow_{\$} [q]$*; runs* $\mathcal{A}^{\mathcal{O}_{\varnothing}^{SC}}(z)$ *until (just before) the* i*-th query, measures the input register and outputs the measurement outcome* x*. Then*

$$\Pr[\mathsf{Find} : \mathcal{A}^{\mathcal{O}_S^{SC}}(z), (S, z) \leftarrow D] \leq 4q \cdot \Pr[x \in S : x \leftarrow \mathcal{B}(z), (S, z) \leftarrow D].$$

In the above theorem, for the simple case that z is independent of S, we have the following corollary.

Corollary 1 ([1, Corollary 1]). *Suppose that* S *and* z *are independent, and that* $\mathcal{A}$ *is a* q*-query algorithm. Let* $P_{\max} := \max_{x \in \mathcal{X}} \Pr[x \in S]$*, then* $\Pr[\mathsf{Find} : \mathcal{A}^{\mathcal{O}_S^{SC}}(z), (S, z) \leftarrow D] \leq 4q \cdot P_{\max}$*.*

2.4 Compressed Oracle Technique

We briefly introduce the compressed oracle technique presented by Zhandry in [41]. Roughly speaking, the core idea of this technique is to purify the query state of the quantum random oracle and then encode the oracle's internal state into a (efficiently controlled) database state. Here we first introduce the database model, and then introduce one particular version of the compressed oracle called the compressed standard oracle. For simplicity, we further assume that the compressed standard oracle will be queried at most q times.

Definition 2 (Database). *Define* $\mathcal{X} := \{0,1\}^m$ *and* $\mathcal{Y} := \{0,1\}^n$*. Introduce a symbol* $\perp$ *that* $\perp \notin \mathcal{X} \cup \mathcal{Y}$*. Then a database* D *is an ordered list of* q*-pairs as:*

$$((x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l), (\perp, 0^n), \ldots, (\perp, 0^n)),$$

[9] [1] actually considered the parallel queries and gave a better bound. In our paper, we do not consider such parallel queries for simplicity.

where $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ *for* $i \in [l], x_1 < \cdots < x_l$, *and all* $(\perp, 0^n)$ *pairs are at the end of the list*[10]. *The database set* $\mathbf{D}_q$ *is a set containing the above databases.*

For an $x \in \mathcal{X}$, we write $D(x) = y$ if there exists a $y \in \mathcal{Y}$ s.t. $(x, y) \in D$, otherwise write $D(x) = \perp$. Let $n(D)$ denote the number of (x, y) pairs in the database D such that $D(x) \neq \perp$. For any pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and any database D which satisfies $D(x) = \perp$ and $n(D) < q$, let $D \cup (x, y)$ be the updated database which first deletes a pair $(\perp, 0^n)$ and then inserts (x, y) appropriately into D to maintain its order. Conversely, for any pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and any database D which satisfies $D(x) = y$, let $D \backslash (x, y)$ denote the new database which first removes the pair (x, y) from D and then adds a pair $(\perp, 0^n)$ at the end.

Denote D_q as the quantum register defined over the database set $\mathbf{D}_q$. That is, D_q is the Hilbert space with orthonormal basis $\{|D\rangle\}_{D \in \mathbf{D}_q}$, where the basis state is labeled by elements in $\mathbf{D}_q$. Indeed, this basis is also called the computational basis by the basic quantum computation. In the following, we refer to D_q as the database register.

Define a superposition of the database state as

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

Then for any $x \in \mathcal{X}$, define the local decompression procedure $\mathsf{StdDecomp}_x$ applied on D_q as follows:

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

It is obvious that $\mathsf{StdDecomp}_x$ is unitary and is an involution, since applying it twice results in the identity operation. Meanwhile, by the above definition, it is easy to check that

$$\mathsf{StdDecomp}_x |D \cup (x, y)\rangle = |D \cup (x, y)\rangle + \frac{1}{\sqrt{2^n}} |D\rangle - \frac{1}{\sqrt{2^n}} |D \cup (x, \beta_{0^n})\rangle, \quad (1)$$

for a $D \in \mathbf{D}_q$ that $D(x) = \perp$ and $n(D) < q$, which is a very useful property.

Definition 3 (Compressed Standard Oracle). *Let* X *(resp.* Y*) denote the input (resp. output) register defined over* $\mathcal{X}$ *(resp.* $\mathcal{Y}$*). Let* $|D^{\perp}\rangle$ *be the initial state on database register* D_q*, where* $D^{\perp}$ *is the database containing* q $(\perp, 0^n)$ *pairs. A quantum query to the compressed standard oracle is implemented by performing the following unitary operation on registers* XYD_q.

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

[10] Here we only give the basic definition of the database, the quantum implementation details of the datadase can be found in [9,10].

Here $\mathsf{CNOT}^x|y, D\rangle_{\mathsf{YD}_q} = |y \oplus D(x), D\rangle_{\mathsf{YD}_q}$ *if* $D(x) \neq \perp$ *and* $|y, D\rangle_{\mathsf{YD}_q}$ *otherwise*[11].

Theorem 3 ([41, Lemma 4]). *For any adversary* $\mathcal{A}$ *making at most* q *oracle queries, the compressed standard oracle (Definition 3) and the quantum random oracle* $\mathsf{H} : \mathcal{X} \to \mathcal{Y}$ *are perfectly indistinguishable.*

Remark 4. As analyzed in [12, Theorem 4.3], using the compressed standard oracle to perfectly simulate the quantum random oracle with query upper bound q leads to an $\mathcal{O}(q^2)$ increase in running time.

3 The Security of OAEP Transform

In this section, we first review the structure of OAEP transform and then give its security theorem (Theorem 4). This theorem shows that, in the QROM, the IND-qCCA security of OAEP can be reduced to the PD-OW security of the underlying trapdoor injective function. We refer to Supplementary Material C for details of cryptographic primitives and security definitions.

OAEP Transform. Let $\mathsf{TDI} = (\mathsf{Gen}, f, f^{-1})$ be a trapdoor injective function with input space $\{0,1\}^{n+k_1} \times \{0,1\}^{k_0}$ and output space $\{0,1\}^m$. Define a parameter $\mathcal{T}_f := \max_{(\mathsf{pk},\mathsf{sk})} \mathcal{T}_{f_{\mathsf{pk}}}$, where the maximum is according to all the $(\mathsf{pk}, \mathsf{sk})$ sampled by Gen, and $\mathcal{T}_{f_{\mathsf{pk}}}$ denotes the running time of computing f_{pk}. Let $\mathsf{G} : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ and $\mathsf{H} : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ be hash functions. We associate the randomized PKE scheme

$$\mathsf{OAEP}[\mathsf{TDI}, \mathsf{G}, \mathsf{H}] = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}).$$

This scheme has message space $\mathcal{M} = \{0,1\}^n$, randomness space $\mathcal{R} = \{0,1\}^{k_0}$ and ciphertext space $\mathcal{C} = \{0,1\}^m$. The constituting algorithms of this scheme are shown in Fig. 6.

Gen	$\mathsf{Enc}(\mathsf{pk}, m \in \{0,1\}^n)$	$\mathsf{Dec}(\mathsf{sk}, c)$
1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$	1 : $r \leftarrow_{\$} \{0,1\}^{k_0}$	1 : **if** $f_{\mathsf{sk}}^{-1}(c) = \perp$, **return** $\perp$
2 : **return** $(\mathsf{pk}, \mathsf{sk})$	2 : $s = (m \Vert 0^{k_1}) \oplus \mathsf{G}(r)$	2 : **else** parse $f_{\mathsf{sk}}^{-1}(c)$ into $s \Vert t$
	3 : $t = r \oplus \mathsf{H}(s)$	3 : $r = t \oplus \mathsf{H}(s), M = s \oplus \mathsf{G}(r)$
	4 : $c = f_{\mathsf{pk}}(s, t)$	4 : **if** $[M]_{k_1} = 0^{k_1}$, **return** $[M]^n$
	5 : **return** c	5 : **else return** $\perp$

Fig. 6. Public key encryption scheme $\mathsf{OAEP}[\mathsf{TDI}, \mathsf{G}, \mathsf{H}]$.

[11] After performing CNOT^x, the state on YD_q remains unchanged if $D(x) = \perp$. In fact, this is equivalent to the definition that "$y \oplus \perp = y$" defined in [41].

Theorem 4 (PD-OW of TDI $\overset{\text{QROM}}{\Rightarrow}$ IND-qCCA of OAEP[TDI, G, H]). *Let $\mathcal{A}$ be an* IND-qCCA *adversary against* OAEP[TDI, G, H] *in the QROM, making at most q_{D}, q_{G} and q_{H} quantum queries to the decryption oracle* qDec*, random oracle* G *and random oracle* H*. Let $q := q_{\mathsf{G}} + q_{\mathsf{H}} + 1$. Then we can construct a* PD-OW *adversary $\mathcal{B}$ against the trapdoor injective function* TDI *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{OAEP[TDI,G,H]},\mathcal{A}} \leq & (32q_{\mathsf{D}}+4)q \cdot \sqrt{\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}}} + (42q_{\mathsf{D}}+8)q \cdot \sqrt{2^{-k_0}} \\ & + 24q_{\mathsf{D}} \cdot \sqrt{2^{-k_1}},\end{aligned}$$

and $\mathcal{T}_{\mathcal{B}} \leq \mathcal{T}_{\mathcal{A}} + \mathcal{O}(q_{\mathsf{G}}^2 + q_{\mathsf{H}}^2) + \mathcal{O}(q_{\mathsf{D}} q_{\mathsf{G}} q_{\mathsf{H}}) \cdot \mathcal{T}_f$.

3.1 Quantum Accessible Decryption Oracle qDec

Before proving Theorem 4, we first show how to simulate the quantum accessible decryption oracle qDec.

Denote I/O as the input/output registers of qDec, where I is defined over $\{0,1\}^m$ and O over $\{0,1\}^n \cup \{\perp\}$[12]. Let S and T be two ancillary registers, which are defined over $\{0,1\}^{n+k_1} \cup \{\perp\}$ and $\{0,1\}^{k_0} \cup \{\perp\}$, respectively. Now we define the unitary operation $\mathrm{U}_{\mathsf{qDec}}$ acting on registers I, O, S and T:

$$\mathrm{U}_{\mathsf{qDec}} := \mathrm{U}_{f^{-1}} \circ \mathrm{O}_{\mathsf{H}} \circ \mathrm{O}_{\mathsf{G}} \circ \mathrm{U}_{\mathrm{check}} \circ \mathrm{O}_{\mathsf{G}} \circ \mathrm{O}_{\mathsf{H}} \circ \mathrm{U}_{f^{-1}}. \tag{2}$$

Here

$$\mathrm{U}_{f^{-1}} : |c,s,t\rangle_{\mathsf{IST}} \mapsto \begin{cases} |c, s \oplus s', t \oplus t'\rangle_{\mathsf{IST}}, & if\ f_{\mathsf{sk}}^{-1}(c) \neq \perp \text{ and } f_{\mathsf{sk}}^{-1}(c) = s'||t', \\ |c, s \oplus \perp, t \oplus \perp\rangle_{\mathsf{IST}}, & otherwise. \end{cases}$$

focuses on lifting the step 1 and 2 of $\mathsf{Dec}(\mathsf{sk}, c)$ (Fig. 6) to the quantum setting. $\mathrm{O}_{\mathsf{H}} : |x,y\rangle \mapsto |x, y \oplus \mathsf{H}(x)\rangle$ (resp. $\mathrm{O}_{\mathsf{G}} : |x,y\rangle \mapsto |x, y \oplus \mathsf{G}(x)\rangle$) with input/output registers S/T (resp. T/S) focuses on lifting the step 3 of $\mathsf{Dec}(\mathsf{sk}, c)$. As for the $\mathrm{U}_{\mathrm{check}}$, it is defined as:

$$\mathrm{U}_{\mathrm{check}} : |M,y\rangle_{\mathsf{SO}} \mapsto \begin{cases} |M, y \oplus [M]^n\rangle_{\mathsf{SO}}, & if\ M \in \{0,1\}^{n+k_1} \wedge [M]_{k_1} = 0^{k_1}, \\ |M, y \oplus \perp\rangle_{\mathsf{SO}}, & otherwise. \end{cases}$$

and it is used to lift the step 4 and step 5 of $\mathsf{Dec}(\mathsf{sk}, c)$ to the quantum setting. Note that the implementation of $\mathrm{U}_{f^{-1}}$ may introduce $s \oplus \perp$ on register S, which indicates that the computation has failed. So in the above definition of $\mathrm{U}_{\mathsf{check}}$, we impose the restriction $M \in \{0,1\}^{n+k_1}$ to exclude such failure cases. This exclusion is valid because, as shown in Supplementary Material B, our quantum representation of $\perp$ ensures that $s \oplus \perp \notin 0,1^{n+k_1}$ for any $s \in 0,1^{n+k_1}$. Additionally, since $\mathrm{U}_{f^{-1}}$ can also introduce $t \oplus \perp$ on register T, WLOG we assume $\mathsf{G}(\perp) = \mathsf{H}(\perp) = 0$ in the definition of O_{G} and O_{H} above.

[12] Here and in the following, we always use symbol $\perp$ to denote "computational failure". See Supplementary Material B for a detailed quantum representation of this symbol.

At this point, one can easily check that

$$\mathrm{U}_{\mathsf{qDec}}|c, y, 0, 0\rangle_{\mathsf{IOST}} = |c, y \oplus \mathsf{Dec}(\mathsf{sk}, c), 0, 0\rangle_{\mathsf{IOST}}.$$

So if we initialize registers S and T to the state $|0\rangle$ at the beginning and ensure that they are not accessed by any other operations, then their states will remain $|0\rangle$ before and after each implementation of $\mathrm{U}_{\mathsf{qDec}}$. Hence, for $(\mathsf{pk}, \mathsf{sk})$ sampled from Gen, let S and T be the internal registers of the quantum-accessible decryption oracle qDec. By initializing both registers to $|0\rangle$, we can directly simulate qDec using $\mathrm{U}_{\mathsf{qDec}}$ before the challenge ciphertext c^* is defined. After c^* is defined, we can use a controlled version of $\mathrm{U}_{\mathsf{qDec}}$ to simulate qDec, which is defined as:

$$\mathrm{U}^*_{\mathsf{qDec}} := \mathrm{U}_{\perp} \circ \mathrm{P}_{c^*} + \mathrm{U}_{\mathsf{qDec}} \circ (\mathrm{I} - \mathrm{P}_{c^*}). \tag{3}$$

Here $\mathrm{P}_{c^*} := |c^*\rangle\langle c^*|$ is a projector acting on register I, and $\mathrm{U}_{\perp}$ is a unitary operation acting on the register O which maps $|y\rangle$ to $|y \oplus \perp\rangle$ for all y. I is the identity operator.

4 Proof of Theorem 4

We prove Theorem 4 by a sequence of games $\mathbf{G}_0$ to $\mathbf{G}_{17}$ as shown in Fig. 7, Fig. 8 and Fig. 9. These games can be divided into the following three parts:

1. Games $\mathbf{G}_0$ to $\mathbf{G}_5$ analyzed in Section 4.1. This part focuses on changing the simulation of the quantum accessible decryption oracle qDec until the secret key sk is no longer needed.
2. Games $\mathbf{G}_5$ to $\mathbf{G}_{14}$ analyzed in Section 4.2. This part focuses on reprogramming the oracles G and H queried before $\mathcal{A}$ outputs (m_0^*, m_1^*).
3. Games $\mathbf{G}_{14}$ to $\mathbf{G}_{17}$ analyzed in Section 4.3. This part focuses on reprogramming the oracles G and H queried during the generation of b'.

After completing these three parts, we finally get a trivial game $\mathbf{G}_{17}$ that outputs 1 with probability 1/2, and combining the security bounds derived in each part, we obtain the security bound claimed in Theorem 4.

In the following proof, for simplicity, we may choose to include or omit the subscripts that denote the quantum registers of a quantum state, e.g., we may simplify $|c\rangle_{\mathsf{I}}$ to $|c\rangle$. Generally, we will include the subscripts when they are crucial to our proof; otherwise, we may choose to omit them.

Game $\mathbf{G}_0$: This is the $\mathsf{IND\text{-}qCCA}$ game of $\mathsf{OAEP}[\mathsf{TDI}, \mathsf{G}, \mathsf{H}]$ with the adversary $\mathcal{A}$ in the QROM. Before the challenge ciphertext c^* is defined, the oracle qDec is simulated by $\mathrm{U}_{\mathsf{qDec}}$ (Eq. (2)). After c^* is defined, the oracle qDec is simulated by $\mathrm{U}^*_{\mathsf{qDec}}$ (Eq. (3)). Obviously, we have

$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{OAEP}[\mathsf{TDI},\mathsf{G},\mathsf{H}],\mathcal{A}} = \left|\Pr[1 \leftarrow \mathbf{G}_0] - \frac{1}{2}\right|. \tag{4}$$

4.1 Change the Simulation of qDec

In this section, based on game $\mathbf{G_0}$, we introduce games $\mathbf{G_1}$ to $\mathbf{G_5}$ as shown in Fig. 7 to change the simulation of the quantum accessible decryption oracle qDec. In game $\mathbf{G_5}$, the simulation of qDec no longer requires the secret key sk.

GAMES $\mathbf{G_0}$ to $\mathbf{G_5}$

1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$ //$\mathbf{G_0}$-$\mathbf{G_5}$
   $\mathsf{G} \leftarrow_{\$} \Omega_{\mathsf{G}}, \mathsf{H} \leftarrow_{\$} \Omega_{\mathsf{H}}$ //$\mathbf{G_0}$-$\mathbf{G_2}$
2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}}(\mathsf{pk})$ //$\mathbf{G_0}$-$\mathbf{G_5}$
3. $b \leftarrow_{\$} \{0,1\}$ //$\mathbf{G_0}$-$\mathbf{G_5}$
   $r^* \leftarrow_{\$} \{0,1\}^{k_0}$ //$\mathbf{G_0}$-$\mathbf{G_5}$
   $s^* = (m_b^* || 0^{k_1}) \oplus \mathsf{G}(r^*)$ //$\mathbf{G_0}$-$\mathbf{G_5}$
   $t^* = r^* \oplus \mathsf{H}(s^*)$ //$\mathbf{G_0}$-$\mathbf{G_5}$
   $c^* = f_{\mathsf{pk}}(s^*, t^*)$ //$\mathbf{G_0}$-$\mathbf{G_5}$
4. $b' \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}}(\mathsf{pk}, c^*)$ //$\mathbf{G_0}$-$\mathbf{G_5}$
5. **return** $[\![b' = b]\!]$ //$\mathbf{G_0}$-$\mathbf{G_5}$

$\mathsf{G}(|x_{\mathsf{G}}, y_{\mathsf{G}}\rangle)$

apply $|x_{\mathsf{G}}, y_{\mathsf{G}}\rangle \mapsto |x_{\mathsf{G}}, y_{\mathsf{G}} \oplus \mathsf{G}(x_{\mathsf{G}})\rangle$ //$\mathbf{G_0}$
apply $\mathsf{CStO}_{\mathsf{G}}$ //$\mathbf{G_1}$-$\mathbf{G_5}$

$\mathsf{H}(|x_{\mathsf{H}}, y_{\mathsf{H}}\rangle)$

apply $|x_{\mathsf{H}}, y_{\mathsf{H}}\rangle \mapsto |x_{\mathsf{H}}, y_{\mathsf{H}} \oplus \mathsf{H}(x_{\mathsf{H}})\rangle$ //$\mathbf{G_0}$-$\mathbf{G_2}$
apply $\mathsf{CStO}_{\mathsf{H}}$ //$\mathbf{G_3}$-$\mathbf{G_5}$

$\mathsf{qDec}(|c, y\rangle)$

if c^* has not been defined
  apply $\mathrm{U}_{\mathsf{qDec}}$ //$\mathbf{G_0}$
  apply $\mathrm{U}_{\mathsf{Search}}$ //$\mathbf{G_5}$
else apply $\mathrm{U}^*_{\mathsf{qDec}}$ //$\mathbf{G_0}$
    apply $\mathrm{U}^*_{\mathsf{Search}}$ //$\mathbf{G_5}$

$\mathsf{qDec}(|c, y\rangle)$

if c^* has not been defined
  apply $\mathrm{U}^i_{\mathsf{qDec}}$ //$\mathbf{G}_i, i \in [4]$
else apply $\mathrm{U}^{i,*}_{\mathsf{qDec}}$ //$\mathbf{G}_i, i \in [4]$

Fig. 7. Games $\mathbf{G}_0$ to $\mathbf{G}_5$ in the proof of Theorem 4. Here Ω_{G} and Ω_{H} denote the set of all functions $\mathsf{G} : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ and $\mathsf{H} : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$, respectively.

Game $\mathbf{G_1}$: This game is the same as game $\mathbf{G_0}$, except that G is simulated as a compressed standard oracle with the database register $\mathsf{D}_{q_{\mathsf{G}}}$ (Definition 3). Denote $\mathsf{CStO}_{\mathsf{G}}$ as the unitary operation used to implement that compressed standard oracle, then in this game, the oracle qDec is simulated by

$$\mathrm{U}^1_{\mathsf{qDec}} := \mathrm{U}_{f^{-1}} \circ \mathrm{O}_{\mathsf{H}} \circ \mathsf{CStO}_{\mathsf{G}} \circ \mathrm{U}_{\mathrm{check}} \circ \mathsf{CStO}_{\mathsf{G}} \circ \mathrm{O}_{\mathsf{H}} \circ \mathrm{U}_{f^{-1}}, \tag{5}$$

and $\mathrm{U}^{1,*}_{\mathsf{qDec}} := \mathrm{U}_{\perp} \circ \mathrm{P}_{c^*} + \mathrm{U}^1_{\mathsf{qDec}} \circ (\mathrm{I} - \mathrm{P}_{c^*})$ before and after c^* is defined, respectively. Since the adversary $\mathcal{A}$ makes at most q_{G} queries to G, the two G oracles in games $\mathbf{G_0}$ and $\mathbf{G_1}$ are perfectly indistinguishable by Theorem 3, and thus

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

Game $\mathbf{G_2}$: This game is the same as game $\mathbf{G_1}$, except that $\mathrm{U}^1_{\mathsf{qDec}}$ is replaced by

$$\mathrm{U}^2_{\mathsf{qDec}} := \mathrm{U}_{f^{-1}} \circ \mathrm{O}_{\mathsf{H}} \circ \mathsf{Test}_{\mathsf{G}} \circ \mathrm{U}_{\mathrm{check}} \circ \mathsf{Test}_{\mathsf{G}} \circ \mathrm{O}_{\mathsf{H}} \circ \mathrm{U}_{f^{-1}}, \tag{7}$$

and $\mathrm{U}^{1,*}_{\mathsf{qDec}}$ is replaced by $\mathrm{U}^{2,*}_{\mathsf{qDec}} := \mathrm{U}_{\perp} \circ \mathrm{P}_{c^*} + \mathrm{U}^{2}_{\mathsf{qDec}} \circ (\mathrm{I} - \mathrm{P}_{c^*})$. Here $\mathsf{Test}_{\mathsf{G}}$ is a new unitary operation acting on the registers S, T and $\mathsf{D}_{q_{\mathsf{G}}}$ as

$$\mathsf{Test}_{\mathsf{G}} : |s, t, D_{\mathsf{G}}\rangle \mapsto \begin{cases} |s \oplus D_{\mathsf{G}}(t), t, D_{\mathsf{G}}\rangle, & if\ D_{\mathsf{G}}(t) \neq \perp, \\ |s \oplus \perp, t, D_{\mathsf{G}}\rangle, & otherwise. \end{cases}$$

In game $\mathbf{G_2}$, since the $\mathsf{Test}_{\mathsf{G}}$ does not change the database state $|D_{\mathsf{G}}\rangle$ in the computational basis and only "read" some information from D_{G}, the simulation of qDec will no longer update the (computational basis) state of the database register $\mathsf{D}_{q_{\mathsf{G}}}$. As for the difference between game $\mathbf{G_1}$ and $\mathbf{G_2}$, we can prove the following lemma, and the detailed proof is given in Supplementary Material D.1.

Lemma 1. $|\Pr[1 \leftarrow \mathbf{G_1}] - \Pr[1 \leftarrow \mathbf{G_2}]| \leq 8q_{\mathsf{D}} \cdot \sqrt{2^{-k_1}}$.

Game $\mathbf{G_3}$: This game is the same as game $\mathbf{G_2}$, except that H is simulated as a compressed standard oracle with the database register $\mathsf{D}_{q_{\mathsf{H}}}$ (Definition 3). Denote $\mathsf{CStO}_{\mathsf{H}}$ as the unitary operation used to implement that compressed standard oracle, then in this game, the oracle qDec is simulated by

$$\mathrm{U}^{3}_{\mathsf{qDec}} := \mathrm{U}_{f^{-1}} \circ \mathsf{CStO}_{\mathsf{H}} \circ \mathsf{Test}_{\mathsf{G}} \circ \mathrm{U}_{\mathrm{check}} \circ \mathsf{Test}_{\mathsf{G}} \circ \mathsf{CStO}_{\mathsf{H}} \circ \mathrm{U}_{f^{-1}}, \tag{8}$$

and $\mathrm{U}^{3,*}_{\mathsf{qDec}} := \mathrm{U}_{\perp} \circ \mathrm{P}_{c^*} + \mathrm{U}^{3}_{\mathsf{qDec}} \circ (\mathrm{I} - \mathrm{P}_{c^*})$ before and after c^* is defined, respectively.

Similar to the analysis of the game hop from $\mathbf{G_0}$ to $\mathbf{G_1}$, we have

$$\Pr[1 \leftarrow \mathbf{G_2}] = \Pr[1 \leftarrow \mathbf{G_3}]. \tag{9}$$

Game $\mathbf{G_4}$: This game is the same as game $\mathbf{G_3}$, except that $\mathrm{U}^{3}_{\mathsf{qDec}}$ is replaced by

$$\mathrm{U}^{4}_{\mathsf{qDec}} := \mathrm{U}_{f^{-1}} \circ \mathsf{Test}_{\mathsf{H}} \circ \mathsf{Test}_{\mathsf{G}} \circ \mathrm{U}_{\mathrm{check}} \circ \mathsf{Test}_{\mathsf{G}} \circ \mathsf{Test}_{\mathsf{H}} \circ \mathrm{U}_{f^{-1}}, \tag{10}$$

and $\mathrm{U}^{3,*}_{\mathsf{qDec}}$ is replaced by $\mathrm{U}^{4,*}_{\mathsf{qDec}} := \mathrm{U}_{\perp} \circ \mathrm{P}_{c^*} + \mathrm{U}^{4}_{\mathsf{qDec}} \circ (\mathrm{I} - \mathrm{P}_{c^*})$. Here $\mathsf{Test}_{\mathsf{H}}$ is a new unitary operation acting on the registers S, T and $\mathsf{D}_{q_{\mathsf{H}}}$ as

$$\mathsf{Test}_{\mathsf{H}} : |s, t, D_{\mathsf{H}}\rangle \mapsto \begin{cases} |s, t \oplus D_{\mathsf{H}}(s), D_{\mathsf{H}}\rangle, & if\ D_{\mathsf{H}}(s) \neq \perp, \\ |s, t \oplus \perp, D_{\mathsf{H}}\rangle, & otherwise. \end{cases}$$

In the simulation of qDec in game $\mathbf{G_4}$, both the compressed standard oracle implementation of G and H are changed into a direct database "read" (i.e., $\mathsf{Test}_{\mathsf{G}}$ and $\mathsf{Test}_{\mathsf{H}}$). Thus the simulation of qDec in game $\mathbf{G_4}$ will no longer update the (computational basis) state on the database registers $\mathsf{D}_{q_{\mathsf{G}}}$ and $\mathsf{D}_{q_{\mathsf{H}}}$. As for the difference between game $\mathbf{G_3}$ and $\mathbf{G_4}$, similar to the proof strategy of Lemma 1, we can prove the following lemma, and the detailed proof is given in Supplementary Material D.3.

Lemma 2. $|\Pr[1 \leftarrow \mathbf{G_3}] - \Pr[1 \leftarrow \mathbf{G_4}]| \leq 8q_{\mathsf{D}}\sqrt{q_{\mathsf{G}}} \cdot \sqrt{2^{-k_0}}$.

Game $\mathbf{G_5}$: This game is the same as game $\mathbf{G_4}$, except that $\mathrm{U}^4_{\mathsf{qDec}}$ is replaced by

$$\mathrm{U}_{\mathsf{Search}} : |c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle \mapsto |c, y \oplus \mathsf{Search}(\mathsf{pk}, c, D_{\mathsf{G}}, D_{\mathsf{H}}), D_{\mathsf{G}}, D_{\mathsf{H}}\rangle,$$

which acts on the registers $\mathsf{I}, \mathsf{O}, \mathsf{D}_{q_{\mathsf{H}}}$ and $\mathsf{D}_{q_{\mathsf{G}}}$, and $\mathrm{U}^{4,*}_{\mathsf{qDec}}$ is replaced by $\mathrm{U}^*_{\mathsf{Search}} := \mathrm{U}_{\perp} \circ \mathrm{P}_{c^*} + \mathrm{U}_{\mathsf{Search}} \circ (\mathrm{I} - \mathrm{P}_{c^*})$. Here Search is a classical function defined as

$$\mathsf{Search}\,(\mathsf{pk}, c, D_{\mathsf{G}}, D_{\mathsf{H}}) = \begin{cases} [s \oplus D_{\mathsf{G}}(r)]^n & \exists\,(s, D_{\mathsf{H}}(s))\; and\; (r, D_{\mathsf{G}}(r))\; s.t. \\ & f_{\mathsf{pk}}(s, r \oplus D_{\mathsf{H}}(s)) = c \wedge [s]_{k_1} = [D_{\mathsf{G}}(r)]_{k_1}, \\ \perp & otherwise. \end{cases} \tag{11}$$

One can notice that the computation of Search does not use the secret key sk and only uses the public key pk to compute $f_{\mathsf{pk}}(\cdot)$. So the simulation of qDec in game $\mathbf{G_5}$ no longer uses the secret key sk. Next, in Supplementary Material D.4 we prove that for any joint state $|\Phi\rangle$ just before one decryption query on registers $\mathsf{I}, \mathsf{O}, \mathsf{D}_{q_{\mathsf{H}}}$ and $\mathsf{D}_{q_{\mathsf{G}}}$, the following equations hold:

$$\mathrm{U}^4_{\mathsf{qDec}}|\Phi\rangle|0\rangle_{\mathsf{ST}} = \mathrm{U}_{\mathsf{Search}}|\Phi\rangle|0\rangle_{\mathsf{ST}} \text{ and } \mathrm{U}^{4,*}_{\mathsf{qDec}}|\Phi\rangle|0\rangle_{\mathsf{ST}} = \mathrm{U}^*_{\mathsf{Search}}|\Phi\rangle|0\rangle_{\mathsf{ST}}.$$

So it is obvious that adversary $\mathcal{A}$ has an identical view between game $\mathbf{G_4}$ and $\mathbf{G_5}$. Hence, for the difference between game $\mathbf{G_4}$ and $\mathbf{G_5}$, we have

$$\Pr[1 \leftarrow \mathbf{G_4}] = \Pr[1 \leftarrow \mathbf{G_5}]. \tag{12}$$

Now, by combining Eq. (6), (9), (12), Lemma 1 and Lemma 2, we obtain

$$|\Pr[1 \leftarrow \mathbf{G_0}] - \Pr[1 \leftarrow \mathbf{G_5}]| \leq 8q_{\mathsf{D}} \cdot \sqrt{2^{-k_1}} + 8q_{\mathsf{D}}\sqrt{q_{\mathsf{G}}} \cdot \sqrt{2^{-k_0}}. \tag{13}$$

4.2 Reprogram G and H Queried before $\mathcal{A}$ Outputs (m_0^*, m_1^*)

Here we introduce games $\mathbf{G_6}$ to $\mathbf{G_{14}}$ as shown in Fig. 8 to reprogram oracles G and H queried before $\mathcal{A}$ outputs (m_0^*, m_1^*). After this reprogramming, the $\mathsf{G}(r^*)$ and $\mathsf{H}(s^*)$ used to generate the challenge ciphertext c^* are independent of oracles G and H queried before $\mathcal{A}$ outputs (m_0^*, m_1^*)[13]. Hence, from now on, we can focus only on reprogramming the oracles G and H queried during the generation of b', which is actually the topic of the next section.

Game $\mathbf{G_6}$: Compared with game $\mathbf{G_5}$, this new game has the following two changes:

- Before $\mathcal{A}$ outputs (m_0^*, m_1^*), G is simulated by $\mathsf{CStO}_{\mathsf{G}} \circ \mathcal{O}^{SC}_{\{r^*\}}$, which is an operation that first queries the semi-classical oracle $\mathcal{O}^{SC}_{\{r^*\}}$ (Definition 1) on the input register of G and then performs $\mathsf{CStO}_{\mathsf{G}}$.
- If Find_{r^*} occurs before $\mathcal{A}$ outputs (m_0^*, m_1^*), the game finally returns $\perp$. Here Find_{r^*} is the event that the measurement of $\mathcal{O}^{SC}_{\{r^*\}}$ ever has result 1.

We first construct a quantum oracle algorithm $\mathcal{A}_1$ as follows, which has oracle access to a constant zero function F.

[13] But not independent of the oracles G and H queried during the generation of b'.

GAMES $\mathbf{G_5}$ to $\mathbf{G_{14}}$

1. $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}, r^* \leftarrow_\$ \{0,1\}^{k_0}$ //$\mathbf{G_5}$-$\mathbf{G_{14}}$
   $g^* \leftarrow_\$ \{0,1\}^{n+k_1}$ //$\mathbf{G_7}$
   $s^* \leftarrow_\$ \{0,1\}^{n+k_1}$ //$\mathbf{G_8}$-$\mathbf{G_{14}}$
   $h^* \leftarrow_\$ \{0,1\}^{k_0}$ //$\mathbf{G_{10}}$
   $t^* \leftarrow_\$ \{0,1\}^{k_0}$ //$\mathbf{G_{11}}$-$\mathbf{G_{14}}$
2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}}(\mathsf{pk})$ //$\mathbf{G_5}$-$\mathbf{G_{14}}$
3. $b \leftarrow_\$ \{0,1\}$ //$\mathbf{G_5}$-$\mathbf{G_{14}}$
   $s^* = (m_b^* || 0^{k_1}) \oplus \mathsf{G}(r^*)$ //$\mathbf{G_5}$-$\mathbf{G_6}$
   $s^* = (m_b^* || 0^{k_1}) \oplus g^*$ //$\mathbf{G_7}$
   perform $\mathsf{StdDecomp}_{r^*} \circ \mathrm{U}_{g^*}^{r^*}$ //$\mathbf{G_7}$-$\mathbf{G_{12}}$
   perform $\mathrm{U}_{g^*}^{r^*}$ //$\mathbf{G_{13}}$-$\mathbf{G_{14}}$
   $g^* = (m_b^* || 0^{k_1}) \oplus s^*$ //$\mathbf{G_8}$-$\mathbf{G_{14}}$
   $t^* = r^* \oplus \mathsf{H}(s^*)$ //$\mathbf{G_5}$-$\mathbf{G_9}$
   $t^* = r^* \oplus h^*$ //$\mathbf{G_{10}}$
   $h^* = r^* \oplus t^*$ //$\mathbf{G_{11}}$-$\mathbf{G_{14}}$
   perform $\mathsf{StdDecomp}_{s^*} \circ \mathrm{U}_{h^*}^{s^*}$ //$\mathbf{G_{10}}$-$\mathbf{G_{12}}$
   perform $\mathrm{U}_{h^*}^{s^*}$ //$\mathbf{G_{13}}$-$\mathbf{G_{14}}$
   $c^* = f_{\mathsf{pk}}(s^*, t^*)$ //$\mathbf{G_5}$-$\mathbf{G_{14}}$
4. $b' \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}}(\mathsf{pk}, c^*)$ //$\mathbf{G_5}$-$\mathbf{G_{14}}$
5. **return** $[\![b' = b]\!]$ //$\mathbf{G_5}$-$\mathbf{G_{14}}$

$\mathsf{G}(|x_\mathsf{G}, y_\mathsf{G}\rangle)$

if (m_0^*, m_1^*) has not been computed
   apply CStO_G //$\mathbf{G_5}, \mathbf{G_{14}}$
   apply $\mathsf{CStO}_\mathsf{G} \circ \mathcal{O}^{SC}_{\{r^*\}}$ //$\mathbf{G_6}$-$\mathbf{G_{13}}$
      if Find_{r^*} occurs, **return** $\perp$
else
   apply CStO_G //$\mathbf{G_5}$-$\mathbf{G_{11}}$
   apply $\mathsf{CStO}_\mathsf{G}^{r^*,g^*}$ //$\mathbf{G_{12}}$-$\mathbf{G_{14}}$

$\mathsf{H}(|x_\mathsf{H}, y_\mathsf{H}\rangle)$

if (m_0^*, m_1^*) has not been computed
   apply CStO_H //$\mathbf{G_5}$-$\mathbf{G_8}, \mathbf{G_{14}}$
   apply $\mathsf{CStO}_\mathsf{H} \circ \mathcal{O}^{SC}_{\{s^*\}}$ //$\mathbf{G_9}$-$\mathbf{G_{13}}$
      if Find_{s^*} occurs, **return** $\perp$
else
   apply CStO_H //$\mathbf{G_5}$-$\mathbf{G_{11}}$
   apply $\mathsf{CStO}_\mathsf{H}^{s^*,h^*}$ //$\mathbf{G_{12}}$-$\mathbf{G_{14}}$

$\mathsf{qDec}(|c, y\rangle)$

if c^* has not been computed
   apply $\mathrm{U}_{\mathsf{Search}}$ //$\mathbf{G_5}$-$\mathbf{G_{14}}$
else apply $\mathrm{U}^*_{\mathsf{Search}}$ //$\mathbf{G_5}$-$\mathbf{G_{14}}$

Fig. 8. Games $\mathbf{G}_5$ to $\mathbf{G}_{14}$ in the proof of Theorem 4. The game $\mathbf{G}_5$ shows in this figure modifies the random selection of r^* to be placed just after the generation of $(\mathsf{pk},\mathsf{sk})$, which is only a conceptual change compared with the game $\mathbf{G}_5$ shown in Fig. 7.

– $\mathcal{A}_1$ runs game $\mathbf{G_5}$. Specifically, when $\mathcal{A}$ queries G using an (informal) query state $\sum |x,y\rangle_{\mathsf{X_G Y_G}}$ before $\mathcal{A}$ outputs (m_0^*, m_1^*), $\mathcal{A}_1$ will perform an additional operation: first expand this query state into $\sum |x,y\rangle \otimes_{\mathsf{X_G Y_G}} |0\rangle_{\mathsf{O_F}}$ and then forward the input/output register $\mathsf{X_G}/\mathsf{O_F}$ to its function oracle F.

Obviously $\mathcal{A}_1$ makes at most q_G queries to its oracle, and such additional queries do not disturb $\mathcal{A}$'s computation since F is a constant zero function. Hence,

$$\Pr[1 \leftarrow \mathbf{G_5}] = \Pr[b = 1 : b \leftarrow \mathcal{A}_1^{\mathsf{F}}(r^*, z), r^* \leftarrow_\$ \{0,1\}^{k_0}, z \leftarrow D_{\mathcal{A}_1}].$$

Here we particularly write r^* and assume (r^*, z) as the classical input of $\mathcal{A}_1$ without loss of generality, where r^* is uniformly sampled from $\{0,1\}^{k_0}$ and z is sampled according to a distribution $D_{\mathcal{A}_1}$. Then, if we change F into $\mathsf{F}\backslash\{r^*\}$, i.e., the oracle that first queries $\mathcal{O}^{SC}_{\{r^*\}}$ on the input register of F and then queries F, it is easy to check that

$$\Pr[1 \leftarrow \mathbf{G_6}] = \Pr[(b = 1) \wedge \neg \mathsf{Find}_{r^*} : b \leftarrow \mathcal{A}_1^{\mathsf{F}\backslash\{r^*\}}(r^*, z), r^* \leftarrow_\$ \{0,1\}^{k_0}, z \leftarrow D_{\mathcal{A}_1}].$$

Let D be a joint distribution of $(\mathsf{E}, \mathsf{F}, \{r^*\}, z)$, where E, F are both constant zero functions[14], $r^* \leftarrow_\$ \{0,1\}^{k_0}$ and $z \leftarrow D_{\mathcal{A}_1}$. Then by applying Theorem 1 with this joint distribution D, we have

$$\begin{aligned} |\Pr[1 \leftarrow \mathbf{G_5}] - \Pr[1 \leftarrow \mathbf{G_6}]| &\leq \sqrt{(q_\mathsf{G}+1) \cdot \Pr[\mathsf{Find}_{r^*} : \mathcal{A}_1^{\mathsf{F}\backslash\{r^*\}}, (\mathsf{E}, \mathsf{F}, \{r^*\}, z) \leftarrow D]} \\ &= \sqrt{(q_\mathsf{G}+1) \cdot \Pr[\mathsf{Find}_{r^*} : \mathbf{G_6}]}. \end{aligned} \tag{14}$$

The above final equation follows from the definition of $\mathcal{A}_1$ and game $\mathbf{G_6}$.

Game $\mathbf{G_7}$: This game is the same as game $\mathbf{G_6}$ except that the computation of s^* is changed. In this game, s^* is generated as $(m_b^* || 0^{k_1}) \oplus g^*$, where g^* is a new bitstring uniformly sampled from $\{0,1\}^{n+k_1}$ at the beginning of the game. And after the generation of s^*, the $\mathrm{U}^{r^*}_{g^*}$ and $\mathsf{StdDecomp}_{r^*}$ are sequentially performed on the database register $\mathsf{D}_{q_\mathsf{G}}$, where $\mathsf{StdDecomp}_{r^*}$ is the local decompression procedure defined in Section 2.4 and $\mathrm{U}^{r^*}_{g^*}$ is an involution defined as:

$$\mathrm{U}^{r^*}_{g^*} : |D_\mathsf{G}\rangle \mapsto \begin{cases} |D_\mathsf{G} \cup (r^*, g^*)\rangle & if\ D_\mathsf{G}(r^*) = \perp, \\ |D_\mathsf{G} \backslash (r^*, g^*)\rangle & if\ D_\mathsf{G}(r^*) = g^*, \\ |D_\mathsf{G}\rangle & otherwise. \end{cases} \tag{15}$$

In game $\mathbf{G_6}$, s^* is generated as $(m_b^* || 0^{k_1}) \oplus \mathsf{G}(r^*)$, so a classical query to G by r^* is needed. Since G is simulated by $\mathsf{CStO_G}$, this classical query after (m_0^*, m_1^*) has been computed in game $\mathbf{G_6}$ is actually processed as follows:

1. Initialize the query registers $\mathsf{X_G Y_G}$ and prepare the initial state as $|r^*, 0\rangle$.

[14] Here function E is introduced only to ensure that the conditions for applying Theorem 1 are met. In the corresponding games $\mathbf{G_5}$ and $\mathbf{G_6}$, the adversary $\mathcal{A}$ has access to the oracles F and $\mathsf{F}\backslash\{r^*\}$, respectively.

2. Perform $\mathsf{CStO}_{\mathsf{G}}$ on the registers $\mathsf{X}_{\mathsf{G}}\mathsf{Y}_{\mathsf{G}}\mathsf{D}_{q_{\mathsf{G}}}$, measure $\mathsf{X}_{\mathsf{G}}\mathsf{Y}_{\mathsf{G}}$ to get the outcome (r^*, y) and then rephrase y as $\mathsf{G}(r^*)$.

We note that if Find_{r^*} does not occur in game $\mathbf{G_6}$, this y must be uniformly random in $\{0,1\}^{n+k_1}$. In fact, by the definition of the semi-classical oracle (Definition 1), if Find_{r^*} does not occur, r^* is never queried by the adversary, and hence the whole (unit) state of the game just before computing s^* can be written as $\sum_{z,D_{\mathsf{G}}(r^*)=\perp}\alpha_{z,D_{\mathsf{G}}}|z,D_{\mathsf{G}}\rangle$, where $|z\rangle$ denotes the other state that may entangle with $|D_{\mathsf{G}}\rangle$. Then, the classical query to G by r^* has the following computation:

$$\sum_{z,D_{\mathsf{G}}(r^*)=\perp}\alpha_{z,D_{\mathsf{G}}}|z,D_{\mathsf{G}}\rangle \xrightarrow{\text{Initialize } \mathsf{X}_{\mathsf{G}}\mathsf{Y}_{\mathsf{G}}} |r^*,0\rangle \sum_{z,D_{\mathsf{G}}(r^*)=\perp}\alpha_{z,D_{\mathsf{G}}}|z,D_{\mathsf{G}}\rangle \xrightarrow{\mathsf{StdDecomp}_{r^*}}$$

$$\sum_{z,D_{\mathsf{G}}(r^*)=\perp}\frac{\alpha_{z,D_{\mathsf{G}}}}{\sqrt{2^{n+k_1}}}|r^*,0\rangle \sum_{y_0\in\{0,1\}^{n+k_1}}|z,D_{\mathsf{G}}\cup(r^*,y_0)\rangle \xrightarrow{\mathsf{CNOT}^{r^*}}$$

$$\sum_{y_0\in\{0,1\}^{n+k_1}}|r^*,y_0\rangle \sum_{z,D_{\mathsf{G}}(r^*)=\perp}\frac{\alpha_{z,D_{\mathsf{G}}}}{\sqrt{2^{n+k_1}}}|z,D_{\mathsf{G}}\cup(r^*,y_0)\rangle \xrightarrow{\mathsf{StdDecomp}_{r^*}}$$

$$|\psi\rangle := \sum_{y_0\in\{0,1\}^{n+k_1}}|r^*,y_0\rangle \sum_{z,D_{\mathsf{G}}(r^*)=\perp}\frac{\alpha_{z,D_{\mathsf{G}}}}{\sqrt{2^{n+k_1}}}\mathsf{StdDecomp}_{r^*}|z,D_{\mathsf{G}}\cup(r^*,y_0)\rangle.$$

At this point, based on the above state $|\psi\rangle$, it is obvious to see that the measurement on registers $\mathsf{X}_{\mathsf{G}}\mathsf{Y}_{\mathsf{G}}$ will get (r^*, y) for any $y \in \{0,1\}^{n+k_1}$ with probability $1/2^{n+k_1}$, since $\sum_{z,D_{\mathsf{G}}(r^*)=\perp}\alpha_{z,D_{\mathsf{G}}}|z,D_{\mathsf{G}}\rangle$ is unit and $\mathsf{StdDecomp}_{r^*}$ is a unitary operation. So we indeed get a uniformly random y after we make a classical query to G by r^*. Meanwhile, the state $|\psi\rangle$ will collapse into $\sum_{z,D_{\mathsf{G}}(r^*)=\perp}\alpha_{z,D_{\mathsf{G}}}\mathsf{StdDecomp}_{r^*}|z,D_{\mathsf{G}}\cup(r^*,y)\rangle$[15] after we get the measurement outcome y and rephrase it as $\mathsf{G}(r^*)$.

Instead of querying G to get $\mathsf{G}(r^*)$, game $\mathbf{G_7}$ directly chooses a uniformly random g^* from $\{0,1\}^{n+k_1}$ as $\mathsf{G}(r^*)$ and then sequentially performs $\mathrm{U}^{r^*}_{g^*}$ (Eq. (15)) and $\mathsf{StdDecomp}_{r^*}$ to update the database state. In fact, if Find_{r^*} does not occur in game $\mathbf{G_7}$, the whole (unit) state of the game just before computing s^* can also be written as $\sum_{z,D_{\mathsf{G}}(r^*)=\perp}\alpha_{z,D_{\mathsf{G}}}|z,D_{\mathsf{G}}\rangle$, and one can easily check that, after sequentially performing $\mathrm{U}^{r^*}_{g^*}$ and $\mathsf{StdDecomp}_{r^*}$, that whole state will become $\sum_{z,D_{\mathsf{G}}(r^*)=\perp}\alpha_{z,D_{\mathsf{G}}}\mathsf{StdDecomp}_{r^*}|z,D_{\mathsf{G}}\cup(r^*,g^*)\rangle$, where g^* is uniformly random.

Now we can conclude that, if Find_{r^*} does not occur in games $\mathbf{G_6}$ and $\mathbf{G_7}$, their corresponding whole states just after the simulation of $\mathsf{G}(r^*)$ are identical[16] even though they apply different computing methods. This means that these two games must have the same final output statistics. Thus

[15] Since the query registers $\mathsf{X}_{\mathsf{G}}\mathsf{Y}_{\mathsf{G}}$ are measured, we omit $|r^*, y\rangle$ in this collapsed state.

[16] More rigorously speaking, these two whole states are both mixed states and have the same density matrix.

$$\begin{aligned}\Pr[b=1\wedge\neg\mathsf{Find}_{r^*}:b\leftarrow\mathbf{G_6}] &= \Pr[b=1|\neg\mathsf{Find}_{r^*}:b\leftarrow\mathbf{G_6}]\cdot\Pr[\neg\mathsf{Find}_{r^*}:\mathbf{G_6}]\\ &= \Pr[b=1|\neg\mathsf{Find}_{r^*}:b\leftarrow\mathbf{G_7}]\cdot\Pr[\neg\mathsf{Find}_{r^*}:\mathbf{G_6}]\\ &\overset{(a)}{=} \Pr[b=1|\neg\mathsf{Find}_{r^*}:b\leftarrow\mathbf{G_7}]\cdot\Pr[\neg\mathsf{Find}_{r^*}:\mathbf{G_7}]\\ &= \Pr[b=1\wedge\neg\mathsf{Find}_{r^*}:b\leftarrow\mathbf{G_7}].\end{aligned}$$

Here (a) uses the fact that $\Pr[\neg\mathsf{Find}_{r^*}:\mathbf{G_6}]=\Pr[\neg\mathsf{Find}_{r^*}:\mathbf{G_7}]$, which is obvious since the changes made in game $\mathbf{G_7}$ do not affect the probability that Find_{r^*} occurs. Then, since the construction of games $\mathbf{G_6}$ and $\mathbf{G_7}$ ensures that games $\mathbf{G_6}$ and $\mathbf{G_7}$ returning 1 implies that Find_{r^*} does not occur, the above equation means that

$$\Pr[1\leftarrow\mathbf{G_6}]=\Pr[1\leftarrow\mathbf{G_7}]. \tag{16}$$

Game $\mathbf{G_8}$: Different from game $\mathbf{G_7}$, which first uniformly samples g^* and then computes $s^*=(m_b^*||0^{k_1})\oplus g^*$, this game first uniformly samples s^* at the beginning and then computes $g^*=(m_b^*||0^{k_1})\oplus s^*$. Obviously this is only a conceptual change, thus

$$\Pr[1\leftarrow\mathbf{G_7}]=\Pr[1\leftarrow\mathbf{G_8}]. \tag{17}$$

Game $\mathbf{G_9}$: Compared with game $\mathbf{G_8}$, this new game has the following two changes:

- Before $\mathcal{A}$ outputs (m_0^*,m_1^*), H is simulated by $\mathsf{CStO}_{\mathsf{H}}\circ\mathcal{O}^{SC}_{\{s^*\}}$, which is an operation that first queries the semi-classical oracle $\mathcal{O}^{SC}_{\{s^*\}}$ (Definition 1) on the input register of H and then performs $\mathsf{CStO}_{\mathsf{H}}$.
- If Find_{s^*} occurs before $\mathcal{A}$ outputs (m_0^*,m_1^*), the game finally returns $\perp$. Here Find_{s^*} is the event that the measurement of $\mathcal{O}^{SC}_{\{s^*\}}$ ever has result 1.

Following a similar analysis in the game hop from $\mathbf{G_5}$ to $\mathbf{G_6}$, and shift the perspective from oracle G to oracle H, we can obtain the following formula.

$$|\Pr[1\leftarrow\mathbf{G_8}]-\Pr[1\leftarrow\mathbf{G_9}]|\leq\sqrt{(q_{\mathsf{H}}+1)\cdot\Pr[\mathsf{Find}_{s^*}:\mathbf{G_9}]}. \tag{18}$$

As for the $\Pr[\mathsf{Find}_{r^*}:\mathbf{G_6}]$ used in Eq. (14) and the above $\Pr[\mathsf{Find}_{s^*}:\mathbf{G_9}]$, we can prove the following lemma using Corollary 1, and the detailed proof is given in Supplementary Material D.5.

Lemma 3. $\Pr[\mathsf{Find}_{r^*}:\mathbf{G_6}]\leq 4q_{\mathsf{G}}\cdot 2^{-k_0}$ *and* $\Pr[\mathsf{Find}_{s^*}:\mathbf{G_9}]\leq 4q_{\mathsf{H}}\cdot 2^{-n-k_1}$.

Game $\mathbf{G_{10}}$: This game is the same as game $\mathbf{G_9}$ except that the computation of t^* is changed. In this game, t^* is generated as $r^*\oplus h^*$, where h^* is a new bitstring uniformly sampled from $\{0,1\}^{k_0}$ at the beginning of the game. And after the generation of t^*, the $\mathrm{U}^{s^*}_{h^*}$ and $\mathsf{StdDecomp}_{s^*}$ are sequentially performed on the database register $\mathsf{D}_{q_{\mathsf{H}}}$, where $\mathsf{StdDecomp}_{s^*}$ is the local decompression procedure defined in Section 2.4 and $\mathrm{U}^{s^*}_{h^*}$ is an involution defined as:

$$\mathrm{U}^{s^*}_{h^*}:|D_{\mathsf{H}}\rangle\mapsto\begin{cases}|D_{\mathsf{H}}\cup(s^*,h^*)\rangle & if\ D_{\mathsf{H}}(s^*)=\perp,\\ |D_{\mathsf{H}}\backslash(s^*,h^*)\rangle & if\ D_{\mathsf{H}}(s^*)=h^*,\\ |D_{\mathsf{H}}\rangle & otherwise.\end{cases} \tag{19}$$

Following a similar analysis in the game hop from $\mathbf{G_6}$ to $\mathbf{G_7}$, and shift the perspective from oracle G to oracle H, we can obtain

$$\Pr[1 \leftarrow \mathbf{G_9}] = \Pr[1 \leftarrow \mathbf{G_{10}}]. \tag{20}$$

Game $\mathbf{G_{11}}$: Different from game $\mathbf{G_{10}}$, which first uniformly samples h^* and then computes $t^* = r^* \oplus h^*$, this game first uniformly samples t^* at the beginning and then computes $h^* = r^* \oplus t^*$. Obviously this is only a conceptual change, thus

$$\Pr[1 \leftarrow \mathbf{G_{10}}] = \Pr[1 \leftarrow \mathbf{G_{11}}]. \tag{21}$$

Game $\mathbf{G_{12}}$: This game is the same as game $\mathbf{G_{11}}$, except that the simulation of oracles G and H during the generation of b' is changed. In fact, the $\mathsf{CStO}_{\mathsf{G}}$ used to simulate G is changed into

$$\begin{aligned}\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*} := &\sum_{x\in\{0,1\}^{k_0}\setminus\{r^*\}} |x\rangle\langle x| \otimes \mathsf{StdDecomp}_x \circ \mathsf{CNOT}^x \circ \mathsf{StdDecomp}_x \\ &+ |r^*\rangle\langle r^*| \otimes \mathsf{CNOT}_{g^*},\end{aligned} \tag{22}$$

and the $\mathsf{CStO}_{\mathsf{H}}$ used to simulate H is changed to

$$\begin{aligned}\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*} := &\sum_{x\in\{0,1\}^{n+k_1}\setminus\{s^*\}} |x\rangle\langle x| \otimes \mathsf{StdDecomp}_x \circ \mathsf{CNOT}^x \circ \mathsf{StdDecomp}_x \\ &+ |s^*\rangle\langle s^*| \otimes \mathsf{CNOT}_{h^*}.\end{aligned} \tag{23}$$

Here CNOT_{g^*} (resp. CNOT_{h^*}) acts on the output register of G (resp. H) and maps $|y\rangle$ into $|y \oplus g^*\rangle$ (resp. $|y \oplus h^*\rangle$).

Similar to the analysis in the game hop from $\mathbf{G_6}$ to $\mathbf{G_7}$, if Find_{r^*} and Find_{s^*} both do not occur in game $\mathbf{G_{11}}$, we know that the whole state of this game just after the computation of c^* can be written as

$$\sum_{\substack{z, D_{\mathsf{G}}(r^*)=\perp \\ D_{\mathsf{H}}(s^*)=\perp}} \alpha^z_{D_{\mathsf{G}},D_{\mathsf{H}}} |z\rangle \otimes \mathsf{StdDecomp}_{r^*}|D_{\mathsf{G}} \cup (r^*, g^*)\rangle \otimes \mathsf{StdDecomp}_{s^*}|D_{\mathsf{H}} \cup (s^*, h^*)\rangle.$$

Here $|z\rangle$ denotes the other state that may entangle with $|D_{\mathsf{G}}\rangle$ and $|D_{\mathsf{H}}\rangle$. As for the next generation of b', we note that the $\mathrm{U}^*_{\mathsf{Search}}$ used to simulate the decryption oracle qDec may disturb the database state and make it not in the form $\mathsf{StdDecomp}_{r^*}|D_{\mathsf{G}} \cup (r^*, g^*)\rangle \otimes \mathsf{StdDecomp}_{s^*}|D_{\mathsf{H}} \cup (s^*, h^*)\rangle$, and the reason is that $\mathrm{U}^*_{\mathsf{Search}}$ may not commute with $\mathsf{StdDecomp}_{r^*}$ or $\mathsf{StdDecomp}_{s^*}$[17]. Correspondingly, if we additionally assume that $\mathrm{U}^*_{\mathsf{Search}}$ commutes with $\mathsf{StdDecomp}_{r^*}$ and

[17] In fact, $\mathrm{U}^*_{\mathsf{Search}}$ defined in game $\mathbf{G_5}$ makes a database read in the computational basis, while the $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ defined in Section 2.4 perform some operations under the Fourier basis, making it challenging to directly claim that these operations are commutative.

$\mathsf{StdDecomp}_{s^*}$, the whole state of game $\mathbf{G_{11}}$ just before the first random oracle query (to G or H) during the generation of b' can be written as

$$\sum_{\substack{z, D_{\mathsf{G}}(r^*)=\perp \\ D_{\mathsf{H}}(s^*)=\perp}} \beta^z_{D_{\mathsf{G}},D_{\mathsf{H}}}|z\rangle \otimes \mathsf{StdDecomp}_{r^*}|D_{\mathsf{G}} \cup (r^*, g^*)\rangle \otimes \mathsf{StdDecomp}_{s^*}|D_{\mathsf{H}} \cup (s^*, h^*)\rangle,$$

since $\mathrm{U}^*_{\mathsf{Search}}$ does not change the database state $|D_{\mathsf{G}}\rangle$ and $|D_{\mathsf{H}}\rangle$ in the computational basis. In the following proof we denote the above state as $|\Phi\rangle$. Based on this observation, we can prove the following claim.

Claim. In game $\mathbf{G_{11}}$, if neither Find_{r^*} nor Find_{s^*} occurs and $\mathrm{U}^*_{\mathsf{Search}}$ commutes with both $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$, we can conclude that $\mathsf{G}(r^*) = g^*$ and $\mathsf{H}(s^*) = h^*$ in the adversary's view during the generation of b'.

Proof. As analyzed above, under the condition of this claim, the whole state of game $\mathbf{G_{11}}$ just before the first random oracle query during the generation of b' is a particular state $|\Phi\rangle$. Without loss of generality, we assume that the first random oracle query during the generation of b' is a G query.

Now for this first G query, the corresponding simulation process of $\mathsf{G}(r^*)$ can be described as follows, which we denote $|r^*, y\rangle$ as the query state.

$$\begin{aligned}
&|r^*, y\rangle|\Phi\rangle \xrightarrow{\mathsf{StdDecomp}_{r^*}} \\
&|r^*, y\rangle \sum \beta^z_{D_{\mathsf{G}},D_{\mathsf{H}}}|z\rangle \otimes |D_{\mathsf{G}} \cup (r^*, g^*)\rangle \otimes \mathsf{StdDecomp}_{s^*}|D_{\mathsf{H}} \cup (s^*, h^*)\rangle \xrightarrow{\mathsf{CNOT}^{r^*}} \\
&|r^*, y \oplus g^*\rangle \sum \beta^z_{D_{\mathsf{G}},D_{\mathsf{H}}}|z\rangle \otimes |D_{\mathsf{G}} \cup (r^*, g^*)\rangle \otimes \mathsf{StdDecomp}_{s^*}|D_{\mathsf{H}} \cup (s^*, h^*)\rangle \\
&\xrightarrow{\mathsf{StdDecomp}_{r^*}} |r^*, y \oplus g^*\rangle|\Phi\rangle.
\end{aligned}$$

Therefore, according to this claim, the response to the adversary's first G query will consistently satisfy $\mathsf{G}(r^*) = g^*$, and the database state corresponding to the query state $|r^*, y\rangle$ remains unchanged after the simulation of this G query.

This actually means that, the reply to the subsequent queries to G will also maintain $\mathsf{G}(r^*) = g^*$. Hence according to this claim, $\mathsf{G}(r^*)$ is exactly g^* in the adversary's view before it outputs b'. A similar analysis applies to $\mathsf{H}(s^*)$, and it is exactly h^* in the adversary's view. Thus we complete the proof.

Based on the above claim, we make a modification in game $\mathbf{G_{12}}$, which is, the simulation of G (resp. H) is switched from $\mathsf{CStO}_{\mathsf{G}}$ (resp. $\mathsf{CStO}_{\mathsf{H}}$) to $\mathsf{CStO}^{r^*,g^*}_{\mathsf{G}}$ (resp. $\mathsf{CStO}^{s^*,h^*}_{\mathsf{H}}$) defined in Eq. (22) (resp. Eq. (23)). Now for the query $|r^*, y\rangle$ to G (resp. $|s^*, y\rangle$ to H) during the generation of b', the simulation of game $\mathbf{G_{12}}$ will not perform any operation on the database state and directly XOR g^* (resp. h^*) onto y, where g^* (resp. h^*) is actually rephrased as $\mathsf{G}(r^*)$ (resp. $\mathsf{H}(s^*)$). Notice that the construction of games $\mathbf{G_{11}}$ and $\mathbf{G_{12}}$ ensures that the two games returning 1 implies that Find_{r^*} and Find_{s^*} do not occur, hence we have $\Pr[1 \leftarrow \mathbf{G_{11}}] = \Pr[1 \leftarrow \mathbf{G_{12}}]$ under the condition that $\mathrm{U}^*_{\mathsf{Search}}$ commutes with $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$.

Lemma 4. *Let* $[\mathrm{A},\mathrm{B}]$ *be* $\mathrm{AB}-\mathrm{BA}$ *for any operations* A *and* B*, then we have* $\|[\mathrm{U}^*_{\mathsf{Search}}, \mathsf{StdDecomp}_{r^*}]\| \leq 8\cdot\sqrt{2^{-k_1}}$, $\|[\mathrm{U}^*_{\mathsf{Search}}, \mathsf{StdDecomp}_{s^*}]\| \leq 8\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}$.

In the above we introduce Lemma 4, which is proved in Supplementary Material D.6. It shows that $\mathrm{U}^*_{\mathsf{Search}}$ indeed commutes with $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ except for a fixed error. Since $\mathrm{U}^*_{\mathsf{Search}}$ is implemented at most q_{D} times during the generation of b', we finally obtain

$$|\Pr[1\leftarrow \mathbf{G_{11}}]-\Pr[1\leftarrow \mathbf{G_{12}}]| \leq 8q_{\mathsf{D}}\cdot\sqrt{2^{-k_1}}+8q_{\mathsf{D}}\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}. \tag{24}$$

Game $\mathbf{G_{13}}$: This game is the same as game $\mathbf{G_{12}}$, except that the performing of $\mathsf{StdDecomp}_{r^*}$ (resp. $\mathsf{StdDecomp}_{s^*}$) just after the performing of $\mathrm{U}^{r^*}_{g^*}$ (resp. $\mathrm{U}^{s^*}_{h^*}$) is canceled.

To show the difference between game $\mathbf{G_{12}}$ and game $\mathbf{G_{13}}$, we present the following Lemma 5. At a high level, the core proof idea of this lemma is to exploit the commutativity of $\mathsf{StdDecomp}_{r^*}$, $\mathsf{StdDecomp}_{s^*}$, $\mathrm{U}^{r^*}_{g^*}$, $\mathrm{U}^{s^*}_{h^*}$ and $\mathrm{U}^*_{\mathsf{Search}}$. For example, $\mathsf{StdDecomp}_{r^*}$ must commute with $\mathsf{StdDecomp}_{s^*}$ and $\mathrm{U}^{s^*}_{h^*}$ since they act on different database registers. Moreover, by Lemma 4, both $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ almost commute with $\mathrm{U}^*_{\mathsf{Search}}$. Due to space limitations, we give the detailed proof of Lemma 5 in Supplementary Material D.7.

Lemma 5. $|\Pr[1\leftarrow \mathbf{G_{12}}]-\Pr[1\leftarrow \mathbf{G_{13}}]| \leq 8q_{\mathsf{D}}\cdot\sqrt{2^{-k_1}}+8q_{\mathsf{D}}\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}$.

Game $\mathbf{G_{14}}$: Before $\mathcal{A}$ outputs (m_0^*, m_1^*), the queries to the semi-classical oracles $\mathcal{O}^{SC}_{\{r^*\}}$ and $\mathcal{O}^{SC}_{\{s^*\}}$ performed in game $\mathbf{G_{13}}$ are canceled, meanwhile the two flags "Find_{r^*} occurs, game finally returns $\perp$" and "Find_{s^*} occurs, game finally returns $\perp$" are both removed.

By mimicking the analysis given in the game hop from $\mathbf{G_5}$ to $\mathbf{G_6}$ and $\mathbf{G_8}$ to $\mathbf{G_9}$, and then using Lemma 3, we obtain

$$|\Pr[1\leftarrow \mathbf{G_{13}}]-\Pr[1\leftarrow \mathbf{G_{14}}]| \leq 2(q_{\mathsf{G}}+1)\cdot\sqrt{2^{-k_0}}+2(q_{\mathsf{H}}+1)\cdot\sqrt{2^{-n-k_1}}. \tag{25}$$

Now by combining Eq. (14), (16), (17), (18), (20), (21), (24), (25) and Lemma 3, 5, we obtain

$$\begin{aligned}|\Pr[1\leftarrow \mathbf{G_5}]-\Pr[1\leftarrow \mathbf{G_{14}}]| \leq & 4(q_{\mathsf{H}}+1)\cdot\sqrt{2^{-n-k_1}}+16q_{\mathsf{D}}\cdot\sqrt{2^{-k_1}} \\ & +(16q_{\mathsf{D}}\sqrt{q_{\mathsf{G}}}+4q_{\mathsf{G}}+4)\cdot\sqrt{2^{-k_0}}.\end{aligned} \tag{26}$$

4.3 Reprogram G and H Queried During the Generation of b'

In this section, we introduce games $\mathbf{G_{15}}$ to $\mathbf{G_{17}}$ as shown in Fig. 9 to reprogram the oracles G and H queried during the generation of b'. After this reprogramming, we get a trivial game $\mathbf{G_{17}}$ which returns 1 with the probability $1/2$.

GAMES $\mathbf{G_{14}}$ to $\mathbf{G_{17}}$

1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
   $r^* \leftarrow_{\$} \{0,1\}^{k_0}$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
   $s^* \leftarrow_{\$} \{0,1\}^{n+k_1}$, $t^* \leftarrow_{\$} \{0,1\}^{k_0}$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}}(\mathsf{pk})$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
3. $b \xleftarrow{\$} \{0,1\}$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
   $g^* = (m_b^* || 0^{k_1}) \oplus s^*, h^* = r^* \oplus t^*$ //$\mathbf{G_{14}}$-$\mathbf{G_{15}}$
   perform $\mathrm{U}_{h^*}^{s^*} \circ \mathrm{U}_{g^*}^{r^*}$ //$\mathbf{G_{14}}$-$\mathbf{G_{15}}$
   $c^* = f_{\mathsf{pk}}(s^*, t^*)$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
4. $b' \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}}(\mathsf{pk}, c^*)$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
5. **return** $[\![b' = b]\!]$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$

$\mathsf{G}(|x_{\mathsf{G}}, y_{\mathsf{G}}\rangle)/\mathsf{H}(|x_{\mathsf{H}}, y_{\mathsf{H}}\rangle)$

if (m_0^*, m_1^*) has not been defined
  apply $\mathsf{CStO}_{\mathsf{G}}/\mathsf{CStO}_{\mathsf{H}}$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
else
  apply $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}/\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*}$ //$\mathbf{G_{14}}$-$\mathbf{G_{16}}$
  apply $\mathsf{CStO}_{\mathsf{G}}/\mathsf{CStO}_{\mathsf{H}}$ //$\mathbf{G_{17}}$

$\mathsf{qDec}(|c, y\rangle)$

if c^* has not been defined
  apply $\mathrm{U}_{\mathsf{Search}}$ //$\mathbf{G_{14}}$-$\mathbf{G_{17}}$
else apply $\mathrm{U}_{\mathsf{Search}}^{*}$ //$\mathbf{G_{14}}$, $\mathbf{G_{16}}$-$\mathbf{G_{17}}$
else apply $\mathrm{U}_{\mathsf{Search}}^{**}$ //$\mathbf{G_{15}}$

Fig. 9. Games $\mathbf{G}_{14}$ to $\mathbf{G}_{17}$ in the proof of Theorem 4.

Game $\mathbf{G_{15}}$: This game is the same as game $\mathbf{G_{14}}$, except that the $\mathrm{U}_{\mathsf{Search}}^{*}$ used to simulate the decryption oracle qDec during the generation of b' is replaced by

$$\mathrm{U}_{\mathsf{Search}}^{**} := \mathrm{U}_{h^*}^{s^*} \circ \mathrm{U}_{g^*}^{r^*} \circ \mathrm{U}_{\mathsf{Search}}^{*} \circ \mathrm{U}_{h^*}^{s^*} \circ \mathrm{U}_{g^*}^{r^*},$$

where $\mathrm{U}_{g^*}^{r^*}$ (Eq. (15)) acts on $\mathsf{D}_{q_{\mathsf{G}}}$ and $\mathrm{U}_{h^*}^{s^*}$ (Eq. (19)) acts on $\mathsf{D}_{q_{\mathsf{H}}}$.

Here we introduce the following Lemma 6, which shows that the difference between game $\mathbf{G_{14}}$ and game $\mathbf{G_{15}}$ can be upper bounded by the advantage of a PD-OW adversary $\mathcal{B}_1$ against TDI with some additional loss. The proof of this lemma mainly uses Theorem 1 and Theorem 6, and the detailed proof is given in Supplementary Material D.8.

Lemma 6. *There exists a* PD-OW *adversary* $\mathcal{B}_1$ *against* TDI *such that*

$$\begin{aligned}|\Pr[1 \leftarrow \mathbf{G_{14}}] - \Pr[1 \leftarrow \mathbf{G_{15}}]| \leq & 18 q_{\mathsf{D}} q \cdot \sqrt{2^{-k_0}} + 8 q_{\mathsf{D}}(q_{\mathsf{H}}+1) \cdot \sqrt{2^{-n-k_1}} \\ & + 8 q_{\mathsf{D}} q \cdot \sqrt{\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}_1}},\end{aligned}$$

and $\mathcal{T}_{\mathcal{B}_1} \leq \mathcal{T}_{\mathcal{A}} + \mathcal{O}(q_{\mathsf{G}}^2 + q_{\mathsf{H}}^2) + \mathcal{O}(q_{\mathsf{D}} q_{\mathsf{G}} q_{\mathsf{H}}) \cdot \mathcal{T}_f$. *Here* $q := q_{\mathsf{G}} + q_{\mathsf{H}} + 1$.

Game $\mathbf{G_{16}}$: Compared with game $\mathbf{G_{15}}$, this new game makes two changes: the computing of g^* and h^*, and the performing of $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ just before the generation of c^* are canceled, and the $\mathrm{U}_{\mathsf{Search}}^{**}$ introduced in game $\mathbf{G_{15}}$ is changed back to $\mathrm{U}_{\mathsf{Search}}^{*}$.

To show the difference between game $\mathbf{G_{15}}$ and game $\mathbf{G_{16}}$, we present the following Lemma 7. At a high level, the core idea of its proof is to use the facts that $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ are both involutions, and that $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ both commute with $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$ and $\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*}$. Due to space limitations, we give the detailed proof of Lemma 7 in Supplementary Material D.9.

Lemma 7. $\Pr[1 \leftarrow \mathbf{G_{15}}] = \Pr[1 \leftarrow \mathbf{G_{16}}]$.

Game $\mathbf{G_{17}}$: This game is the same as game $\mathbf{G_{16}}$, except that the simulation of G and H during the generation of b' is changed. In fact, the $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$ used to simulate G is changed back to $\mathsf{CStO}_{\mathsf{G}}$, and the $\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*}$ used to simulate H is changed back to $\mathsf{CStO}_{\mathsf{H}}$.

In fact we can prove the following lemma, which shows that the difference between game $\mathbf{G_{16}}$ and game $\mathbf{G_{17}}$ can be upper bounded by the advantage of a PD-OW adversary $\mathcal{B}_2$ against TDI with some additional loss. The detailed proof of this lemma is given in Supplementary Material D.10.

Lemma 8. *There exists a* PD-OW *adversary* $\mathcal{B}_2$ *against* TDI *such that*

$$|\Pr[1 \leftarrow \mathbf{G_{16}}] - \Pr[1 \leftarrow \mathbf{G_{17}}]| \leq 4q \cdot \sqrt{2^{-k_0}} + 4q \cdot \sqrt{\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}_2}},$$

and $\mathcal{T}_{\mathcal{B}_2} \leq \mathcal{T}_{\mathcal{A}} + \mathcal{O}(q_{\mathsf{G}}^2 + q_{\mathsf{H}}^2) + \mathcal{O}(q_{\mathsf{D}} q_{\mathsf{G}} q_{\mathsf{H}}) \cdot \mathcal{T}_f$. *Here* $q := q_{\mathsf{G}} + q_{\mathsf{H}} + 1$.

It is easy to check that $\Pr[1 \leftarrow \mathbf{G_{17}}] = 1/2$, then by combining this equation with Lemma 6, 7 and 8, we obtain (denote $q := q_{\mathsf{G}} + q_{\mathsf{H}} + 1$)

$$\begin{aligned}\left|\Pr[1 \leftarrow \mathbf{G_{14}}] - \frac{1}{2}\right| \leq & (18q_{\mathsf{D}} + 4)q \cdot \sqrt{2^{-k_0}} + 8q_{\mathsf{D}}(q_{\mathsf{H}} + 1) \cdot \sqrt{2^{-n-k_1}} \\ & + 8q_{\mathsf{D}} q \cdot \left(\sqrt{\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}_1}} + \sqrt{\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}_2}}\right).\end{aligned} \tag{27}$$

4.4 Putting Everything Together

For the PD-OW adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ involved in Eq. (27), it is evident that we can fold them into one PD-OW adversary $\mathcal{B}$ such that $\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}_1} + \mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}_2} = 2 \cdot \mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}}$ and $\mathcal{T}_{\mathcal{B}} \leq \max\{\mathcal{T}_{\mathcal{B}_1}, \mathcal{T}_{\mathcal{B}_2}\}$. Here we also have $2^{-n-k_1} \leq \mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}}$ since the aim of $\mathcal{B}$ is to guess a bitstring with length $n + k_1$. Now by combining Eq. (4), Eq. (13), Eq. (26) and Eq. (27), and using the above two properties, we finally prove our Theorem 4, and we restate it below.

Theorem 5 (PD-OW of TDI $\overset{\text{QROM}}{\Rightarrow}$ IND-qCCA of OAEP[TDI, G, H]). *Let* $\mathcal{A}$ *be an* IND-qCCA *adversary against* OAEP[TDI, G, H] *in the QROM, making at most* q_{D}, q_{G} *and* q_{H} *quantum queries to the decryption oracle, random oracle* G *and random oracle* H. *Let* $q := q_{\mathsf{G}} + q_{\mathsf{H}} + 1$. *Then we can construct a* PD-OW *adversary* $\mathcal{B}$ *against the trapdoor injective function* TDI *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{OAEP[TDI,G,H]},\mathcal{A}} \leq & (32q_{\mathsf{D}} + 4)q \cdot \sqrt{\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}}} + (42q_{\mathsf{D}} + 8)q \cdot \sqrt{2^{-k_0}} \\ & + 24q_{\mathsf{D}} \cdot \sqrt{2^{-k_1}},\end{aligned}$$

and $\mathcal{T}_{\mathcal{B}} \leq \mathcal{T}_{\mathcal{A}} + \mathcal{O}(q_{\mathsf{G}}^2 + q_{\mathsf{H}}^2) + \mathcal{O}(q_{\mathsf{D}} q_{\mathsf{G}} q_{\mathsf{H}}) \cdot \mathcal{T}_f$.

Acknowledgments. We thank anonymous reviewers for their insightful comments. This work is supported by the National Natural Science Foundation of China (Grants No. 62172405, 62302496).

References

1. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. In: Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Proceedings, Part II. pp. 269–295. Springer (2019). https://doi.org/10.1007/978-3-030-26951-7_10
2. Backes, M., Dürmuth, M., Unruh, D.: OAEP is secure under key-dependent messages. In: Pieprzyk, J. (ed.) Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings. pp. 506–523. Springer (2008). https://doi.org/10.1007/978-3-540-89255-7_31
3. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: Santis, A.D. (ed.) Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Proceedings. Lecture Notes in Computer Science, vol. 950, pp. 92–111. Springer (1994). https://doi.org/10.1007/BFB0053428
4. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of CCA security in the quantum random oracle model. In: Theory of Cryptography Conference. pp. 61–90. Springer (2019). https://doi.org/10.1007/978-3-030-36033-7_3
5. Boneh, D.: Simplified OAEP for the RSA and rabin functions. In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Proceedings. pp. 275–291. Springer (2001). https://doi.org/10.1007/3-540-44647-8_17
6. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. pp. 41–69. Springer (2011). https://doi.org/10.1007/978-3-642-25385-0_3
7. Boyko, V.: On the security properties of OAEP as an all-or-nothing transform. In: Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Proceedings. pp. 503–518. Springer (1999). https://doi.org/10.1007/3-540-48405-1_32
8. Chiesa, A., Manohar, P., Spooner, N.: Succinct arguments in the quantum random oracle model. In: Theory of Cryptography - 17th International Conference, TCC 2019, Proceedings, Part II. pp. 1–29. Springer (2019). https://doi.org/10.1007/978-3-030-36033-7_1
9. Chung, K., Fehr, S., Huang, Y., Liao, T.: On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In: Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part II. pp. 598–629. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_21
10. Czajkowski, J., Majenz, C., Schaffner, C., Zur, S.: Quantum lazy sampling and game-playing proofs for quantum indifferentiability. IACR Cryptol. ePrint Arch. p. 428 (2019), https://eprint.iacr.org/2019/428
11. Dodis, Y., Fiore, D.: Interactive encryption and message authentication. In: Security and Cryptography for Networks - 9th International Conference, SCN 2014, Proceedings. vol. 8642, pp. 494–513. Springer (2014), https://doi.org/10.1007/978-3-319-10879-7_28

12. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Online-extractability in the quantum random-oracle model. In: Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part III. pp. 677–706. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_24
13. Ebrahimi, E.: Post-quantum security of plain OAEP transform. In: Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13177, pp. 34–51. Springer (2022). https://doi.org/10.1007/978-3-030-97121-2_2
14. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer (1999). https://doi.org/10.1007/3-540-48405-1_34
15. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptol. **26**(1), 80–101 (2013). https://doi.org/10.1007/S00145-011-9114-1
16. Ge, J., Shan, T., Xue, R.: Tighter qcca-secure key encapsulation mechanism with explicit rejection in the quantum random oracle model. In: Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Proceedings, Part V. pp. 292–324. Springer (2023). https://doi.org/10.1007/978-3-031-38554-4_10
17. Grilo, A.B., Hövelmanns, K., Hülsing, A., Majenz, C.: Tight adaptive reprogramming in the QROM. In: Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Part I. pp. 637–667. Springer (2021). https://doi.org/10.1007/978-3-030-92062-3_22
18. Hashimoto, K., Katsumata, S., Wiggers, T.: Bundled authenticated key exchange: A concrete treatment of (post-quantum) signal's handshake protocol. IACR Cryptol. ePrint Arch. p. 040 (2025), https://eprint.iacr.org/2025/040
19. Hosoyamada, A., Iwata, T.: 4-round luby-rackoff construction is a qprp. In: Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Part I. pp. 145–174. Springer (2019). https://doi.org/10.1007/978-3-030-34578-5_6
20. Hövelmanns, K., Hülsing, A., Majenz, C.: Failing gracefully: Decryption failures and the fujisaki-okamoto transform. In: Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13794, pp. 414–443. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_15
21. Huguenin-Dumittan, L., Vaudenay, S.: On ind-qcca security in the ROM and its applications - CPA security is sufficient for TLS 1.3. In: Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part III. vol. 13277, pp. 613–642. Springer (2022), https://doi.org/10.1007/978-3-031-07082-2_22
22. Jaeger, J.: Nonadaptive one-way to hiding implies adaptive quantum reprogramming. To appear in Proceedings of ASIACERYPT 2025. IACR Cryptol. ePrint Arch. p. 797 (2024), https://eprint.iacr.org/2024/797

23. Jiang, H., Ma, Z., Zhang, Z.: Post-quantum security of key encapsulation mechanism against CCA attacks with a single decapsulation query. In: Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 14441, pp. 434–468. Springer (2023), https://doi.org/10.1007/978-981-99-8730-6_14
24. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Proceedings, Part III. pp. 96–125. Springer (2018). https://doi.org/10.1007/978-3-319-96878-0_4
25. Jost, D., Maurer, U., Mularczyk, M.: Efficient ratcheting: Almost-optimal guarantees for secure messaging. In: Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part I. vol. 11476, pp. 159–188. Springer (2019), https://doi.org/10.1007/978-3-030-17653-2_6
26. Krämer, J., Struck, P.: On quantum ciphertext indistinguishability, recoverability, and OAEP. In: Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022, Proceedings. pp. 286–306. Springer (2022). https://doi.org/10.1007/978-3-031-17234-2_14
27. Kuchta, V., Sakzad, A., Stehlé, D., Steinfeld, R., Sun, S.: Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In: Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part III. pp. 703–728. Springer (2020). https://doi.org/10.1007/978-3-030-45727-3_24
28. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information (10th Anniversary edition). Cambridge University Press (2016)
29. NIST: National institute for standards and technology. post-quantum cryptography standardization process. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization (2017)
30. 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)
31. NTRU: A submission to the nist post-quantum standardization effort. https://ntru.org/ (2018)
32. Pan, J., Zeng, R.: Selective opening security in the quantum random oracle model, revisited. In: Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Proceedings, Part III. pp. 92–122. Springer (2024). https://doi.org/10.1007/978-3-031-57725-3_4
33. Phan, D.H., Pointcheval, D.: OAEP 3-round: A generic and secure asymmetric encryption padding. In: Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings. pp. 63–77. Springer (2004). https://doi.org/10.1007/978-3-540-30539-2_5
34. Shan, T., Ge, J., Xue, R.: 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 Cryptogra-

phy, Proceedings, Part I. pp. 36–64. Springer (2023). https://doi.org/10.1007/978-3-031-31368-4_2
35. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999). https://doi.org/10.1137/S0036144598347011
36. Shoup, V.: OAEP reconsidered. J. Cryptol. **15**(4), 223–249 (2002). https://doi.org/10.1007/S00145-002-0133-9
37. Targhi, E.E., Unruh, D.: Post-quantum security of the fujisaki-okamoto and OAEP transforms. In: Theory of Cryptography Conference. pp. 192–216 (2016). https://doi.org/10.1007/978-3-662-53644-5_8
38. Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., Homma, N.: Curse of re-encryption: A generic power/em analysis on post-quantum kems. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2022**, 296–322 (2022). https://doi.org/10.46586/TCHES.V2022.I1.296-322
39. Winter, A.J.: Coding theorem and strong converse for quantum channels. IEEE Trans. Inf. Theory **45**(7), 2481–2485 (1999). https://doi.org/10.1109/18.796385
40. Yang, Z., Chen, Y., Luo, S.: Two-message key exchange with strong security from ideal lattices. In: Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, Proceedings. vol. 10808, pp. 98–115. Springer (2018), https://doi.org/10.1007/978-3-319-76953-0_6
41. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Proceedings, Part II. pp. 239–268. Springer (2019). https://doi.org/10.1007/978-3-030-26951-7_9

A A Potential Gap in the Security Proof in [13]

In this section, to make it more accurate and coherent, we adhere to the original symbols used in [13] as much as possible.

Game 7: This is identical to Game 6, except it measures all the queries to CStO_G with the projective measurements $\mathbb{M}_{r^*}$. If there is an 1-output measurement, it aborts and returns a random bit.

Game 7:

let $(pk, sk) \leftarrow \mathrm{Gen}(1^n)$, $r^* \xleftarrow{\$} \{0,1\}^{k_0}$, $b \xleftarrow{\$} \{0,1\}$
$\mathbb{M}_{r^*} = \{\mathbb{P}_1 = |r^*\rangle\langle r^*|, \mathbb{P}_0 = \mathbb{I} - \mathbb{P}_1\}$,
run until *there is an 1-output measurement with* $\mathbb{M}_{r^*}$
  let $m_0, m_1 \leftarrow A^{\mathsf{CStO}_H, \mathsf{CStO}_G, \mathbb{U}_{\mathrm{Dec}^{(4)}}}(pk)$
  let $s^* := m_b||0^{k_1} \oplus G(r^*)$, $t^* := r^* \oplus H(s^*)$, $c^* := f(s^*, t^*)$
  let $b' \leftarrow A^{\mathsf{CStO}_H, \mathsf{CStO}_G, \mathbb{U}_{\mathrm{Dec}^{(4)}}}(c^*)$
return $[b = b']$

Fig. 10. Game 7 in the proof of [13].

In [13], the author first changed the simulation of the decryption oracle and finally constructed Game 6, which uses $\mathbb{U}_{\mathrm{Dec}^{(4)}}$ to simulate the decryption oracle, and the running of $\mathbb{U}_{\mathrm{Dec}^{(4)}}$ no longer required the secret key sk. In order to bound the probability $\Pr[1 \leftarrow \text{ Game 6}]$, [13] designed Game 7 as shown in Fig. 10, and then proved the following formula using the gentle-measurement lemma [39]:

$$|\Pr[1 \leftarrow \text{Game 6 }] - \Pr[1 \leftarrow \text{ Game 7}]| \leq 2q_{G_1}\sqrt{2^{-k_0}} + 2q_{G_2}\sqrt{\epsilon}.$$

Here ϵ denotes the security bound of the underlying scheme and $q_{G_1} + q_{G_2} = q_G + 3q_{dec}$, where q_G (resp. q_{dec}) is the total number of queries to the random oracle G (resp. decryption oracle). In fact, the above bound corresponds to the highlighted $q \cdot \sqrt{\epsilon} + q \cdot \sqrt{2^{-k_0}}$ in our Table 1. Next, the following analysis is given in [13] to argue that $\Pr[1 \leftarrow \text{Game 7}] = 1/2$.

"it is clear that Game 7 returns 1 with the probability 1/2 because if one of the measurements returns 1, the output of the game is a random bit. If none of the measurements return 1, $G(r^*)$ remains an uniformly random value for A and consequently $m_b||0^{k_1} \oplus G(r^*)$ is an uniformly random value for A. So the probability that A guesses b is 1/2."

However, the above analysis is not complete. Note that the random oracle G in Game 7 is simulated as a compressed oracle, which means that the $G(r^*)$ used to compute s^* is actually obtained by making the classical query r^* to the compressed oracle. Next, by the basic rules of compressed oracle, the simulation of $G(r^*)$ will process the database state and eventually insert the pair $(r^*, G(r^*))$ into the database state[18]. At this time the key point is that, even if we ensure

[18] More precisely speaking, after this simulation, the database state is a superposition of $\mathsf{StdDecomp}_{r^*}|\mathfrak{D}_G \cup (r^*, G(r^*))\rangle$ for $\mathfrak{D}_G(r^*) = \perp$.

that the adversary A does not query G by r^* to get $G(r^*)$ in Game 7, A can still obtain some information about $G(r^*)$ by querying the decryption oracle, because this oracle is simulated by $\mathbb{U}_{\mathrm{Dec}^{(4)}}$, it will search the whole database state in the computational basis, which may involve the $(r^*, G(r^*))$ pair inserted by the compressed oracle simulation of $G(r^*)$, thereby making A's view not independent of $G(r^*)$. In the following we also give a simple example to support our observation.

- Let database $\mathfrak{D}_H$ satisfy that there exists an element s such that $\mathfrak{D}_H(s) \neq \perp$, $[s]_{k_1} = [s^*]_{k_1}$ and $s \neq s^*$. Then for this s, define $c := f(s, r^* \oplus \mathfrak{D}_H(s))$. Obviously, we have $[G(r^*) \oplus s]_{k_1} = 0^{k_1}$ and $c \neq c^*$ since $s^* := m_b || 0^{k_1} \oplus G(r^*)$ and $s \neq s^*$. Now by the definition of $\mathbb{U}_{\mathrm{Dec}^{(4)}}$ given in [13], the search it performed will transform the state $|c, y, \mathfrak{D}_G \cup (r^*, G(r^*)), \mathfrak{D}_H\rangle$ into $|c, y \oplus [G(r^*) \oplus s]^n, \mathfrak{D}_G \cup (r^*, G(r^*)), \mathfrak{D}_H\rangle$, where the first two parts denote the state on the query register of the decryption oracle. This actually means that the simulation of the decryption oracle on $|c, y, \mathfrak{D}_G \cup (r^*, G(r^*)), \mathfrak{D}_H\rangle$ will leak some information about $G(r^*)$ (i.e. the $[G(r^*) \oplus s]^n$) to the adversary A, and makes its view not independent of $G(r^*)$.

Hence the argument given in the above paragraph is actually not sufficient to demonstrate that $\Pr[1 \leftarrow \text{Game 7}] = 1/2$. In other words, When trying to transform $G(r^*)$ into a fresh random value, it is necessary not only to restrict A not to query G by r^*, but also to consider the potential detection of $G(r^*)$ indirectly derived from the simulation of the decryption oracle.

In addition, it seems non-trivial to fill the above gap in [13] since there are two main obstacles: first, the adversary A can make quantum queries to the decryption oracle since [13] considered the $\mathsf{IND\text{-}qCCA}$ security, which makes it hard to restrict A not to query the decryption oracle by some particular c[19]; second, the powerful Semi-Classical One-Way to Hiding theorem [1] cannot be used here, as it only focuses on the function-like oracle (i.e., $|x, y\rangle \mapsto |x, y \oplus H(x)\rangle$), while the above gap involves the simulation of the compressed oracle and it cannot be rewritten in a function-like form by its definition given in [41].

In our paper, we did not fill the above gap in [13], but instead avoided it by using a new proof route. Roughly speaking, thanks to our step-by-step G and H reprogramming, after replacing $\mathsf{G}(r^*)$ with a fresh random value g^* in our game $\mathbf{G_7}$ (Fig. 8), we can directly use "perform $\mathsf{StdDecopm}^*_r \circ \mathrm{U}^{r^*}_{g^*}$" to visualize the potential detection of the simulation of the decryption oracle, and then eventually bound it by using the Fixed Permutation One-Way to Hiding theorem [22] proposed very recently. Here we emphasize that our method will introduce an additional loss of q_{D}, resulting in the OAEP bound shown in our Table 1 appearing $q_{\mathsf{D}} \cdot q \cdot \sqrt{\epsilon} + q_{\mathsf{D}} \cdot q \cdot \sqrt{2^{-k_0}}$, which is a tolerable drawback when q_{D} is small. Nevertheless, either proposing concrete fixes to eliminate this drawback or directly filling the gap in [13], are still very interesting theoretical questions, and we leave them for further investigation.

[19] For instance, the c we designed in the above example.

B Quantum Background

A quantum system (register) Q is a complex Hilbert space $\mathcal{H}_Q$ with an inner product $\langle\cdot|\cdot\rangle$, and notation like '$|\cdot\rangle$' or '$\langle\cdot|$' is called Dirac notation. A quantum state on Q is a vector on $\mathcal{H}_Q$. We denote $\mathcal{H}_Q = \mathbb{C}[X]$ if Q is defined over a finite set X. The orthonormal basis of $\mathbb{C}[X]$ is $\{|x\rangle\}_{x\in X}$, where the basis state $|x\rangle$ is labeled by the element x of X. We also refer to $\{|x\rangle\}_{x\in X}$ as the computational basis. The norm of a state $|\psi\rangle$ is defined as $\||\psi\rangle\| := \sqrt{\langle\psi|\psi\rangle}$, and we say $|\psi\rangle$ is a unit state if $\||\psi\rangle\| = 1$. The norm of an operator A is defined as $\|A\| := \max_{|\psi\rangle} \|A|\psi\rangle\|$, where max is over all unit states on Q. Sometimes we also write $|\psi\rangle$ as $|\psi\rangle_Q$ to emphasize that $|\psi\rangle$ is a quantum state on Q.

Composite quantum system. Given quantum systems Q_1 and Q_2, we call the tensor product $Q_1 \otimes Q_2$ the composite quantum system, the state of which is written as $\sum_{x,y} \alpha_{x,y}|x\rangle \otimes |y\rangle \in Q_1 \otimes Q_2$ where $|x\rangle \in Q_1$, $|y\rangle \in Q_2$ are the basis states of Q_1 and Q_2. We sometimes also abbreviate $|x\rangle \otimes |y\rangle$ into $|x, y\rangle$ for simplicity. If the composite quantum system $Q_1 \otimes Q_2$ is in state $|\psi\rangle$, we say that systems Q_0 and Q_1 are entangled if $|\psi\rangle$ can not be written as a product state $|\psi_1\rangle \otimes |\psi_2\rangle$ (here $|\psi_1\rangle \in Q_1$, $|\psi_2\rangle \in Q_2$).

A single qubit is a quantum system defined over $\{0, 1\}$. A single 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 $|b\rangle$. An n-qubit system is $Q^{\otimes n}$ where Q is a single-qubit system.

Evolution of quantum systems. A pure state $|\psi\rangle$ can be manipulated by performing a unitary operation U, and the resulting state can be denoted as $|\psi'\rangle = \mathrm{U}|\psi\rangle$. For any unitary operation U, we have $\mathrm{UU}^\dagger = \mathrm{I}$, where $\mathrm{U}^\dagger$ is the Hermitian transpose of U and I is the identity operator.

Projector on the quantum system. We introduce a special operation called projector. For the state $|\psi\rangle$ of a quantum system defined over a finite set X, a projector M_S ($S \subseteq X$) applies the operation $\sum_{y\in S} |y\rangle\langle y|$ to $|\psi\rangle$ to get the new state $\sum_{y\in S} |y\rangle\langle y|\psi\rangle$. We stress that any projector M is Hermitian (i.e., we have $\mathrm{M}^\dagger = \mathrm{M}$) and idempotent (i.e., we have $\mathrm{M}^2 = \mathrm{M}$).

Basic measurement. The state $|\psi\rangle$ in a quantum system can be measured in the computational basis. For example, suppose $|\psi\rangle = \sum_{x\in X} \alpha_x|x\rangle$ ($\sum_{x\in X} |\alpha_x|^2 = 1$) with computational basis $\{|x\rangle\}_{x\in X}$, if we measure $|\psi\rangle$ in basis $\{|x\rangle\}_{x\in X}$, we obtain the measurement result x with probability $|\langle x|\psi\rangle|^2 = |\alpha_x|^2$. Given that result x is obtained, the state $|\psi\rangle$ collapses to state $|x\rangle$. Moreover, we explain an important special class of measurements called projective measurement which is defined by a set of projectors $\mathrm{M}_1, \ldots, \mathrm{M}_n$ where M_i are mutually orthogonal and $\sum_{i=1}^n \mathrm{M}_i = \mathrm{I}$. Upon measuring the state $|\psi\rangle$, the probability of obtaining i ($i \in \{1, \ldots, n\}$) is $\langle\psi|\mathrm{M}_i|\psi\rangle$ ($= \|\mathrm{M}_i|\psi\rangle\|^2$). Given that result i is obtained, the state $|\psi\rangle$ collapses to state $\mathrm{M}_i|\psi\rangle/\sqrt{\langle\psi|\mathrm{M}_i|\psi\rangle}$.

Quantum oracle algorithm. A quantum oracle algorithm $\mathcal{A}$ is an algorithm that has quantum access to oracles, and that can perform a mix of classical and quantum unitary algorithms. In this paper, without loss of generality, we default that $\mathcal{A}$'s final output is a single bit $b \in \{0,1\}$. We say that a quantum oracle algorithm $\mathcal{A}^{\mathsf{H}}(z)$ making q oracle queries is a unitary quantum oracle algorithm if its entire execution can be described as

$$\mathbb{M}_{\mathcal{A}} \circ \mathrm{U}_q \circ \mathrm{O}_{\mathsf{H}} \circ \mathrm{U}_{q-1} \circ \mathrm{O}_{\mathsf{H}} \cdots \mathrm{U}_2 \circ \mathrm{O}_{\mathsf{H}} \circ \mathrm{U}_1 \circ \mathrm{O}_{\mathsf{H}} |\psi_z\rangle.$$

Here $|\psi_z\rangle$ is an initial pure state that may depend on input z, $\mathrm{U}_1, \ldots, \mathrm{U}_q$ are the fixed unitary operations applied between oracle queries, and O_{H} is the unitary operation used to implement the oracle H. $\mathbb{M}_{\mathcal{A}} := \{\mathrm{M}_0, \mathrm{M}_1\}$ is the final binary projective measurement performed by $\mathcal{A}$ on its quantum register, and the measurement result of $\mathbb{M}_{\mathcal{A}}$ (0 or 1) is the final output of $\mathcal{A}$. Next, we introduce the following well-known fact about the quantum oracle algorithm. This fact implies that we can directly assume the quantum oracle algorithms to be unitary.

Fact 1. *Any quantum oracle algorithm can be transformed into a unitary quantum oracle algorithm with constant factor computational overhead and the same query number.*

Representation of the special symbol $\perp$. In this paper, we sometimes use a special symbol $\perp$ to expand a finite set $\{0,1\}^n$. Roughly speaking, the reason is that when we define a particular unitary operation, we need "$\perp$" to denote "not defined (yet)" or "computational failure".

As for the detailed representation of $\{0,1\}^n \cup \{\perp\}$, we use the extension method used in [9,19]. 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 $x, y \in \{0,1\}^n \cup \{\perp\}$ actually refers to $enc(x) \oplus enc(y)$, and the quantum register defined over the set $\{0,1\}^n \cup \{\perp\}$ actually refers to the register defined over the set $\{0,1\}^{n+1}$.

Here we also give a more detailed example. Let c_1, c_2 be two ciphertexts with $\mathsf{Dec}(sk, c_1) = m \in \{0,1\}^n$ and $\mathsf{Dec}(sk, c_2) = \perp$. So when making one quantum query to the decryption oracle using the uniform superposition of c_1 and c_2, the adversary can prepare the query state as $1/\sqrt{2}|c_1\rangle_{\mathsf{I}}|0||y\rangle_{\mathsf{O}} + 1/\sqrt{2}|c_2\rangle_{\mathsf{I}}|0||y\rangle_{\mathsf{O}}$, where I/O is the input/output register and y is a fixed value in $\{0,1\}^n$. In fact, the 0 padded in front of y denotes that the current value of the output register is exactly y, not a value like $y \oplus \perp$. After the decryption query, the adversary will get $1/\sqrt{2}|c_1\rangle_{\mathsf{I}}|0||y \oplus m\rangle_{\mathsf{O}} + 1/\sqrt{2}|c_2\rangle_{\mathsf{I}}|1||y\rangle_{\mathsf{O}}$ since we denote $\perp$ by $1||0^n$. Now the adversary "knows"[20] that c_2 failed to decrypt, because it gets the output $1||y$ and finds that this value is actually $y \oplus \perp$.

[20] In fact, an adversary needs to perform a measurement to know something, so here is an intuitive explanation, but this does not lose generality.

C Cryptographic Primitives and Security Definitions

Definition 4 (Public key encryption). *A public key encryption (PKE) scheme* $\mathsf{P} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *consists of three polynomial-time algorithms (in the security parameter* λ*) with a finite message space* $\mathcal{M}$ *such that*

1. Gen*, the key generation algorithm, is a probabilistic algorithm that on input* 1^λ *outputs a public/secret key pair* $(\mathsf{pk}, \mathsf{sk})$*.*
2. Enc*, the encryption algorithm, on input pk and a message* $m \in \mathcal{M}$*, it chooses* $r \leftarrow_\$ \mathcal{R}$ *(*$\mathcal{R}$ *is the randomness space), computes* $c = \mathsf{Enc}_{\mathsf{pk}}(m; r) \in \mathcal{C}$ *(*$\mathcal{C}$ *is the ciphertext space) and outputs the ciphertext* c*. If* Enc *does not use randomness* r *to compute* c*,* Enc *is a deterministic algorithm and outputs* $c = \mathsf{Enc}_{\mathsf{pk}}(m)$*.*
3. Dec*, the decryption algorithm, is a deterministic algorithm that on input ciphertext* $c \in \mathcal{C}$ *outputs a message* $m = \mathsf{Dec}_{\mathsf{sk}}(c)$*, or a special symbol* $\perp \notin \mathcal{M}$ *to indicate that* c *is not a valid ciphertext.*

Definition 5 (IND-qCCA security for PKE scheme). *For any polynomial-time quantum adversary* $\mathcal{A}$*, we define its* IND-qCCA *advantage against the PKE scheme* $\mathsf{P} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

$$\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{P},\mathcal{A}} := \left| \Pr[1 \leftarrow \mathbf{G}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{P},\mathcal{A}}] - \frac{1}{2} \right|.$$

Here game $\mathbf{G}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{P},\mathcal{A}}$ *is shown in Fig. 11. We say that scheme* P *is* IND-qCCA *secure if for any polynomial-time quantum adversary* $\mathcal{A}$*, the* $\mathrm{Adv}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{P},\mathcal{A}}$ *is negligible.*

$\mathbf{G}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{P},\mathcal{A}}$	$\mathsf{qDec}(\lvert c, z\rangle)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$	**if** c^* has not been defined
$(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{qDec}}(\mathsf{pk})$	**return** $\lvert c, z \oplus \mathsf{Dec}_{\mathsf{sk}}(\mathsf{c})\rangle$
$b \leftarrow_\$ \{0,1\}$	**else return** $\lvert c, z \oplus f_{c^*}(c)\rangle$
$c^* = \mathsf{Enc}_{\mathsf{pk}}(m_b^*)$	$f_{c^*}(c)$
$b' \leftarrow \mathcal{A}^{\mathsf{qDec}}(\mathsf{pk}, c^*)$	**if** $c = c^*$
return $[\![b' = b]\!]$	**return** $\perp$
	else return $\mathsf{Dec}_{\mathsf{sk}}(c)$

Fig. 11. Game $\mathbf{G}^{\mathsf{IND\text{-}qCCA}}_{\mathsf{P},\mathcal{A}}$.

We also define IND-qCCA security of PKE schemes in the QROM, where the adversary $\mathcal{A}$ can make quantum queries to random oracles.

Definition 6 (Trapdoor injective function). *A trapdoor injective function* $\mathsf{TDI} = (\mathsf{Gen}, f, f^{-1})$ *consists of one polynomial-time algorithm and two functions (in the security parameter* λ*) such that:*

- Gen *is the key generation algorithm, which generates a public/secret key pair* $(\mathsf{pk}, \mathsf{sk})$ *on input* 1^λ*.*
- $f_{\mathsf{pk}}(\cdot)$ *is an injective function with domain/range space* $\mathcal{X}/\mathcal{Y}$ *for all* pk*.*
- $f_{\mathsf{sk}}^{-1}(\cdot)$ *is the inverse function of* $f_{\mathsf{pk}}(\cdot)$*. That is to say, for any* $y \in \mathcal{Y}$*,*

$$f_{\mathsf{sk}}^{-1}(y) = \begin{cases} x, & if\ \exists x \in \mathcal{X}\ s.t. f_{\mathsf{pk}}(x) = y, \\ \perp, & otherwise. \end{cases}$$

The above definition of f_{sk}^{-1} *introduces no ambiguity since* f *is an injective function.*

Definition 7 (Quantum partial domain one-wayness for trapdoor injective function). *For any polynomial-time quantum adversary* $\mathcal{A}$*, we define its* PD-OW *advantage against the trapdoor injective function* $\mathsf{TDI} = (\mathsf{Gen}, f, f^{-1})$ *as follows:*

$$\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{A}} := \Pr[1 \leftarrow \mathbf{G}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{A}}].$$

Here game $\mathbf{G}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{A}}$ *is shown in Fig. 12. We say that* TDI *is* PD-OW *secure if for any polynomial-time quantum adversary* $\mathcal{A}$*,* $\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{A}}$ *is negligible.*

$\mathbf{G}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{A}}$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$
2 : $(s^*, t^*) \leftarrow_{\$} \mathcal{X}$
3 : $s \leftarrow \mathcal{A}(\mathsf{pk}, f_{\mathsf{pk}}(s^*, t^*))$
4 : **return** $[\![s = s^*]\!]$

Fig. 12. Game $\mathbf{G}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{A}}$.

D Missing Proofs in Section 4

Lemma 9 (Measurement Prob v.s. Norm Distance [1, Lemma 3, 4]). *Let* $|\Phi_1\rangle$, $|\Phi_2\rangle$ *be two pure unit states. Let* $\mathbb{M} = \{\mathrm{M}_0, \mathrm{M}_1\}$ *be a binary projective measurement. Let* P_i *be the probability that* $\mathbb{M}$ *returns 1 when measuring* $|\Phi_i\rangle$ $(i \in \{1, 2\})$*. Then we have* $|\mathrm{P}_1 - \mathrm{P}_2| \leq \||\Phi_1\rangle - |\Phi_2\rangle\|$.

Lemma 10 ([16, Corollary 2]). *For any quantum states* $|\psi_1\rangle, \ldots, |\psi_q\rangle$*, we have* $\left\|\sum_{i=1}^{q} |\psi_i\rangle\right\|^2 \leq q \cdot \sum_{i=1}^{q} \||\psi_i\rangle\|^2$.

D.1 Proof of Lemma 1

Let qDec_1 denote the oracle that simulated by $\mathrm{U}^1_{\mathsf{qDec}}$ (resp. $\mathrm{U}^{1,*}_{\mathsf{qDec}}$) before (resp. after) c^* is defined. Then, the running process of game $\mathbf{G_1}$ under a fixed $(\mathsf{pk},\mathsf{sk},\mathsf{H})$ pair can be rewritten as the following artificial quantum oracle algorithm $\mathcal{B}^{\mathsf{qDec}_1}$:

– After receiving $(\mathsf{pk},\mathsf{sk},\mathsf{H})$, $\mathcal{B}$ runs the adversary $\mathcal{A}$ in game $\mathbf{G}_1$. When $\mathcal{A}$ queries oracle G or H, $\mathcal{B}$ uses the method of game $\mathbf{G}_1$ to simulate. When $\mathcal{A}$ queries oracle qDec, $\mathcal{B}$ replies by querying its oracle qDec_1. $\mathcal{B}$ finally gets b' from $\mathcal{A}$ and outputs $[\![b=b']\!]$.

Next, if we change the oracle qDec_1 into qDec_2, which is the same as qDec_1 except that the $\mathrm{U}^1_{\mathsf{qDec}}$ and $\mathrm{U}^{1,*}_{\mathsf{qDec}}$ is replaced by $\mathrm{U}^2_{\mathsf{qDec}}$ and $\mathrm{U}^{2,*}_{\mathsf{qDec}}$, respectively, the resulting quantum oracle algorithm $\mathcal{B}^{\mathsf{qDec}_2}(\mathsf{pk},\mathsf{sk},\mathsf{H})$ will run game $\mathbf{G}_2$. Hence,

$$\begin{aligned}\Pr\left[1\leftarrow\mathcal{B}^{\mathsf{qDec}_1}(\mathsf{pk},\mathsf{sk},\mathsf{H})\right]&=\Pr\left[1\leftarrow\mathbf{G}_1:(\mathsf{pk},\mathsf{sk},\mathsf{H})\right],\\ \Pr\left[1\leftarrow\mathcal{B}^{\mathsf{qDec}_2}(\mathsf{pk},\mathsf{sk},\mathsf{H})\right]&=\Pr\left[1\leftarrow\mathbf{G}_2:(\mathsf{pk},\mathsf{sk},\mathsf{H})\right].\end{aligned}\tag{28}$$

Here $\Pr[1\leftarrow\mathbf{G}_i:(\mathsf{pk},\mathsf{sk},\mathsf{H})]$ denotes the probability of game $\mathbf{G}_i$ ($i\in[2]$) outputs 1 under a fixed $(\mathsf{pk},\mathsf{sk},\mathsf{H})$ pair.

As introduced in Supplementary Material B, any quantum oracle algorithm can be transformed into a unitary one. Thus, for the $\mathcal{B}^{\mathcal{O}}(\mathsf{pk},\mathsf{sk},\mathsf{H})$, we can construct its unitary variant $\mathcal{B}^{\mathcal{O}}_{\mathrm{U}}(\mathsf{pk},\mathsf{sk},\mathsf{H})$ that acts on the registers $\mathsf{Z},\mathsf{I},\mathsf{O}$ and $\mathsf{D}_{q_\mathsf{G}}$, where I/O is the input/output register of the oracle $\mathcal{O}$, $\mathsf{D}_{q_\mathsf{G}}$ is the database register used to reply $\mathcal{A}$'s G queries, and Z denotes some other ancillary registers used during the execution of $\mathcal{B}^{\mathcal{O}}_{\mathrm{U}}(\mathsf{pk},\mathsf{sk},\mathsf{H})$. Now the joint state of $\mathcal{B}^{\mathsf{qDec}_1}_{\mathrm{U}}(\mathsf{pk},\mathsf{sk},\mathsf{H})$ and $\mathcal{B}^{\mathsf{qDec}_2}_{\mathrm{U}}(\mathsf{pk},\mathsf{sk},\mathsf{H})$ just before the final binary projective measurement $\mathbb{M}:=\{\mathrm{M}_0,\mathrm{M}_1\}$ can be written as:

$$\begin{aligned}\mathcal{B}^{\mathsf{qDec}_1}_{\mathrm{U}}:|\Psi_1\rangle|0\rangle_{\mathsf{ST}}&=\mathrm{U}_{q_\mathsf{D}}\circ\mathrm{U}^{1,*}_{\mathsf{qDec}}\circ\cdots\mathrm{U}_1\circ\mathrm{U}^1_{\mathsf{qDec}}|\psi\rangle|0\rangle_{\mathsf{ST}},\\ \mathcal{B}^{\mathsf{qDec}_2}_{\mathrm{U}}:|\Psi_2\rangle|0\rangle_{\mathsf{ST}}&=\mathrm{U}_{q_\mathsf{D}}\circ\mathrm{U}^{2,*}_{\mathsf{qDec}}\circ\cdots\mathrm{U}_1\circ\mathrm{U}^2_{\mathsf{qDec}}|\psi\rangle|0\rangle_{\mathsf{ST}}.\end{aligned}\tag{29}$$

Here $|\psi\rangle$ is the initial state on the registers $\mathsf{Z},\mathsf{I},\mathsf{O}$ and $\mathsf{D}_{q_\mathsf{G}}$, and $\mathrm{U}_1,\ldots,\mathrm{U}_{q_\mathsf{D}}$ are unitary operations acting on these registers between the $\mathsf{qDec}_1/\mathsf{qDec}_2$ queries. Obviously, we have

$$\begin{aligned}\Pr[1\leftarrow\mathcal{B}^{\mathsf{qDec}_1}(\mathsf{pk},\mathsf{sk},\mathsf{H})]&=\Pr[1\leftarrow\mathcal{B}^{\mathsf{qDec}_1}_{\mathrm{U}}(\mathsf{pk},\mathsf{sk},\mathsf{H})]=\|\mathrm{M}_1|\Psi_1\rangle|0\rangle_{\mathsf{ST}}\|^2,\\ \Pr[1\leftarrow\mathcal{B}^{\mathsf{qDec}_2}(\mathsf{pk},\mathsf{sk},\mathsf{H})]&=\Pr[1\leftarrow\mathcal{B}^{\mathsf{qDec}_2}_{\mathrm{U}}(\mathsf{pk},\mathsf{sk},\mathsf{H})]=\|\mathrm{M}_1|\Psi_2\rangle|0\rangle_{\mathsf{ST}}\|^2.\end{aligned}\tag{30}$$

As proved in Supplementary Material D.2, $\||\Psi_1\rangle|0\rangle_{\mathsf{ST}}-|\Psi_2\rangle|0\rangle_{\mathsf{ST}}\|\le 8q_\mathsf{D}\cdot\sqrt{2^{-k_1}}$. Then by using Lemma 9 and Eq. (30), we have

$$\left|\Pr\left[1\leftarrow\mathcal{B}^{\mathsf{qDec}_1}(\mathsf{pk},\mathsf{sk},\mathsf{H})\right]-\Pr\left[1\leftarrow\mathcal{B}^{\mathsf{qDec}_2}(\mathsf{pk},\mathsf{sk},\mathsf{H})\right]\right|\le 8q_\mathsf{D}\cdot\sqrt{2^{-k_1}}.$$

By combining the above equation with Eq. (28) and averaging the $(\mathsf{pk},\mathsf{sk},\mathsf{H})$, we finally get $|\Pr[1\leftarrow\mathbf{G_1}]-\Pr[1\leftarrow\mathbf{G_2}]|\le 8q_\mathsf{D}\cdot\sqrt{2^{-k_1}}$.

D.2 Bound on $\||\Psi_1\rangle|0\rangle_{\mathsf{ST}} - |\Psi_2\rangle|0\rangle_{\mathsf{ST}}\|$

In the running process of $\mathcal{B}_{\mathrm{U}}^{\mathsf{qDec}_1}$ and $\mathcal{B}_{\mathrm{U}}^{\mathsf{qDec}_2}$, the compressed standard oracle is implemented based on $\mathsf{D}_{q_\mathsf{G}}$. Recall that the database state $|D_\mathsf{G}\rangle$ on the database register $\mathsf{D}_{q_\mathsf{G}}$ must satisfy $n(D_\mathsf{G}) \leq q_\mathsf{G}$, and the simulation of the decryption query[21] also needs to perform two compressed standard oracle queries. Hence, we can conclude that the database state $|D_\mathsf{G}\rangle$ just before one decryption query must satisfy $n(D_\mathsf{G}) \leq q_\mathsf{G} - 2$. Based on this fact, we denote the unit joint state on the registers $\mathsf{Z}, \mathsf{I}, \mathsf{O}, \mathsf{D}_{q_\mathsf{G}}$ just before performing one decryption query as:

$$|\Phi\rangle := \sum_{\substack{z\in\{0,1\}^*, c\in\mathcal{C}, y\in\{0,1\}^{n+k_1}\cup\{\perp\}, D_\mathsf{G}\in\mathbf{D}_{q_\mathsf{G}} \\ n(D_\mathsf{G})\leq q_\mathsf{G}-2}} \alpha_{y,D_\mathsf{G}}^{z,c} |z\rangle_\mathsf{Z}|c,y\rangle_{\mathsf{IO}}|D_\mathsf{G}\rangle_{\mathsf{D}_{q_\mathsf{G}}}.$$

Here $\mathcal{C}$ is the ciphertext space and set $\mathbf{D}_{q_\mathsf{G}}$ follows Definition 2. In the following proof, for simplicity, we omit the set notations such as "$\in \{0,1\}^*$" used in the sum of the above equation, and abbreviate $|z\rangle_\mathsf{Z}|c,y\rangle_{\mathsf{IO}}|D_\mathsf{G}\rangle_{\mathsf{D}_{q_\mathsf{G}}}$ into $|z,c,y,D_\mathsf{G}\rangle$.

In order to compute the bound separately, we divide $|\Phi\rangle$ into four complete and mutually orthogonal parts:

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

where

$$|\Phi_1\rangle := \sum_{\substack{z,c,y,D_\mathsf{G}, n(D_\mathsf{G})\leq q_\mathsf{G}-2 \\ f_{\mathsf{sk}}^{-1}(c)=\perp}} \beta_{y,D_\mathsf{G}}^{z,c} |z,c,y,D_\mathsf{G}\rangle,$$

$$|\Phi_2\rangle := \sum_{\substack{z,c,y,D_\mathsf{G}, n(D_\mathsf{G})\leq q_\mathsf{G}-2 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s), D_\mathsf{G}(r)=\perp}} \beta_{y,D_\mathsf{G}}^{z,c} |z,c,y,D_\mathsf{G}\rangle,$$

$$|\Phi_3\rangle := \sum_{\substack{z,c,y,D_\mathsf{G}, n(D_\mathsf{G})\leq q_\mathsf{G}-3 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s), D_\mathsf{G}(r)=\perp \\ u\in\{0,1\}^{n+k_1}, u\neq 0^{n+k_1}}} \frac{\beta_{y,D_\mathsf{G},u}^{z,c}}{\sqrt{2^{n+k_1}}} \sum_{y_1\in\{0,1\}^{n+k_1}} (-1)^{y_1\cdot u} |z,c,y,D_\mathsf{G}\cup(r,y_1)\rangle,$$

$$|\Phi_4\rangle := \sum_{\substack{z,c,y,D_\mathsf{G}, n(D_\mathsf{G})\leq q_\mathsf{G}-3 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s), D_\mathsf{G}(r)=\perp \\ u=0^{n+k_1}}} \frac{\beta_{y,D_\mathsf{G},u}^{z,c}}{\sqrt{2^{n+k_1}}} \sum_{y_1\in\{0,1\}^{n+k_1}} |z,c,y,D_\mathsf{G}\cup(r,y_1)\rangle.$$

We first compute the upper bound of $\|(\mathrm{U}_{\mathsf{qDec}}^1 - \mathrm{U}_{\mathsf{qDec}}^2)|\Phi\rangle|0\rangle_{\mathsf{ST}}\|$, which corresponds to the case that the challenge ciphertext c^* has not been computed yet. For the fixed tuple (z,c,y,D_G) with $f_{\mathsf{sk}}^{-1}(c) = (s,t) \neq \perp$, and $D_\mathsf{G}(r) = \perp$

[21] In our setting, this query is simulated by using $\mathrm{U}_{\mathsf{qDec}}^1/\mathrm{U}_{\mathsf{qDec}}^{1,*}$ or $\mathrm{U}_{\mathsf{qDec}}^2/\mathrm{U}_{\mathsf{qDec}}^{2,*}$.

$(r := t \oplus \mathsf{H}(s))$, we define the following states that will be used later. Here $w \in \{0,1\}^{n+k_1}$ and $\nu \in \{0,1\}^n \cup \{\bot\}$.

$$
\begin{aligned}
|\gamma_1[w,\nu]\rangle^{z,c}_{y,D_{\mathsf{G}}} &:= \sum_{[y_1]_{k_1}=[s]_{k_1}} (-1)^{y_1\cdot w}|z,c,y\oplus\nu,D_{\mathsf{G}}\cup(r,y_1)\rangle,\\
|\gamma_2[w,\nu]\rangle^{z,c}_{y,D_{\mathsf{G}}} &:= \sum_{[y_1]_{k_1}=[s]_{k_1}} (-1)^{y_1\cdot w}|z,c,y\oplus\nu,D_{\mathsf{G}}\rangle,\\
|\gamma_3[w,\nu]\rangle^{z,c}_{y,D_{\mathsf{G}}} &:= \sum_{[y_1]_{k_1}=[s]_{k_1}} \sum_{y_2\in\{0,1\}^{n+k_1}} \frac{(-1)^{y_1\cdot w}}{\sqrt{2^{n+k_1}}}|z,c,y\oplus\nu,D_{\mathsf{G}}\cup(r,y_2)\rangle.
\end{aligned}
\tag{31}
$$

Let S_x be an abbreviation of the local decompression procedure $\mathsf{StdDecomp}_x$. Then by the quantum simulation of the unitary operation $\mathrm{U}^1_{\mathsf{qDec}}$ (resp. $\mathrm{U}^2_{\mathsf{qDec}}$) given in Eq. (5) (resp. Eq. (7)), we can compute

$$
\mathrm{U}^1_{\mathsf{qDec}}|\Phi_1\rangle|0\rangle_{\mathsf{ST}} = \mathrm{U}^2_{\mathsf{qDec}}|\Phi_1\rangle|0\rangle_{\mathsf{ST}} = \sum_{\substack{z,c,y,D_{\mathsf{G}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-2\\ f_{\mathsf{sk}}^{-1}(c)=\bot}} \beta^{z,c}_{y,D_{\mathsf{G}}}|z,c,y\oplus\bot,D_{\mathsf{G}}\rangle|0\rangle_{\mathsf{ST}},
$$

$$
\begin{aligned}
&(\mathrm{U}^1_{\mathsf{qDec}}-\mathrm{U}^2_{\mathsf{qDec}})|\Phi_2\rangle|0\rangle_{\mathsf{ST}}\\
&\quad= \sum_{\substack{z,c,y,D_{\mathsf{G}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-2\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\bot\\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\bot}} \frac{\beta^{z,c}_{y,D_{\mathsf{G}}}}{\sqrt{2^{n+k_1}}}\mathsf{S}_r\begin{bmatrix}|\gamma_1[0^{n+k_1},[s\oplus y_1]^n]\rangle\\ -|\gamma_1[0^{n+k_1},\bot]\rangle\end{bmatrix}|0\rangle_{\mathsf{ST}},
\end{aligned}
$$

$$
\begin{aligned}
&(\mathrm{U}^1_{\mathsf{qDec}}-\mathrm{U}^2_{\mathsf{qDec}})|\Phi_3\rangle|0\rangle_{\mathsf{ST}}\\
&\quad= \sum_{\substack{z,c,y,D_{\mathsf{G}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-3\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\bot\\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\bot\\ u\in\{0,1\}^{n+k_1},u\neq 0^{n+k_1}}} \frac{\beta^{z,c,y}_{D_{\mathsf{G}},u}}{2^{n+k_1}}\begin{bmatrix}|\gamma_2[u,[s\oplus y_1]^n]\rangle-|\gamma_2[u,\bot]\rangle\\ -|\gamma_3[u,[s\oplus y_1]^n]\rangle+|\gamma_3[u,\bot]\rangle\end{bmatrix}|0\rangle_{\mathsf{ST}}.
\end{aligned}
$$

Next, we note that the state in the form $\sqrt{1/2^{n+k_1}}\sum_{y_1}|z,c,y,D_{\mathsf{G}}\cup(r,y_1)\rangle$ cannot emerge if we use $\mathrm{U}^1_{\mathsf{qDec}}/\mathrm{U}^{1,*}_{\mathsf{qDec}}$ to simulate the decryption query, since the quantum simulation of $\mathrm{U}^1_{\mathsf{qDec}}/\mathrm{U}^{1,*}_{\mathsf{qDec}}$ as shown in Eq. (5) only queries the compressed standard oracle and does not disturb its implementation[22]. However, the state in that form might emerge if we use $\mathrm{U}^2_{\mathsf{qDec}}/\mathrm{U}^{2,*}_{\mathsf{qDec}}$ to simulate the decryption query, since the unitary operation$\mathsf{Test}_{\mathsf{G}}$ used by them will disturb the implementation of the compressed standard oracle. Hence, in order to compute the difference between $\mathrm{U}^1_{\mathsf{qDec}}|\Phi_4\rangle|0\rangle_{\mathsf{ST}}$ and $\mathrm{U}^2_{\mathsf{qDec}}|\Phi_4\rangle|0\rangle_{\mathsf{ST}}$, for any z,c,y,D_{G},r

[22] In fact, the uniform superposition of database state like $\sqrt{1/2^{n+k_1}}\sum_{y_1}|D_{\mathsf{G}}\cup(r,y_1)\rangle$ cannot emerge when we implement a compressed standard oracle without being disturbed. This property is also used in the proof of Lemma 5 in [41].

such that $D_{\mathsf{G}}(r) = \perp$ and $n(D_{\mathsf{G}}) \leq q_{\mathsf{G}} - 1$, we add a complement of $\mathrm{U}^1_{\mathsf{qDec}}$ as:

$$\mathrm{U}^1_{\mathsf{qDec}} \frac{1}{\sqrt{2^{n+k_1}}} \sum_{y_1 \in \{0,1\}^{n+k_1}} |z, c, y, D_{\mathsf{G}} \cup (r, y_1)\rangle |0\rangle_{\mathsf{ST}}$$
$$:= \frac{1}{\sqrt{2^{n+k_1}}} \sum_{y_1 \in \{0,1\}^{n+k_1}} |z, c, y \oplus \perp, D_{\mathsf{G}} \cup (r, y_1)\rangle |0\rangle_{\mathsf{ST}}.$$

Since the (uniform superposition of the database) state $\sqrt{1/2^{n+k_1}} \sum_{y_1 \in \{0,1\}^{n+k_1}} |z, c, y, D_{\mathsf{G}} \cup (r, y_1)\rangle$ must be orthogonal with $|\Phi_1\rangle$, $|\Phi_2\rangle$, $|\Phi_3\rangle$ and applying $\mathrm{U}^1_{\mathsf{qDec}}$ on all $|\Phi_1\rangle$, $|\Phi_2\rangle$, $|\Phi_3\rangle$ cannot generate such uniform superposition of the database, we know that the above complement causes no ambiguity and is workable. With this complement, we can compute

$$(\mathrm{U}^1_{\mathsf{qDec}} - \mathrm{U}^2_{\mathsf{qDec}})|\Phi_4\rangle|0\rangle_{\mathsf{ST}}$$
$$= \sum_{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}}) \leq q_{\mathsf{G}} - 3 \\ f_{\mathsf{sk}}^{-1}(c) = (s,t) \neq \perp \\ r := t \oplus \mathsf{H}(s), D_{\mathsf{G}}(r) = \perp \\ u = 0^{n+k_1}}} \frac{\beta^{z,c}_{y,D_{\mathsf{G}},u}}{\sqrt{2^{n+k_1}}} \begin{bmatrix} |\gamma_1[0^{n+k_1}, \perp]\rangle - \\ |\gamma_1[0^{n+k_1}, [s \oplus y_1]^n]\rangle \end{bmatrix} |0\rangle_{\mathsf{ST}}.$$

At this point, we have $\|(\mathrm{U}^1_{\mathsf{qDec}} - \mathrm{U}^2_{\mathsf{qDec}})|\Phi_1\rangle|0\rangle_{\mathsf{ST}}\| = 0$, and we can compute

$$\left\|(\mathrm{U}^1_{\mathsf{qDec}} - \mathrm{U}^2_{\mathsf{qDec}})|\Phi_2\rangle|0\rangle_{\mathsf{ST}}\right\|^2$$
$$= \left\| \sum_{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}}) \leq q_{\mathsf{G}} - 2 \\ f_{\mathsf{sk}}^{-1}(c) = (s,t) \neq \perp \\ r := t \oplus \mathsf{H}(s), D_{\mathsf{G}}(r) = \perp}} \frac{\beta^{z,c}_{y,D_{\mathsf{G}}}}{\sqrt{2^{n+k_1}}} \mathsf{S}_r \left[|\gamma_1[0^{n+k_1}, [s \oplus y_1]^n]\rangle - |\gamma_1[0^{n+k_1}, \perp]\rangle \right] \right\|^2$$
$$\overset{(a)}{\leq} 2 \left\| \sum_{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}}) \leq q_{\mathsf{G}} - 2 \\ f_{\mathsf{sk}}^{-1}(c) = (s,t) \neq \perp \\ r := t \oplus \mathsf{H}(s), D_{\mathsf{G}}(r) = \perp}} \frac{\beta^{z,c}_{y,D_{\mathsf{G}}}}{\sqrt{2^{n+k_1}}} \mathsf{S}_r |\gamma_1[0^{n+k_1}, [s \oplus y_1]^n]\rangle \right\|^2$$
$$+ 2 \left\| \sum_{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}}) \leq q_{\mathsf{G}} - 2 \\ f_{\mathsf{sk}}^{-1}(c) = (s,t) \neq \perp \\ r := t \oplus \mathsf{H}(s), D_{\mathsf{G}}(r) = \perp}} \frac{\beta^{z,c}_{y,D_{\mathsf{G}}}}{\sqrt{2^{n+k_1}}} \mathsf{S}_r |\gamma_1[0^{n+k_1}, \perp]\rangle \right\|^2$$
$$\overset{(b)}{=} 2 \sum_{\substack{z,c,D_{\mathsf{G}} \\ n(D_{\mathsf{G}}) \leq q_{\mathsf{G}} - 2 \\ f_{\mathsf{sk}}^{-1}(c) = (s,t) \neq \perp \\ r := t \oplus \mathsf{H}(s), D_{\mathsf{G}}(r) = \perp}} \left\| \sum_{y} \frac{\beta^{z,c}_{y,D_{\mathsf{G}}}}{\sqrt{2^{n+k_1}}} |\gamma_1[0^{n+k_1}, [s \oplus y_1]^n]\rangle \right\|^2$$

$$
\begin{aligned}
&\quad + 2 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp}}^{\substack{z,c,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-2}} \left\| \sum_{y} \frac{\beta_{y,D_{\mathsf{G}}}^{z,c}}{\sqrt{2^{n+k_1}}} |\gamma_1[0^{n+k_1},\perp]\rangle \right\|^2 \\
&\overset{(c)}{=} 2 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-2}} \sum_{[y_1]_{k_1}=[s]_{k_1}} \left\| \frac{\beta_{y,D_{\mathsf{G}}}^{z,c}}{\sqrt{2^{n+k_1}}} |z,c,y\oplus[s\oplus y_1]^n, D_{\mathsf{G}}\cup(r,y_1)\rangle \right\|^2 \\
&\quad + 2 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-2}} \sum_{[y_1]_{k_1}=[s]_{k_1}} \left\| \frac{\beta_{y,D_{\mathsf{G}}}^{z,c}}{\sqrt{2^{n+k_1}}} |z,c,y\oplus\perp, D_{\mathsf{G}}\cup(r,y_1)\rangle \right\|^2 \\
&= 4 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-2}} \frac{2^n \left|\beta_{y,D_{\mathsf{G}}}^{z,c}\right|^2}{2^{n+k_1}} = 4\cdot 2^{-k_1} \left\||\Phi_2\rangle\right\|^2 .
\end{aligned}
\tag{32}
$$

Here (a) uses Lemma 10, (b) follows from the definition of $|\gamma_1[w,\nu]\rangle_{y,D_{\mathsf{G}}}^{z,c}$ as shown in Eq. (31) and the fact that S_r is a unitary operation. (c) also follows from the definition of $|\gamma_1[w,\nu]\rangle_{y,D_{\mathsf{G}}}^{z,c}$. Additionally, similar to the above computation process we can get:

$$
\begin{aligned}
&\left\|(\mathrm{U}_{\mathsf{qDec}}^1 - \mathrm{U}_{\mathsf{qDec}}^2)|\Phi_4\rangle|0\rangle_{\mathsf{ST}}\right\|^2 \\
&= \left\| \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp,u=0^{n+k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-3}} \frac{\beta_{y,D_{\mathsf{G}},u}^{z,c}}{\sqrt{2^{n+k_1}}} \left[|\gamma_1[0^{n+k_1},\perp]\rangle - |\gamma_1[0^{n+k_1},[s\oplus y_1]^n]\rangle \right] \right\|^2 \\
&\overset{(d)}{\leq} 2 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp,u=0^{n+k_1}}}^{\substack{z,c,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-3}} \left\| \sum_{y} \frac{\beta_{y,D_{\mathsf{G}},u}^{z,c}}{\sqrt{2^{n+k_1}}} |\gamma_1[0^{n+k_1},\perp]\rangle \right\|^2 \\
&\quad + 2 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp,u=0^{n+k_1}}}^{\substack{z,c,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-3}} \left\| \sum_{y} \frac{\beta_{y,D_{\mathsf{G}},u}^{z,c}}{\sqrt{2^{n+k_1}}} |\gamma_1[0^{n+k_1},[s\oplus y_1]^n]\rangle \right\|^2 \\
&\overset{(e)}{=} 2 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp,u=0^{n+k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-3}} \sum_{[y_1]_{k_1}=[s]_{k_1}} \left\| \frac{\beta_{y,D_{\mathsf{G}},u}^{z,c}}{\sqrt{2^{n+k_1}}} |z,c,y\oplus\perp, D_{\mathsf{G}}\cup(r,y_1)\rangle \right\|^2
\end{aligned}
$$

$$
\begin{aligned}
&+2 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp \\ u=0^{n+k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-3}} \sum_{[y_1]_{k_1}=[s]_{k_1}} \left\| \frac{\beta^{z,c}_{y,D_{\mathsf{G}},u}}{\sqrt{2^{n+k_1}}} |z,c,y\oplus[s\oplus y_1]^n, D_{\mathsf{G}}\cup(r,y_1)\rangle \right\|^2 \\
=&4 \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp,u=0^{n+k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-3}} \frac{2^n \left|\beta^{z,c}_{y,D_{\mathsf{G}},u}\right|^2}{2^{n+k_1}} = 4\cdot 2^{-k_1} \left\| |\Phi_4\rangle \right\|^2 .
\end{aligned} \tag{33}
$$

Here (d) uses Lemma 10 and the definition of $|\gamma_1[w,\nu]\rangle^{z,c}_{y,D_{\mathsf{G}}}$ as shown in Eq. (31), (e) also follows from the definition of $|\gamma_1[w,\nu]\rangle^{z,c}_{y,D_{\mathsf{G}}}$.

As for $\|(\mathrm{U}^1_{\mathsf{qDec}}-\mathrm{U}^2_{\mathsf{qDec}})|\Phi_3\rangle|0\rangle_{\mathsf{ST}}\|$, we have

$$
\begin{aligned}
&\left\|(\mathrm{U}^1_{\mathsf{qDec}}-\mathrm{U}^2_{\mathsf{qDec}})|\Phi_3\rangle|0\rangle_{\mathsf{ST}}\right\|^2 \\
=&\left\| \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp \\ u\in\{0,1\}^{n+k_1},u\neq 0^{n+k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-3}} \frac{\beta^{z,c}_{y,D_{\mathsf{G}},u}}{2^{n+k_1}} \begin{bmatrix} |\gamma_2[u,[s\oplus y_1]^n]\rangle - |\gamma_2[u,\perp]\rangle \\ -|\gamma_3[u,[s\oplus y_1]^n]\rangle + |\gamma_3[u,\perp]\rangle \end{bmatrix} \right\|^2 \\
\overset{(f)}{\le}& \frac{4}{2^{k_1}} \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp \\ [y_1]_{k_1}=[s]_{k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-3}} \left\| \sum_{u\neq 0^{n+k_1}}^{u\in\{0,1\}^{n+k_1}} \frac{(-1)^{y_1\cdot u}\beta^{z,c,y}_{D_{\mathsf{G}},u}}{\sqrt{2^{n+k_1}}} |z,c,y\oplus[s\oplus y_1]^n, D_{\mathsf{G}}\rangle \right\|^2 \\
&+\frac{4}{2^{k_1}} \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp \\ [y_1]_{k_1}=[s]_{k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-3}} \left\| \sum_{u\neq 0^{n+k_1}}^{u\in\{0,1\}^{n+k_1}} \frac{(-1)^{y_1\cdot u}\beta^{z,c,y}_{D_{\mathsf{G}},u}}{\sqrt{2^{n+k_1}}} |z,c,y\oplus\perp, D_{\mathsf{G}}\rangle \right\|^2 \\
&+\frac{4}{2^{k_1}} \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp \\ [y_1]_{k_1}=[s]_{k_1} \\ y_2\in\{0,1\}^{n+k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-3}} \left\| \sum_{u\neq 0^{n+k_1}}^{u\in\{0,1\}^{n+k_1}} \frac{(-1)^{y_1\cdot u}\beta^{z,c}_{y,D_{\mathsf{G}},u}}{2^{n+k_1}} \left| \begin{matrix} z,c,y\oplus[s\oplus y_1]^n \\ D_{\mathsf{G}}\cup(r,y_2) \end{matrix} \right\rangle \right\|^2 \\
&+\frac{4}{2^{k_1}} \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp \\ [y_1]_{k_1}=[s]_{k_1} \\ y_2\in\{0,1\}^{n+k_1}}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}-3}} \left\| \sum_{u\neq 0^{n+k_1}}^{u\in\{0,1\}^{n+k_1}} \frac{(-1)^{y_1\cdot u}\beta^{z,c}_{y,D_{\mathsf{G}},u}}{2^{n+k_1}} |z,c,y\oplus\perp, D_{\mathsf{G}}\cup(r,y_2)\rangle \right\|^2
\end{aligned}
$$

$$= \frac{16}{2^{k_1}} \sum_{\substack{f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ r:=t\oplus\mathsf{H}(s),D_{\mathsf{G}}(r)=\perp}}^{\substack{z,c,y,D_{\mathsf{G}} \\ n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-3}} \sum_{[y_1]_{k_1}=[s]_{k_1}} \left| \sum_{u\in\{0,1\}^{n+k_1},u\neq 0^{n+k_1}} \frac{(-1)^{y_1\cdot u}\beta_{y,D_{\mathsf{G}},u}^{z,c}}{\sqrt{2^{n+k_1}}} \right|^2 \tag{34}$$

$$\overset{(g)}{\leq} 16\cdot 2^{-k_1}\left\||\Phi_3\rangle\right\|^2.$$

Here (f) firstly uses Lemma 10 and then follows the definition of $|\gamma_2[w,\nu]\rangle_{y,D_{\mathsf{G}}}^{z,c}$ and $|\gamma_3[w,\nu]\rangle_{y,D_{\mathsf{G}}}^{z,c}$ as shown in Eq. (31), while (g) follows from the definition of $|\Phi_3\rangle$.

Thus, by combining Eq. (32), Eq. (33) and Eq. (34), we finally obtain

$$\begin{aligned}\left\|(\mathrm{U}_{\mathsf{qDec}}^1-\mathrm{U}_{\mathsf{qDec}}^2)|\Phi\rangle|0\rangle_{\mathsf{ST}}\right\| &\overset{(h)}{\leq} \sum_{i=1}^4 \left\|(\mathrm{U}_{\mathsf{qDec}}^1-\mathrm{U}_{\mathsf{qDec}}^2)|\Phi_i\rangle|0\rangle_{\mathsf{ST}}\right\| \\ &\leq 2\cdot\sqrt{2^{-k_1}}(\||\Phi_2\rangle\|+\||\Phi_4\rangle\|)+4\cdot\sqrt{2^{-k_1}}\||\Phi_3\rangle\| \\ &\overset{(i)}{\leq} 8\cdot\sqrt{2^{-k_1}}\||\Phi\rangle\| = 8\cdot\sqrt{2^{-k_1}}.\end{aligned}$$

Here (h) uses the triangle inequality, and (i) comes from the fact that $\||\Phi_i\rangle\| \leq \||\Phi\rangle\|$ for $i\in[4]$. Recall that $\mathrm{U}_{\mathsf{qDec}}^{1,*} = \mathrm{U}_\perp\circ\mathrm{P}_{c^*}+\mathrm{U}_{\mathsf{qDec}}^1\circ(\mathrm{I}-\mathrm{P}_{c^*})$ and $\mathrm{U}_{\mathsf{qDec}}^{2,*} = \mathrm{U}_\perp\circ\mathrm{P}_{c^*}+\mathrm{U}_{\mathsf{qDec}}^2\circ(\mathrm{I}-\mathrm{P}_{c^*})$, thus it is easy to check that

$$\begin{aligned}\left\|(\mathrm{U}_{\mathsf{qDec}}^{1,*}-\mathrm{U}_{\mathsf{qDec}}^{2,*})|\Phi\rangle|0\rangle_{\mathsf{ST}}\right\| &= \left\|(\mathrm{U}_{\mathsf{qDec}}^1-\mathrm{U}_{\mathsf{qDec}}^2)\circ(\mathrm{I}-\mathrm{P}_{c^*})|\Phi\rangle|0\rangle_{\mathsf{ST}}\right\| \\ &\leq 8\cdot\sqrt{2^{-k_1}}\|(\mathrm{I}-\mathrm{P}_{c^*})|\Phi\rangle\| \leq 8\cdot\sqrt{2^{-k_1}}.\end{aligned}$$

Now for any unit joint state $|\Phi\rangle$ on the registers $\mathsf{Z},\mathsf{I},\mathsf{O}$ and $\mathsf{D}_{q_{\mathsf{G}}}$ just before one decryption query, we have

$$\max\left\{\|(\mathrm{U}_{\mathsf{qDec}}^1-\mathrm{U}_{\mathsf{qDec}}^2)|\Phi\rangle|0\rangle_{\mathsf{ST}}\|, \|(\mathrm{U}_{\mathsf{qDec}}^{1,*}-\mathrm{U}_{\mathsf{qDec}}^{2,*})|\Phi\rangle|0\rangle_{\mathsf{ST}}\|\right\} \leq 8\cdot\sqrt{2^{-k_1}}.$$

Then by the hybrid argument and the definition of $|\Psi_1\rangle$ and $|\Psi_2\rangle$ given in Eq. (29), it is straightforward to obtain that $\||\Psi_1\rangle|0\rangle_{\mathsf{ST}}-|\Psi_2\rangle|0\rangle_{\mathsf{ST}}\| \leq 8q_{\mathsf{D}}\cdot\sqrt{2^{-k_1}}$.

D.3 Proof of Lemma 2

Similar to the proof of Lemma 1, the key point of this proof is to prove that the difference between the operations $\mathrm{U}_{\mathsf{qDec}}^3/\mathrm{U}_{\mathsf{qDec}}^{3,*}$ and $\mathrm{U}_{\mathsf{qDec}}^4/\mathrm{U}_{\mathsf{qDec}}^{4,*}$ can be upper bounded by some parameters. In more detail, for any unit joint state $|\Phi\rangle$ on the registers $\mathsf{Z},\mathsf{I},\mathsf{O},\mathsf{D}_{q_{\mathsf{G}}}$ and $\mathsf{D}_{q_{\mathsf{H}}}$ just before one decryption query, we need to bound the following value:

$$\max\left\{\|(\mathrm{U}_{\mathsf{qDec}}^3-\mathrm{U}_{\mathsf{qDec}}^4)|\Phi\rangle|0\rangle_{\mathsf{ST}}\|, \|(\mathrm{U}_{\mathsf{qDec}}^{3,*}-\mathrm{U}_{\mathsf{qDec}}^{4,*})|\Phi\rangle|0\rangle_{\mathsf{ST}}\|\right\}.$$

In the following proof, we first give an upper bound of $\|(\mathrm{U}_{\mathsf{qDec}}^3-\mathrm{U}_{\mathsf{qDec}}^4)|\Phi\rangle|0\rangle_{\mathsf{ST}}\|$.

Similar to Supplementary Material D.2, we divide $|\Phi\rangle$ into the following four complete and mutually orthogonal parts:

$$|\Phi_1\rangle := \sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2\\ f_{\mathsf{sk}}^{-1}(c)=\perp}} \beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\rangle,$$

$$|\Phi_2\rangle := \sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp}} \beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\rangle,$$

$$|\Phi_3\rangle := \sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-3\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp\\ u\in\{0,1\}^{k_0},u\neq 0^{k_0}}} \frac{\beta^{z,c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}} \sum_{y_1\in\{0,1\}^{k_0}} (-1)^{y_1\cdot u}|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\cup(s,y_1)\rangle\rangle,$$

$$|\Phi_4\rangle := \sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-3\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp,u=0^{k_0}}} \frac{\beta^{z,c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}} \sum_{y_1\in\{0,1\}^{k_0}} |z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\cup(s,y_1)\rangle\rangle,$$

and additionally introduce a complement of $\mathrm{U}^3_{\mathsf{qDec}}$ for any $z,c,y,D_{\mathsf{G}},D_{\mathsf{H}},s$ such that $D_{\mathsf{H}}(s)=\perp$ and $n(D_{\mathsf{H}})\le q_{\mathsf{H}}-1$ as:

$$\mathrm{U}^3_{\mathsf{qDec}}\frac{1}{\sqrt{2^{k_0}}}\sum_{y_1\in\{0,1\}^{k_0}}|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\cup(s,y_1)\rangle|0\rangle_{\mathsf{ST}} := \frac{1}{\sqrt{2^{k_0}}}\sum_{y_1\in\{0,1\}^{k_0}}|z,c,y\oplus\perp,D_{\mathsf{G}},D_{\mathsf{H}}\cup(s,y_1)\rangle|0\rangle_{\mathsf{ST}}.$$

For the fixed tuple $(z,c,y,D_{\mathsf{G}},D_{\mathsf{H}})$ with $f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp$ and $D_{\mathsf{H}}(s)=\perp$, we define the following states that will be used later. Here $w\in\{0,1\}^{k_0}$, $\nu\in\{0,1\}^n\cup\{\perp\}$, and $\mathrm{L}^t_{D_{\mathsf{G}}}:=\{y\in\{0,1\}^{k_0}: D_{\mathsf{G}}(t\oplus y)\neq\perp\}$.

$$\begin{aligned}
|\gamma_1[w,\nu]\rangle^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}} &:= \sum_{\substack{y_1\in\mathrm{L}^t_{D_{\mathsf{G}}}\\ [D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}} (-1)^{y_1\cdot w}|z,c,y\oplus\nu,D_{\mathsf{G}},D_{\mathsf{H}}\cup(s,y_1)\rangle\rangle,\\
|\gamma_2[w,\nu]\rangle^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}} &:= \sum_{\substack{y_1\in\mathrm{L}^t_{D_{\mathsf{G}}}\\ [D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}} (-1)^{y_1\cdot w}|z,c,y\oplus\nu,D_{\mathsf{G}},D_{\mathsf{H}}\rangle,\\
|\gamma_3[w,\nu]\rangle^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}} &:= \sum_{\substack{y_1\in\mathrm{L}^t_{D_{\mathsf{G}}},y_2\in\{0,1\}^{k_0}\\ [D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}} \frac{(-1)^{y_1\cdot w}}{\sqrt{2^{k_0}}}|z,c,y\oplus\nu,D_{\mathsf{G}},D_{\mathsf{H}}\cup(s,y_2)\rangle\rangle.
\end{aligned} \tag{35}$$

Let S_x be an abbreviation of the local decompression procedure $\mathsf{StdDecomp}_x$. Then by the quantum simulation of unitary operation $\mathrm{U}^3_{\mathsf{qDec}}$ (resp. $\mathrm{U}^4_{\mathsf{qDec}}$) given in Eq. (8) (resp. Eq. (10)), we have $\|(\mathrm{U}^3_{\mathsf{qDec}} - \mathrm{U}^4_{\mathsf{qDec}})|\Phi_1\rangle|0\rangle_{\mathsf{ST}}\| = 0$ and

$$
\begin{aligned}
&\left\|\left(\mathrm{U}^3_{\mathsf{qDec}} - \mathrm{U}^4_{\mathsf{qDec}}\right)|\Phi_2\rangle|0\rangle_{\mathsf{ST}}\right\|^2 \\
=&\left\|\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}, n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ D_{\mathsf{H}}(s)=\perp}} \frac{\beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}\mathsf{S}_s\left[|\gamma_1[0^{k_0},[s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n]\rangle - |\gamma_1[0^{k_0},\perp]\rangle\right]\right\|^2 \\
\overset{(a)}{\le}& 2\left\|\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}, n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ D_{\mathsf{H}}(s)=\perp}} \frac{\beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}\mathsf{S}_s|\gamma_1[0^{k_0},[s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n]\rangle\right\|^2 \\
&\qquad + 2\left\|\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}, n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ D_{\mathsf{H}}(s)=\perp}} \frac{\beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}\mathsf{S}_s|\gamma_1[0^{k_0},\perp]\rangle\right\|^2 \\
\overset{(b)}{=}& 2\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}} \\ n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ D_{\mathsf{H}}(s)=\perp, y_1\in \mathrm{L}^t_{D_{\mathsf{G}}} \\ [D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}} \left\|\frac{\beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}|z,c,y\oplus[s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n, D_{\mathsf{G}}, D_{\mathsf{H}}\cup(s,y_1)\rangle\right\|^2 \\
&\qquad + 2\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}, n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ D_{\mathsf{H}}(s)=\perp, y_1\in \mathrm{L}^t_{D_{\mathsf{G}}} \\ [D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}} \left\|\frac{\beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}|z,c,y\oplus\perp, D_{\mathsf{G}}, D_{\mathsf{H}}\cup(s,y_1)\rangle\right\|^2 \\
=&4\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}, n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ D_{\mathsf{H}}(s)=\perp}} \frac{\left|\left\{y_1 : y_1\in \mathrm{L}^t_{D_{\mathsf{G}}} \wedge [D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}\right\}\right|\cdot|\beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}|^2}{2^{k_0}} \\
\overset{(c)}{\le}& \frac{4q_{\mathsf{G}}}{2^{k_0}}\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}} \\ n(D_{\mathsf{G}})\le q_{\mathsf{G}}, n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2 \\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp \\ D_{\mathsf{H}}(s)=\perp}} |\beta^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}|^2 = 4q_{\mathsf{G}}\cdot 2^{-k_0}\left\||\Phi_2\rangle\right\|^2.
\end{aligned}
\tag{36}
$$

Here (a) uses Lemma 10, (b) follows from the definition of $|\gamma_1[w,\nu]\rangle^{z,c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}$ as shown in Eq. (35) and the fact that S_r is a unitary operation. (c) follows the fact that $|\mathrm{L}^t_{D_{\mathsf{G}}}| \le q_{\mathsf{G}}$. Additionally, similar to the computation process of the above

formula we can get

$$\left\|\left(\mathrm{U}^3_{\mathsf{qDec}}-\mathrm{U}^4_{\mathsf{qDec}}\right)|\Phi_4\rangle|0\rangle_{\mathsf{ST}}\right\|^2 \le 4q_{\mathsf{G}}\cdot 2^{-k_0}\left\||\Phi_4\rangle\right\|^2. \tag{37}$$

As for $\|(\mathrm{U}^3_{\mathsf{qDec}}-\mathrm{U}^4_{\mathsf{qDec}})|\Phi_3\rangle|0\rangle_{\mathsf{ST}}\|$, we have

$$\begin{aligned}
&\left\|\left(\mathrm{U}^3_{\mathsf{qDec}}-\mathrm{U}^4_{\mathsf{qDec}}\right)|\Phi_3\rangle|0\rangle_{\mathsf{ST}}\right\|^2\\
=&\left\|\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp\\ u\in\{0,1\}^{k_0},u\neq 0^{k_0}}}\frac{\beta^{z,c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{k_0}}\begin{bmatrix}|\gamma_2[u,[s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n]\rangle-|\gamma_2[u,\perp]\rangle\\ -|\gamma_3[u,[s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n]\rangle+|\gamma_3[u,\perp]\rangle\end{bmatrix}\right\|^2\\
\overset{(d)}{\le}&\frac{4q_{\mathsf{G}}}{2^{k_0}}\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp\\ y_1\in \mathrm{L}^t_{D_{\mathsf{G}}},[D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}}\left\|\sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}}\frac{(-1)^{y_1\cdot u}\beta^{z,c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}\left|\begin{matrix}z,c,\\ y\oplus[s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n,\\ D_{\mathsf{G}},D_{\mathsf{H}}\end{matrix}\right\rangle\right\|^2\\
&+\frac{4q_{\mathsf{G}}}{2^{k_0}}\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp\\ y_1\in \mathrm{L}^t_{D_{\mathsf{G}}},[D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}}\left\|\sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}}\frac{(-1)^{y_1\cdot u}\beta^{z,c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}|z,c,y\oplus\perp,D_{\mathsf{G}},D_{\mathsf{H}}\rangle\right\|^2\\
&+\frac{4q_{\mathsf{G}}}{2^{k_0}}\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp,y_2\in\{0,1\}^{k_0}\\ y_1\in \mathrm{L}^t_{D_{\mathsf{G}}},[D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}}\left\|\sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}}\frac{(-1)^{y_1\cdot u}\beta^{z,c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{k_0}}\left|\begin{matrix}z,c,\\ y\oplus[s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n\\ D_{\mathsf{G}},D_{\mathsf{H}}\cup(s,y_2)\end{matrix}\right\rangle\right\|^2\\
&+\frac{4q_{\mathsf{G}}}{2^{k_0}}\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp,y_2\in\{0,1\}^{k_0}\\ y_1\in \mathrm{L}^t_{D_{\mathsf{G}}},[D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}}\left\|\sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}}\frac{(-1)^{y_1\cdot u}\beta^{z,c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{k_0}}\left|\begin{matrix}z,c,y\oplus\perp\\ D_{\mathsf{G}},D_{\mathsf{H}}\cup(s,y_2)\end{matrix}\right\rangle\right\|^2\\
=&\frac{16q_{\mathsf{G}}}{2^{k_0}}\sum_{\substack{z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ n(D_{\mathsf{G}})\le q_{\mathsf{G}},n(D_{\mathsf{H}})\le q_{\mathsf{H}}-2\\ f_{\mathsf{sk}}^{-1}(c)=(s,t)\neq\perp\\ D_{\mathsf{H}}(s)=\perp}}\sum_{y_1\in \mathrm{L}^t_{D_{\mathsf{G}}},[D_{\mathsf{G}}(t\oplus y_1)]_{k_1}=[s]_{k_1}}\left|\sum_{u\in\{0,1\}^{k_0},u\neq 0^{k_0}}\frac{\beta^{z,c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}\right|^2\\
\overset{(e)}{\le}&16q_{\mathsf{G}}\cdot 2^{-k_0}\left\||\Phi_3\rangle\right\|^2.
\end{aligned}\tag{38}$$

Here (d) firstly uses Lemma 10 and the property $n(D_{\mathsf{G}})\le q_{\mathsf{G}}$, and then follows the definition of $|\gamma_2[w,\nu]\rangle^{z,c}_{y,D_{\mathsf{G}}}$ and $|\gamma_3[w,\nu]\rangle^{z,c}_{y,D_{\mathsf{G}}}$ as shown in Eq. (35), while (e) follows the definition of $|\Phi_3\rangle$.

Thus, by combining Eq. (36), Eq. (37) and Eq. (38), we finally obtain

$$\begin{aligned}\left\|\left(\mathrm{U}^3_{\mathsf{qDec}}-\mathrm{U}^4_{\mathsf{qDec}}\right)|\Phi\rangle|0\rangle_{\mathsf{ST}}\right\| &\overset{(f)}{\leq} \sum_{i=1}^{4}\left\|\left(\mathrm{U}^3_{\mathsf{qDec}}-\mathrm{U}^4_{\mathsf{qDec}}\right)|\Phi_i\rangle|0\rangle_{\mathsf{ST}}\right\| \\ &\leq 2\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}(\||\Phi_2\rangle\|+2\||\Phi_3\rangle\|+\||\Phi_4\rangle\|) \\ &\overset{(g)}{\leq} 8\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}\||\Phi\rangle\| = 8\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}.\end{aligned} \tag{39}$$

Here (f) uses the triangle inequality, and (g) comes from the fact that $\||\Phi_i\rangle\| \leq \||\Phi\rangle\|$ for $i \in [4]$. Recall that $\mathrm{U}^{3,*}_{\mathsf{qDec}} = \mathrm{U}_{\perp}\circ\mathrm{P}_{c^*} + \mathrm{U}^3_{\mathsf{qDec}}\circ(\mathrm{I}-\mathrm{P}_{c^*})$ and $\mathrm{U}^{4,*}_{\mathsf{qDec}} = \mathrm{U}_{\perp}\circ\mathrm{P}_{c^*} + \mathrm{U}^4_{\mathsf{qDec}}\circ(\mathrm{I}-\mathrm{P}_{c^*})$, thus it is easy to check that

$$\begin{aligned}\left\|(\mathrm{U}^{3,*}_{\mathsf{qDec}}-\mathrm{U}^{4,*}_{\mathsf{qDec}})|\Phi\rangle|0\rangle_{\mathsf{ST}}\right\| &= \left\|(\mathrm{U}^3_{\mathsf{qDec}}-\mathrm{U}^4_{\mathsf{qDec}})\circ(\mathrm{I}-\mathrm{P}_{c^*})|\Phi\rangle|0\rangle_{\mathsf{ST}}\right\| \\ &\leq 8\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}.\end{aligned}$$

Now for any unit joint state $|\Phi\rangle$ on the registers $\mathsf{Z},\mathsf{I},\mathsf{O},\mathsf{D}_{q_{\mathsf{G}}}$ and $\mathsf{D}_{q_{\mathsf{H}}}$ just before one decryption query, we have

$$\max\left\{\|(\mathrm{U}^3_{\mathsf{qDec}}-\mathrm{U}^4_{\mathsf{qDec}})|\Phi\rangle|0\rangle_{\mathsf{ST}}\|, \|(\mathrm{U}^{3,*}_{\mathsf{qDec}}-\mathrm{U}^{4,*}_{\mathsf{qDec}})|\Phi\rangle|0\rangle_{\mathsf{ST}}\|\right\} \leq 8\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}.$$

Then, similar to the proof given in Supplementary Material D.1, we can prove that $|\Pr[1 \leftarrow \mathbf{G_3}] - \Pr[1 \leftarrow \mathbf{G_4}]| \leq 8q_{\mathsf{D}}\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}$.

D.4 Proof of Eq. (12): $\mathrm{U}^4_{\mathsf{qDec}}/\mathrm{U}^{4,*}_{\mathsf{qDec}}$ is Identical to $\mathrm{U}_{\mathsf{Search}}/\mathrm{U}^*_{\mathsf{Search}}$

For the joint state $|\Phi\rangle$ on registers $\mathsf{Z},\mathsf{I},\mathsf{O},\mathsf{D}_{q_{\mathsf{G}}}$ and $\mathsf{D}_{q_{\mathsf{H}}}$ just before one decryption query, we can denote it as a superposition of the computational basis state $|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\rangle$. Here we introduce Z to denote the ancillary quantum registers that may be used by the game. In the following, we divide $|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\rangle$ into five complete cases according to the tuple $(z,c,y,D_{\mathsf{G}},D_{\mathsf{H}})$ and show that performing $\mathrm{U}^4_{\mathsf{qDec}}$ and $\mathrm{U}_{\mathsf{Search}}$ on each case results in the same state.

1. $\boldsymbol{f^{-1}_{\mathsf{sk}}(c)=\perp}$. It is easy to obtain that $\mathrm{U}^4_{\mathsf{qDec}}$ maps $|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$ to $|z,c,y\oplus\perp,D_{\mathsf{G}},D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$. Moreover, it must hold that $\mathsf{Search}(\mathsf{pk},c,D_{\mathsf{G}},D_{\mathsf{H}}) = \perp$ otherwise it will contradict with the condition $f^{-1}_{\mathsf{sk}}(c)=\perp$, meaning that $\mathrm{U}_{\mathsf{Search}}|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}} = |z,c,y\oplus\perp,D_{\mathsf{G}},D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$.
2. $\boldsymbol{f^{-1}_{\mathsf{sk}}(c)=(s,t), D_{\mathsf{H}}(s)=\perp}$. By a direct computation we know that $\mathrm{U}^4_{\mathsf{qDec}}$ maps $|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$ to $|z,c,y\oplus\perp,D_{\mathsf{G}},D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$ as well. As for the $\mathrm{U}_{\mathsf{Search}}$, suppose $\mathsf{Search}(\mathsf{pk},c,D_{\mathsf{G}},D_{\mathsf{H}}) \neq \perp$, then there must exist $(\tilde{s},D_{\mathsf{H}}(\tilde{s}))$ and $(\tilde{r},D_{\mathsf{G}}(\tilde{r}))$ s.t. $f_{\mathsf{pk}}(\tilde{s},\tilde{r}\oplus D_{\mathsf{H}}(\tilde{s})) = c$. Set $\tilde{t}$ equal to $\tilde{r}\oplus D_{\mathsf{H}}(\tilde{s})$, then we have that $f_{\mathsf{pk}}(s,t) = f_{\mathsf{pk}}(\tilde{s},\tilde{t}) = c$ where $\tilde{s}\neq s$, which contradicts with the injective property of f_{pk}. So in this case we also have $\mathrm{U}_{\mathsf{Search}}|z,c,y,D_{\mathsf{G}},D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}} = |z,c,y\oplus\perp,D_{\mathsf{G}},D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$.

3. $\boldsymbol{f_{\mathsf{sk}}^{-1}(c) = (s,t), D_{\mathsf{H}}(s) \neq \perp, r := t \oplus D_{\mathsf{H}}(s), D_{\mathsf{G}}(r) = \perp}$. Same case 2, $\mathrm{U}^4_{\mathsf{qDec}}$ maps $|z, c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$ to $|z, c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$. As for the $\mathrm{U}_{\mathsf{Search}}$, suppose that $\mathsf{Search}(\mathsf{pk}, c, D_{\mathsf{G}}, D_{\mathsf{H}}) \neq \perp$, then there must exist $(\tilde{s}, D_{\mathsf{H}}(\tilde{s}))$ and $(\tilde{r}, D_{\mathsf{G}}(\tilde{r}))$ s.t. $f_{\mathsf{pk}}(\tilde{s}, \tilde{r} \oplus D_{\mathsf{H}}(\tilde{s})) = c$. Set $\tilde{t}$ equals to $\tilde{r} \oplus D_{\mathsf{H}}(\tilde{s})$, then we have $f_{\mathsf{pk}}(s,t) = f_{\mathsf{pk}}(\tilde{s}, \tilde{t}) = c$ where $\tilde{t} \neq t$, which contradicts with the injective property of f_{pk}. Hence in this case we also have $\mathrm{U}_{\mathsf{Search}}|z, c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}} = |z, c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$.
4. $\boldsymbol{f_{\mathsf{sk}}^{-1}(c) = (s,t),\ D_{\mathsf{H}}(s) \neq \perp,\ r := t \oplus D_{\mathsf{H}}(s),\ D_{\mathsf{G}}(r) \neq \perp, [s]_{k_1} \neq [D_{\mathsf{G}}(r)]_{k_1}}$. It is obvious that $\mathrm{U}^4_{\mathsf{qDec}}|z, c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}} = |z, c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$. As for $\mathrm{U}_{\mathsf{Search}}$, suppose $\mathsf{Search}(\mathsf{pk}, c, D_{\mathsf{G}}, D_{\mathsf{H}}) \neq \perp$, then there must exist $(\tilde{s}, D_{\mathsf{H}}(\tilde{s}))$ and $(\tilde{r}, D_{\mathsf{G}}(\tilde{r}))$ s.t. $f_{\mathsf{pk}}(\tilde{s}, \tilde{r} \oplus D_{\mathsf{H}}(\tilde{s})) = c$ and $[\tilde{s}]_{k_1} = [D_{\mathsf{G}}(\tilde{r})]_{k_1}$. Set $\tilde{t}$ equals to $\tilde{r} \oplus D_{\mathsf{H}}(\tilde{s})$, then we have $f_{\mathsf{pk}}(s,t) = f_{\mathsf{pk}}(\tilde{s}, \tilde{t}) = c$. By the injective property of f_{pk} we know that $(s,t) = (\tilde{s}, \tilde{t})$, thus $D_{\mathsf{H}}(s) = D_{\mathsf{H}}(\tilde{s})$, $D_{\mathsf{G}}(r) = D_{\mathsf{G}}(\tilde{r})$, and $[s]_{k_1} = [\tilde{s}]_{k_1} = [D_{\mathsf{G}}(\tilde{r})]_{k_1} = [D_{\mathsf{G}}(r)]_{k_1}$, which contradicts the condition $[s]_{k_1} \neq [D_{\mathsf{G}}(r)]_{k_1}$. Thus we have $\mathrm{U}_{\mathsf{Search}}|z, c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}} = |z, c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$.
5. $\boldsymbol{f_{\mathsf{sk}}^{-1}(c) = (s,t),\ D_{\mathsf{H}}(s) \neq \perp,\ r := t \oplus D_{\mathsf{H}}(s),\ D_{\mathsf{G}}(r) \neq \perp, [s]_{k_1} = [D_{\mathsf{G}}(r)]_{k_1}}$. In this case both $\mathrm{U}^4_{\mathsf{qDec}}$ and $\mathrm{U}_{\mathsf{Search}}$ map $|z, c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$ to $|z, c, y \oplus [s \oplus D_{\mathsf{G}}(r)]^n, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle|0\rangle_{\mathsf{ST}}$.

According to the above analysis, we have $\mathrm{U}^4_{\mathsf{qDec}}|\Phi\rangle|0\rangle_{\mathsf{ST}} = \mathrm{U}_{\mathsf{Search}}|\Phi\rangle|0\rangle_{\mathsf{ST}}$. Recall that $\mathrm{U}^{4,*}_{\mathsf{qDec}} = \mathrm{U}_{\perp} \circ \mathrm{P}_{c^*} + \mathrm{U}^4_{\mathsf{qDec}} \circ (\mathrm{I} - \mathrm{P}_{c^*})$ and $\mathrm{U}^*_{\mathsf{Search}} = \mathrm{U}_{\perp} \circ \mathrm{P}_{c^*} + \mathrm{U}_{\mathsf{Search}} \circ (\mathrm{I} - \mathrm{P}_{c^*})$, thus we also have $\mathrm{U}^{4,*}_{\mathsf{qDec}}|\Phi\rangle|0\rangle_{\mathsf{ST}} = \mathrm{U}^*_{\mathsf{Search}}|\Phi\rangle|0\rangle_{\mathsf{ST}}$.

D.5 Proof of Lemma 3

Here we first compute the upper bound of $\Pr[\mathsf{Find}_{r^*} : \mathbf{G_6}]$. In fact, based on the r^* uniformly chosen from $\{0,1\}^{k_0}$, we can directly construct a quantum oracle algorithm $\mathcal{B}^{\mathcal{O}^{SC}_{\{r^*\}}}(\mathsf{pk})$ that runs $\mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}}(\mathsf{pk})$ in game $\mathbf{G_6}$ to generate (m_0^*, m_1^*). During this running, $\mathcal{B}$ will reply to $\mathcal{A}$'s query to qDec and H by performing $\mathrm{U}_{\mathsf{Search}}$ and $\mathsf{CStO}_{\mathsf{H}}$, respectively; meanwhile, every time $\mathcal{A}$ queries to G, $\mathcal{B}$ will first query its oracle $\mathcal{O}^{SC}_{\{r^*\}}$ and then perform $\mathsf{CStO}_{\mathsf{G}}$. Now it is evident that

$$\Pr[\mathsf{Find}_{r^*} : \mathbf{G_6}] = \Pr[\mathsf{Find}_{r^*} : \mathcal{B}^{\mathcal{O}^{SC}_{\{r^*\}}}(\mathsf{pk})].$$

Here $\mathcal{B}$ makes at most q_{G} queries to its oracle $\mathcal{O}^{SC}_{\{r^*\}}$, and its input pk is independent of the set $\{r^*\}$. Hence by using Corollary 1 and the fact that r^* is uniformly chosen from $\{0,1\}^{k_0}$, we have

$$\Pr[\mathsf{Find}_{r^*} : \mathcal{B}^{\mathcal{O}^{SC}_{\{r^*\}}}(\mathsf{pk})] \leq 4q_{\mathsf{G}} \cdot \max_{x \in \{0,1\}^{k_0}} \Pr[x \in \{r^*\}] = 4q_{\mathsf{G}} \cdot 2^{-k_0}.$$

That is to say, $\Pr[\mathsf{Find}_{r^*} : \mathbf{G_6}] \leq 4q_{\mathsf{G}} \cdot 2^{-k_0}$. Then, by a similar analysis we also have $\Pr[\mathsf{Find}_{s^*} : \mathbf{G_9}] \leq 4q_{\mathsf{H}} \cdot 2^{-n-k_1}$. This proves Lemma 3.

D.6 Proof of Lemma 4

We first consider the upper bound of $\|[\mathrm{U}^*_{\mathsf{Search}}, \mathsf{StdDecomp}_{r^*}]\|$. Recall that $\mathrm{U}^*_{\mathsf{Search}}$ acts on registers $\mathsf{I}, \mathsf{O}, \mathsf{D}_{q_\mathsf{G}}$ and $\mathsf{D}_{q_\mathsf{H}}$, and $\mathsf{StdDecomp}_{r^*}$ acts on register $\mathsf{D}_{q_\mathsf{G}}$. Here I/O represents the input/output register of the decryption oracle. $\mathsf{D}_{q_\mathsf{G}}$ and $\mathsf{D}_{q_\mathsf{H}}$ are the database registers used to simulate G and H, respectively.

For a unit joint state on the registers $\mathsf{I}, \mathsf{O}, \mathsf{D}_{q_\mathsf{G}}$ and $\mathsf{D}_{q_\mathsf{H}}$, we can denote it as:

$$|\Phi\rangle := \sum_{c,y,D_\mathsf{G},D_\mathsf{H}} \alpha^{c,y}_{D_\mathsf{G},D_\mathsf{H}} |c, y, D_\mathsf{G}, D_\mathsf{H}\rangle.$$

Here similar to Supplementary Material D.2, we omit the detailed set notations for c, y, D_G and D_H, and abbreviate $|c, y\rangle_{\mathsf{IO}} |D_\mathsf{G}\rangle_{\mathsf{D}_{q_\mathsf{G}}} |D_\mathsf{H}\rangle_{\mathsf{D}_{q_\mathsf{H}}}$ into $|c, y, D_\mathsf{G}, D_\mathsf{H}\rangle$. Next, we divide $|\Phi\rangle$ into the following four complete and mutually orthogonal parts:

$$|\Phi_1\rangle := \sum_{\substack{c,y,D_\mathsf{G},D_\mathsf{H} \\ c=c^* \vee (c\neq c^* \wedge \nexists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_\mathsf{H}(s))=c)}} \beta^{c,y}_{D_\mathsf{G},D_\mathsf{H}} |c, y, D_\mathsf{G}, D_\mathsf{H}\rangle,$$

$$|\Phi_2\rangle := \sum_{\substack{c,y,D_\mathsf{G},D_\mathsf{H},c\neq c^* \\ \exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_\mathsf{H}(s))=c \\ D_\mathsf{G}(r^*)=\perp}} \beta^{c,y}_{D_\mathsf{G},D_\mathsf{H}} |c, y, D_\mathsf{G}, D_\mathsf{H}\rangle,$$

$$|\Phi_3\rangle := \sum_{\substack{c,y,D_\mathsf{G},D_\mathsf{H},c\neq c^* \\ \exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_\mathsf{H}(s))=c \\ D_\mathsf{G}(r^*)=\perp, n(D_\mathsf{G})\leq q_\mathsf{G}-1 \\ u\in\{0,1\}^{n+k_1}, u\neq 0^{n+k_1}}} \frac{\beta^{c,y,u}_{D_\mathsf{G},D_\mathsf{H}}}{\sqrt{2^{n+k_1}}} \sum_{y_1\in\{0,1\}^{n+k_1}} (-1)^{y_1\cdot u} |c, y, D_\mathsf{G} \cup (r^*, y_1), D_\mathsf{H}\rangle,$$

$$|\Phi_4\rangle := \sum_{\substack{c,y,D_\mathsf{G},D_\mathsf{H},c\neq c^* \\ \exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_\mathsf{H}(s))=c \\ D_\mathsf{G}(r^*)=\perp, n(D_\mathsf{G})\leq q_\mathsf{G}-1, u=0^{n+k_1}}} \frac{\beta^{c,y,u}_{D_\mathsf{G},D_\mathsf{H}}}{\sqrt{2^{n+k_1}}} \sum_{y_1\in\{0,1\}^{n+k_1}} |c, y, D_\mathsf{G} \cup (r^*, y_1), D_\mathsf{H}\rangle.$$

We emphasize that the notation "$f_{\mathsf{pk}}(s, r^* \oplus D_\mathsf{H}(s))$" assumes that $D_\mathsf{H}(s) \neq \perp$. In addition, since f_{pk} is an injective function, there exists at most one $(s, D_\mathsf{H}(s))$ pair s.t. $f_{\mathsf{pk}}(s, r^* \oplus D_\mathsf{H}(s)) = c$ for a fixed c. This property will be used by our following proof, and we will not specifically point it out for simplicity.

For the fixed tuple $(c, y, D_\mathsf{G}, D_\mathsf{H})$ where $c \neq c^*$, $\exists\, s$ s.t. $f_{\mathsf{pk}}(s, r^* \oplus D_\mathsf{H}(s)) = c$ and $D_\mathsf{G}(r^*) = \perp$, we define the following states that will be used later. Here $w \in \{0,1\}^{n+k_1}$ and $\nu \in \{0,1\}^n \cup \{\perp\}$.

$$\begin{aligned}
|\gamma_1[w,\nu]\rangle^{c,y}_{D_\mathsf{G},D_\mathsf{H}} &:= \sum_{[y_1]_{k_1}=[s]_{k_1}} (-1)^{y_1\cdot w} |c, y\oplus\nu, D_\mathsf{G} \cup (r^*, y_1), D_\mathsf{H}\rangle, \\
|\gamma_2[w,\nu]\rangle^{c,y}_{D_\mathsf{G},D_\mathsf{H}} &:= \sum_{[y_1]_{k_1}=[s]_{k_1}} (-1)^{y_1\cdot w} |c, y\oplus\nu, D_\mathsf{G}, D_\mathsf{H}\rangle, \\
|\gamma_3[w,\nu]\rangle^{c,y}_{D_\mathsf{G},D_\mathsf{H}} &:= \sum_{[y_1]_{k_1}=[s]_{k_1}}^{y_2\in\{0,1\}^{n+k_1}} \frac{(-1)^{y_1\cdot w}}{\sqrt{2^{n+k_1}}} |c, y\oplus\nu, D_\mathsf{G} \cup (r^*, y_2), D_\mathsf{H}\rangle.
\end{aligned} \tag{40}$$

Let S_{r^*} be an abbreviation of the local decompression procedure $\mathsf{StdDecomp}_{r^*}$. Then by the definition of $\mathrm{U}^*_{\mathsf{Search}}$ introduced in game $\mathbf{G_5}$ and the definition of S_{r^*} given in Section 2.4, we have $\|(\mathrm{U}^*_{\mathsf{Search}} \circ \mathsf{S}_{r^*} - \mathsf{S}_{r^*} \circ \mathrm{U}^*_{\mathsf{Search}})|\Phi_1\rangle\| = 0$ and

$$
\begin{aligned}
&\|(\mathrm{U}^*_{\mathsf{Search}} \circ \mathsf{S}_{r^*} - \mathsf{S}_{r^*} \circ \mathrm{U}^*_{\mathsf{Search}})|\Phi_2\rangle\|^2 \\
=&\left\| \sum_{\substack{\exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) = c \\ D_{\mathsf{G}}(r^*) = \perp}}^{c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, c \neq c^*} \frac{\beta^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}} \left[|\gamma_1[0^{n+k_1}, [s \oplus y_1]^n]\rangle - |\gamma_1[0^{n+k_1}, \perp]\rangle \right] \right\|^2 \\
\overset{(a)}{\leq} 2 &\left\| \sum_{\substack{\exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) = c \\ D_{\mathsf{G}}(r^*) = \perp}}^{c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, c \neq c^*} \frac{\beta^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}} |\gamma_1[0^{n+k_1}, [s \oplus y_1]^n]\rangle \right\|^2 \\
&+ 2 \left\| \sum_{\substack{\exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) = c \\ D_{\mathsf{G}}(r^*) = \perp}}^{c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, c \neq c^*} \frac{\beta^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}} |\gamma_1[0^{n+k_1}, \perp]\rangle \right\|^2 \\
\overset{(b)}{=} 2 &\sum_{\substack{\exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) = c \\ D_{\mathsf{G}}(r^*) = \perp}}^{c, D_{\mathsf{G}}, D_{\mathsf{H}}, c \neq c^*} \left\| \sum_{y} \frac{\beta^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}} |\gamma_1[0^{n+k_1}, [s \oplus y_1]^n]\rangle \right\|^2 \\
&+ 2 \sum_{\substack{\exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) = c \\ D_{\mathsf{G}}(r^*) = \perp}}^{c, D_{\mathsf{G}}, D_{\mathsf{H}}, c \neq c^*} \left\| \sum_{y} \frac{\beta^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}} |\gamma_1[0^{n+k_1}, \perp]\rangle \right\|^2 \\
\overset{(c)}{=} 2 &\sum_{\substack{\exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) = c \\ D_{\mathsf{G}}(r^*) = \perp}}^{c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, c \neq c^*} \sum_{[y_1]_{k_1} = [s]_{k_1}} \left\| \frac{\beta^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}} |c, y \oplus [s \oplus y_1]^n, D_{\mathsf{G}} \cup (r^*, y_1), D_{\mathsf{H}}\rangle \right\|^2 \\
&+ 2 \sum_{\substack{\exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) = c \\ D_{\mathsf{G}}(r^*) = \perp}}^{c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, c \neq c^*} \sum_{[y_1]_{k_1} = [s]_{k_1}} \left\| \frac{\beta^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}} |c, y \oplus \perp, D_{\mathsf{G}} \cup (r^*, y_1), D_{\mathsf{H}}\rangle \right\|^2 \\
=&4 \sum_{\substack{\exists\, s \text{ s.t. } f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) = c \\ D_{\mathsf{G}}(r^*) = \perp}}^{c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, c \neq c^*} \frac{2^n \left| \beta^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}} \right|^2}{2^{n+k_1}} = 4 \cdot 2^{-k_1} \||\Phi_2\rangle\|^2 .
\end{aligned}
\tag{41}
$$

Here (a) uses Lemma 10, (b) and (c) both follow the definition of $|\gamma_1[w, \nu]\rangle^{c,y}_{D_{\mathsf{G}}, D_{\mathsf{H}}}$ as shown in Eq. (40). Additionally, similar to the above computation, we can compute

$$
\|(\mathrm{U}^*_{\mathsf{Search}} \circ \mathsf{S}_{r^*} - \mathsf{S}_{r^*} \circ \mathrm{U}^*_{\mathsf{Search}})|\Phi_4\rangle\|^2 \leq 4 \cdot 2^{-k_1} \||\Phi_4\rangle\|^2 . \tag{42}
$$

As for $\|(\mathsf{U}^*_{\mathsf{Search}}\circ\mathsf{S}_{r^*}-\mathsf{S}_{r^*}\circ\mathsf{U}^*_{\mathsf{Search}})|\Phi_3\rangle\|$, with the help of the useful property of the local decompression procedure as shown in Eq. (1), we can compute

$$
\begin{aligned}
&\|(\mathsf{U}^*_{\mathsf{Search}}\circ\mathsf{S}_{r^*}-\mathsf{S}_{r^*}\circ\mathsf{U}^*_{\mathsf{Search}})|\Phi_3\rangle\|^2\\
=&\left\|\sum_{\substack{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*\\ \exists\, s \text{ s.t. } f_{\mathsf{pk}}(s,r^*\oplus D_{\mathsf{H}}(s))=c\\ D_{\mathsf{G}}(r^*)=\perp,n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-1\\ u\in\{0,1\}^{n+k_1},u\neq 0^{n+k_1}}}\frac{\beta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{n+k_1}}\begin{bmatrix}|\gamma_3[u,[s\oplus y_1]^n]\rangle+|\gamma_3[u,\perp]\rangle\\ -|\gamma_2[u,[s\oplus y_1]^n]\rangle-|\gamma_2[u,\perp]\rangle\end{bmatrix}\right\|^2\\
\overset{(d)}{\leq}&\frac{4}{2^{k_1}}\sum_{\substack{c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ c\neq c^*,\exists\, s \text{ s.t.}\\ f_{\mathsf{pk}}(s,r^*\oplus D_{\mathsf{H}}(s))=c\\ D_{\mathsf{G}}(r^*)=\perp,n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-1\\ [y_1]_{k_1}=[s]_{k_1}}}\left\|\sum_{u\in\{0,1\}^{n+k_1}}\frac{(-1)^{y_1\cdot u}\beta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}}\left|\begin{matrix}c,y\oplus[s\oplus y_1]^n\\ D_{\mathsf{G}},D_{\mathsf{H}}\end{matrix}\right\rangle\right\|^2\\
&+\frac{4}{2^{k_1}}\sum_{\substack{c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ c\neq c^*,\exists\, s \text{ s.t.}\\ f_{\mathsf{pk}}(s,r^*\oplus D_{\mathsf{H}}(s))=c\\ D_{\mathsf{G}}(r^*)=\perp,n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-1\\ [y_1]_{k_1}=[s]_{k_1}}}\left\|\sum_{u\in\{0,1\}^{n+k_1}}\frac{(-1)^{y_1\cdot u}\beta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}}|c,y\oplus\perp,D_{\mathsf{G}},D_{\mathsf{H}}\rangle\right\|^2\\
&+\frac{4}{2^{k_1}}\sum_{\substack{c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ c\neq c^*,\exists\, s \text{ s.t.}\\ f_{\mathsf{pk}}(s,r^*\oplus D_{\mathsf{H}}(s))=c\\ D_{\mathsf{G}}(r^*)=\perp,n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-1\\ [y_1]_{k_1}=[s]_{k_1}\\ y_2\in\{0,1\}^{n+k_1}}}\left\|\sum_{u\in\{0,1\}^{n+k_1}}\frac{(-1)^{y_1\cdot u}\beta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{n+k_1}}\left|\begin{matrix}c,y\oplus[s\oplus y_1]^n\\ D_{\mathsf{G}}\cup(r^*,y_2),D_{\mathsf{H}}\end{matrix}\right\rangle\right\|^2\\
&+\frac{4}{2^{k_1}}\sum_{\substack{c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ c\neq c^*,\exists\, s \text{ s.t.}\\ f_{\mathsf{pk}}(s,r^*\oplus D_{\mathsf{H}}(s))=c\\ D_{\mathsf{G}}(r^*)=\perp,n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-1\\ [y_1]_{k_1}=[s]_{k_1}\\ y_2\in\{0,1\}^{n+k_1}}}\left\|\sum_{u\in\{0,1\}^{n+k_1}}\frac{(-1)^{y_1\cdot u}\beta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{n+k_1}}\left|\begin{matrix}c,y\oplus\perp\\ D_{\mathsf{G}}\cup(r^*,y_2),D_{\mathsf{H}}\end{matrix}\right\rangle\right\|^2\\
=&\frac{16}{2^{k_1}}\sum_{\substack{c,y,D_{\mathsf{G}},D_{\mathsf{H}}\\ c\neq c^*,\exists\, s \text{ s.t.}\\ f_{\mathsf{pk}}(s,r^*\oplus D_{\mathsf{H}}(s))=c\\ D_{\mathsf{G}}(r^*)=\perp,n(D_{\mathsf{G}})\leq q_{\mathsf{G}}-1}}\sum_{[y_1]_{k_1}=[s]_{k_1}}\left|\sum_{u\in\{0,1\}^{n+k_1}}\frac{(-1)^{y_1\cdot u}\beta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{n+k_1}}}\right|^2\\
\overset{(e)}{\leq}&16\cdot 2^{-k_1}\||\Phi_3\rangle\|^2.
\end{aligned}
\tag{43}
$$

Here (d) firstly uses Lemma 10 and then follows the definition of $|\gamma_2[w,\nu]\rangle^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}$ and $|\gamma_3[w,\nu]\rangle^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}$ as shown in Eq. (40). (e) follows from the definition of $|\Phi_3\rangle$.

Thus, by combining Eq. (41), Eq. (42) and Eq. (43), for any unit joint state $|\Phi\rangle$, we finally obtain

$$\begin{aligned}\|(\mathrm{U}^*_{\mathsf{Search}}\circ\mathsf{S}_{r^*}-\mathsf{S}_{r^*}\circ\mathrm{U}^*_{\mathsf{Search}})|\Phi\rangle\| &\overset{(f)}{\leq}\sum_{i=1}^{4}\|(\mathrm{U}^*_{\mathsf{Search}}\circ\mathsf{S}_{r^*}-\mathsf{S}_{r^*}\circ\mathrm{U}^*_{\mathsf{Search}})|\Phi_i\rangle\|\\ &\leq 2\cdot\sqrt{2^{-k_1}}(\||\Phi_2\rangle\|+\||\Phi_4\rangle\|+2\||\Phi_3\rangle\|)\\ &\overset{(g)}{\leq} 8\cdot\sqrt{2^{-k_1}}\||\Phi\rangle\|=8\cdot\sqrt{2^{-k_1}}.\end{aligned}$$

Here (f) uses the triangle inequality, and (g) comes from the fact that $\||\Phi_i\rangle\| \leq \||\Phi\rangle\|$ for $i\in[4]$. Now we prove that $\|[\mathrm{U}^*_{\mathsf{Search}},\mathsf{StdDecomp}_{r^*}]\|\leq 8\cdot\sqrt{2^{-k_1}}$.

The upper bound of $\|[\mathrm{U}^*_{\mathsf{Search}},\mathsf{StdDecomp}_{s^*}]\|$ can be deduced using an analogous method. First, we give the unit joint state $|\Phi\rangle$ on the registers $\mathsf{I},\mathsf{O},\mathsf{D}_{q_\mathsf{G}}$ and $\mathsf{D}_{q_\mathsf{H}}$ a new decompose as:

$$|\Phi_1'\rangle := \sum_{\substack{c,y,D_\mathsf{G},D_\mathsf{H},c=c^*\\ \vee(c\neq c^*\wedge\nexists\, t\,\text{s.t.}\, f_{\mathsf{pk}}(s^*,t)=c)\\ \vee(c\neq c^*\wedge\exists\, t\,\text{s.t.}\, f_{\mathsf{pk}}(s^*,t)=c\\ \wedge\nexists\, r\,\text{s.t.}\,[D_\mathsf{G}(r)]_{k_1}=[s^*]_{k_1})}} \eta^{c,y}_{D_\mathsf{G},D_\mathsf{H}}|c,y,D_\mathsf{G},D_\mathsf{H}\rangle,$$

$$|\Phi_2'\rangle := \sum_{\substack{c,y,D_\mathsf{G},D_\mathsf{H},c\neq c^*\\ \exists\, t\,\text{s.t.}\, f_{\mathsf{pk}}(s^*,t)=c, D_\mathsf{H}(s^*)=\perp\\ \exists\, r\,\text{s.t.}\,[D_\mathsf{G}(r)]_{k_1}=[s^*]_{k_1}}} \eta^{c,y}_{D_\mathsf{G},D_\mathsf{H}}|c,y,D_\mathsf{G},D_\mathsf{H}\rangle,$$

$$|\Phi_3'\rangle := \sum_{\substack{c,y,D_\mathsf{G},D_\mathsf{H},c\neq c^*\\ \exists\, t\ \text{s.t.}\ f_{\mathsf{pk}}(s^*,t)=c\\ D_\mathsf{H}(s^*)=\perp, n(D_\mathsf{H})\leq q_\mathsf{H}-1\\ \exists\, r\ \text{s.t.}\ [D_\mathsf{G}(r)]_{k_1}=[s^*]_{k_1}\\ u\in\{0,1\}^{k_0}, u\neq 0^{k_0}}} \frac{\eta^{c,y,u}_{D_\mathsf{G},D_\mathsf{H}}}{\sqrt{2^{k_0}}}\sum_{y_1\in\{0,1\}^{k_0}}(-1)^{y_1\cdot u}|c,y,D_\mathsf{G},D_\mathsf{H}\cup(s^*,y_1)\rangle,$$

$$|\Phi_4'\rangle := \sum_{\substack{c,y,D_\mathsf{G},D_\mathsf{H},c\neq c^*\\ \exists\, t\,\text{s.t.}\, f_{\mathsf{pk}}(s^*,t)=c\\ D_\mathsf{H}(s^*)=\perp, n(D_\mathsf{H})\leq q_\mathsf{H}-1\\ \exists\, r\,\text{s.t.}\,[D_\mathsf{G}(r)]_{k_1}=[s^*]_{k_1}; u=0^{k_0}}} \frac{\eta^{c,y,u}_{D_\mathsf{G},D_\mathsf{H}}}{\sqrt{2^{k_0}}}\sum_{y_1\in\{0,1\}^{k_0}}|c,y,D_\mathsf{G},D_\mathsf{H}\cup(s^*,y_1)\rangle.$$

In the above formulas and during the following proof, we emphasize that the notation "$[D_\mathsf{G}(r)]_{k_1}$" implicitly assumes $D_\mathsf{G}(r)\neq\perp$.

For the fixed tuple $(c,y,D_\mathsf{G},D_\mathsf{H})$ where $c\neq c^*,\exists\, t$ s.t. $f_{\mathsf{pk}}(s^*,t)=c$, $D_\mathsf{H}(s^*)=\perp$, and $\exists\, r$ s.t. $[D_\mathsf{G}(r)]_{k_1}=[s^*]_{k_1}$, we define the following states that will be used later.

$$
\begin{aligned}
|\gamma_1'[w,\nu]\rangle^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}} &:= \sum_{y_1\in \mathrm{L}^c_{D_{\mathsf{G}}}} (-1)^{y_1\cdot w}|c, y\oplus\nu, D_{\mathsf{G}}, D_{\mathsf{H}}\cup(s^*,y_1)\rangle,\\
|\gamma_2'[w,\nu]\rangle^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}} &:= \sum_{y_1\in \mathrm{L}^c_{D_{\mathsf{G}}}} (-1)^{y_1\cdot w}|c, y\oplus\nu, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle,\\
|\gamma_3'[w,\nu]\rangle^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}} &:= \sum_{y_1\in \mathrm{L}^c_{D_{\mathsf{G}}},y_2\in\{0,1\}^{k_0}} \frac{(-1)^{y_1\cdot w}}{\sqrt{2^{k_0}}}|c, y\oplus\nu, D_{\mathsf{G}}, D_{\mathsf{H}}\cup(s^*,y_2)\rangle.
\end{aligned}
\tag{44}
$$

Here $w\in\{0,1\}^{k_0}$, $\nu\in\{0,1\}^n\cup\{\perp\}$, and $\mathrm{L}^c_{D_{\mathsf{G}}} := \{y\in\{0,1\}^{k_0} : \exists r \text{ s.t. } D_{\mathsf{G}}(r)\neq\perp\wedge f_{\mathsf{pk}}(s^*, r\oplus y) = c\wedge[s^*]_{k_1} = [D_{\mathsf{G}}(r)]_{k_1}\}$. Notice that for any c, by the injective property of f_{pk} we can conclude that there exists at most one t s.t. $f_{\mathsf{pk}}(s^*,t)=c$. Further, based on this t, it is evident that $\mathrm{L}^c_{D_{\mathsf{G}}}\subseteq\{r\oplus t : r \text{ s.t. } D_{\mathsf{G}}(r)\neq\perp\}$, which means that $|\mathrm{L}^c_{D_{\mathsf{G}}}|\le q_{\mathsf{G}}$. These two facts will be used by our following proof, and we will not specifically point them out for simplicity.

Let S_{s^*} be an abbreviation of the local decompression procedure $\mathsf{StdDecomp}_{s^*}$. Then by the definition of $\mathrm{U}^*_{\mathsf{Search}}$ introduced in game $\mathbf{G_5}$ and the definition of S_{s^*} given in Section 2.4, we have $\|(\mathrm{U}^*_{\mathsf{Search}}\circ\mathsf{S}_{s^*}-\mathsf{S}_{s^*}\circ\mathrm{U}^*_{\mathsf{Search}})|\Phi_1'\rangle\| = 0$ and

$$
\begin{aligned}
&\|(\mathrm{U}^*_{\mathsf{Search}}\circ\mathsf{S}_{s^*}-\mathsf{S}_{s^*}\circ\mathrm{U}^*_{\mathsf{Search}})|\Phi_2'\rangle\|^2\\
=&\left\|\sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c, D_{\mathsf{H}}(s^*)=\perp\\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1}}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} \frac{\eta^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}\left[|\gamma_1'[0^{k_0},[s^*\oplus D_{\mathsf{G}}(t\oplus y_1)]^n]\rangle - |\gamma_1'[0^{k_0},\perp]\rangle\right]\right\|^2\\
\overset{(h)}{\le}&2\left\|\sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c, D_{\mathsf{H}}(s^*)=\perp\\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1}}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} \frac{\eta^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}|\gamma_1[0^{k_0},[s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n]\rangle\right\|^2\\
&\qquad+2\left\|\sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c, D_{\mathsf{H}}(s^*)=\perp\\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1}}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} \frac{\eta^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}|\gamma_1[0^{k_0},\perp]\rangle\right\|^2\\
\overset{(i)}{=}&2\sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c\\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1}\\ D_{\mathsf{H}}(s^*)=\perp, y_1\in\mathrm{L}^c_{D_{\mathsf{G}}}}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*}\left\|\frac{\eta^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}|c, y\oplus[s^*\oplus D_{\mathsf{G}}(t\oplus y_1)]^n, D_{\mathsf{G}}, D_{\mathsf{H}}\cup(s^*,y_1)\rangle\right\|^2\\
&\qquad+2\sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c\\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1}\\ D_{\mathsf{H}}(s^*)=\perp, y_1\in\mathrm{L}^c_{D_{\mathsf{G}}}}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*}\left\|\frac{\eta^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}}|c, y\oplus\perp, D_{\mathsf{G}}, D_{\mathsf{H}}\cup(s^*,y_1)\rangle\right\|^2
\end{aligned}
$$

$$
\begin{aligned}
=&4 \sum_{\substack{\exists\, t \text{ s.t.} f_{\mathsf{pk}}(s^*,t)=c, D_{\mathsf{H}}(s^*)=\perp \\ \exists\, r \text{ s.t.} [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1}}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} \frac{|\mathrm{L}^c_{D_{\mathsf{G}}}| \cdot |\eta^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}|^2}{2^{k_0}} \\
\leq& \frac{4q_{\mathsf{G}}}{2^{k_0}} \sum_{\substack{\exists\, t \text{ s.t.} f_{\mathsf{pk}}(s^*,t)=c, D_{\mathsf{H}}(s^*)=\perp \\ \exists\, r \text{ s.t.} [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1}}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} |\eta^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}|^2 = 4q_{\mathsf{G}} \cdot 2^{-k_0} \left\||\Phi'_2\rangle\right\|^2 .
\end{aligned}
\tag{45}
$$

Here (h) uses Lemma 10, (i) follows from the definition of $|\gamma'_1[w,\nu]\rangle^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}$ shown in Eq. (44). Additionally, similar to the above computation, we can compute

$$
\left\|(\mathrm{U}^*_{\mathsf{Search}} \circ \mathsf{S}_{s^*} - \mathsf{S}_{s^*} \circ \mathrm{U}^*_{\mathsf{Search}})|\Phi'_4\rangle\right\|^2 \leq 4q_{\mathsf{G}} \cdot 2^{-k_0} \left\||\Phi'_4\rangle\right\|^2 . \tag{46}
$$

As for $\|(\mathrm{U}^*_{\mathsf{Search}} \circ \mathsf{S}_{s^*} - \mathsf{S}_{s^*} \circ \mathrm{U}^*_{\mathsf{Search}})|\Phi'_3\rangle\|$, with the help of the useful property of local decompression procedure as shown in Eq. (1), we can compute

$$
\begin{aligned}
&\left\|(\mathrm{U}^*_{\mathsf{Search}} \circ \mathsf{S}_{s^*} - \mathsf{S}_{s^*} \circ \mathrm{U}^*_{\mathsf{Search}})|\Phi'_3\rangle\right\|^2 \\
=&\left\| \sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c \\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1} \\ D_{\mathsf{H}}(s^*)=\perp, n(D_{\mathsf{H}})\leq q_{\mathsf{H}}-1}}^{\substack{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^* \\ u\in\{0,1\}^{k_0}, u\neq 0^{k_0}}} \frac{\eta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{k_0}} \begin{bmatrix} |\gamma'_2[u,\perp]\rangle - |\gamma'_2[u,[s^*\oplus D_{\mathsf{G}}(t\oplus y_1)]^n\rangle \\ -|\gamma'_3[u,\perp]\rangle + |\gamma'_3[u,[s^*\oplus D_{\mathsf{G}}(t\oplus y_1)]^n]\rangle \end{bmatrix} \right\|^2 \\
\overset{(j)}{\leq}& \frac{4q_{\mathsf{G}}}{2^{k_0}} \sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c, y_1\in \mathrm{L}^c_{D_{\mathsf{G}}} \\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1} \\ D_{\mathsf{H}}(s^*)=\perp, n(D_{\mathsf{H}})\leq q_{\mathsf{H}}-1}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} \left\| \sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}} \frac{(-1)^{y_1\cdot u}\eta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}} \left| \begin{matrix} c, y\oplus [s^*\oplus D_{\mathsf{G}}(t\oplus y_1)]^n \\ D_{\mathsf{G}}, D_{\mathsf{H}} \end{matrix} \right\rangle \right\|^2 \\
&+ \frac{4q_{\mathsf{G}}}{2^{k_0}} \sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c, y_1\in \mathrm{L}^c_{D_{\mathsf{G}}} \\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1} \\ D_{\mathsf{H}}(s^*)=\perp, n(D_{\mathsf{H}})\leq q_{\mathsf{H}}-1}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} \left\| \sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}} \frac{(-1)^{y_1\cdot u}\eta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}} |c, y\oplus\perp, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle \right\|^2 \\
&+ \frac{4q_{\mathsf{G}}}{2^{k_0}} \sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c, y_1\in \mathrm{L}^c_{D_{\mathsf{G}}} \\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1} \\ D_{\mathsf{H}}(s^*)=\perp, n(D_{\mathsf{H}})\leq q_{\mathsf{H}}-1}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} \left\| \sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}} \frac{(-1)^{y_1\cdot u}\eta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{k_0}} \left| \begin{matrix} c, y\oplus [s\oplus D_{\mathsf{G}}(t\oplus y_1)]^n \\ D_{\mathsf{G}}, D_{\mathsf{H}}\cup (s^*, y_2) \end{matrix} \right\rangle \right\|^2 \\
&+ \frac{4q_{\mathsf{G}}}{2^{k_0}} \sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c, y_1\in \mathrm{L}^c_{D_{\mathsf{G}}} \\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1} \\ D_{\mathsf{H}}(s^*)=\perp, n(D_{\mathsf{H}})\leq q_{\mathsf{H}}-1}}^{\substack{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^* \\ u\in\{0,1\}^{k_0}, u\neq 0^{k_0}}} \left\| \sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}} \frac{(-1)^{y_1\cdot u}\eta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{2^{k_0}} \left| \begin{matrix} c, y\oplus\perp \\ D_{\mathsf{G}}, D_{\mathsf{H}}\cup (s^*, y_2) \end{matrix} \right\rangle \right\|^2 \\
=&\frac{16q_{\mathsf{G}}}{2^{k_0}} \sum_{\substack{\exists\, t \text{ s.t. } f_{\mathsf{pk}}(s^*,t)=c \\ \exists\, r \text{ s.t. } [D_{\mathsf{G}}(r)]_{k_1}=[s^*]_{k_1} \\ D_{\mathsf{H}}(s^*)=\perp, n(D_{\mathsf{H}})\leq q_{\mathsf{H}}-1}}^{c,y,D_{\mathsf{G}},D_{\mathsf{H}},c\neq c^*} \sum_{y_1\in \mathrm{L}^c_{D_{\mathsf{G}}}} \left| \sum_{u\neq 0^{k_0}}^{u\in\{0,1\}^{k_0}} \frac{\eta^{c,y,u}_{D_{\mathsf{G}},D_{\mathsf{H}}}}{\sqrt{2^{k_0}}} \right|^2 \overset{(k)}{\leq} 16q_{\mathsf{G}} \cdot 2^{-k_0} \left\||\Phi'_3\rangle\right\|^2 .
\end{aligned}
\tag{47}
$$

Here (j) first uses Lemma 10 and then follows the definition of $|\gamma_2'[w,\nu]\rangle^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}$ and $|\gamma_3'[w,\nu]\rangle^{c,y}_{D_{\mathsf{G}},D_{\mathsf{H}}}$ as shown in Eq. (44), while (k) follows from the definition of $|\Phi_3'\rangle$. Thus, by combining Eq. (45), (46) and (47), for any unit joint state $|\Phi\rangle$, we finally obtain

$$\begin{aligned}\|(\mathrm{U}^*_{\mathsf{Search}}\circ\mathsf{S}_{s^*}-\mathsf{S}_{s^*}\circ\mathrm{U}^*_{\mathsf{Search}})|\Phi\rangle\| &\overset{(l)}{\leq}\sum_{i=1}^{4}\|(\mathrm{U}^*_{\mathsf{Search}}\circ\mathsf{S}_{s^*}-\mathsf{S}_{s^*}\circ\mathrm{U}^*_{\mathsf{Search}})|\Phi_i'\rangle\|\\ &\leq 2\cdot\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}(\||\Phi_2'\rangle\|+\||\Phi_4'\rangle\|+2\||\Phi_3'\rangle\|)\\ &\overset{(m)}{\leq} 8\cdot\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}\||\Phi\rangle\|=8\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}.\end{aligned}$$

Here (l) uses the triangle inequality, and (m) comes from the fact that $\||\Phi_i'\rangle\|\leq\||\Phi\rangle\|$ for $i\in[4]$. Now we prove that $\|[\mathrm{U}^*_{\mathsf{Search}},\mathsf{StdDecomp}_{s^*}]\|\leq 8\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}$.

D.7 Proof of Lemma 5

As shown in Fig. 8, game $\mathbf{G_{13}}$ is the same as game $\mathbf{G_{12}}$, except that the performing of $\mathsf{StdDecomp}_{r^*}$(resp. $\mathsf{StdDecomp}_{s^*}$) just after the performing of $\mathrm{U}^{r^*}_{g^*}$ (resp. $\mathrm{U}^{s^*}_{h^*}$) is canceled.

In game $\mathbf{G_{12}}$, we have the following four observations about $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$:

1. $\mathsf{StdDecomp}_{r^*}$ must commute with $\mathsf{StdDecomp}_{s^*}$ and $\mathrm{U}^{s^*}_{h^*}$ since they act on different database registers, meanwhile $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ both commute with the adversary's operations since the adversary cannot perform operations on the database registers.
2. $\mathsf{StdDecomp}_{r^*}$ must commute with the oracle queries to G performed during the generation of b', since $\mathsf{CStO}^{r^*,g^*}_{\mathsf{G}}$ used to simulate G does not depend on the database state at point r^*.
3. $\mathsf{StdDecomp}_{s^*}$ must commute with the oracle queries to H performed during the generation of b', since $\mathsf{CStO}^{s^*,h^*}_{\mathsf{H}}$ used to simulate H does not depend on the database state at point s^*.
4. $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ must commute with the oracle queries to H and G performed during the generation of b', respectively, since they act on different registers.

In addition, as shown in Lemma 4, $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ both almost commute with the queries to qDec performed during the generation of b' in game $\mathbf{G_{12}}$. Note that qDec is queried at most q_{D} times, hence with an error of $8q_{\mathsf{D}}\cdot\sqrt{2^{-k_1}}+8q_{\mathsf{D}}\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}$, we can move $\mathsf{StdDecomp}_{r^*}$ and $\mathsf{StdDecomp}_{s^*}$ to the end of the game and then remove them without affecting the adversary's output. That is to say, our game $\mathbf{G_{13}}$ given above actually satisfies that

$$|\Pr[1\leftarrow\mathbf{G_{12}}]-\Pr[1\leftarrow\mathbf{G_{13}}]|\leq 8q_{\mathsf{D}}\cdot\sqrt{2^{-k_1}}+8q_{\mathsf{D}}\sqrt{q_{\mathsf{G}}}\cdot\sqrt{2^{-k_0}}.$$

D.8 Proof of Lemma 6

We first introduce the Fixed-Permutation One-Way to Hiding theorem which will be used in this proof. In fact, the O2H theorems introduced in Section 2.3 take the quantum accessible random function H as a unitary operation that maps $|x, y\rangle$ into $|x, y \oplus \mathsf{H}(x)\rangle$. In [22], the quantum accessible permutation P, which is implemented by a more concise unitary operation $|x\rangle \mapsto |\mathsf{P}(x)\rangle$[23], is considered. Based on this unitary operation tailored to the permutation, [22] proved the following new O2H theorem:

Theorem 6 (Fixed-Permutation O2H [22, Theorem 1]). *Let z be a random bitstring. Let P_0, $\mathsf{P}_1 : \mathcal{X} \to \mathcal{X}$ be two fixed permutations. Let $\mathcal{A}$ be a quantum oracle algorithm making at most q queries. Let*

$$P_{\text{left}} := \Pr[b = 1 : b \leftarrow \mathcal{A}^{\mathsf{P}_0}(z)],\ P_{\text{right}} := \Pr[b = 1 : b \leftarrow \mathcal{A}^{\mathsf{P}_1}(z)].$$

Here oracle P_0 (resp. P_1) is implemented as $|x\rangle \mapsto |\mathsf{P}_0(x)\rangle$ (resp. $|x\rangle \mapsto |\mathsf{P}_1(x)\rangle$).

Let $\mathcal{B}^{\mathsf{P}_1}$ be an algorithm that on input z, chooses $i \leftarrow_{\$} [q]$, runs $\mathcal{A}^{\mathsf{P}_1}$ until (just before) the i-th query, measures the query register and outputs the measurement outcome w. Let $P_{\text{guess}} := \Pr[\mathsf{P}_0(w) \neq \mathsf{P}_1(w) : w \leftarrow \mathcal{B}^{\mathsf{P}_1}(z)]$. Then

$$|P_{\text{left}} - P_{\text{right}}| \leq 2q \cdot \sqrt{P_{\text{guess}}} \text{ and } \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq 2q \cdot \sqrt{P_{\text{guess}}}.$$

Next, we present our proof. Recall that $\mathcal{C}$ denotes the ciphertext space and $\mathcal{M} = \{0,1\}^n$ denotes the message space. Define $\mathcal{M}_\perp := \mathcal{M} \cup \{\perp\} \cup \{y \oplus \perp : y \in \mathcal{M}\}$[24]. Let S_{pk} be the set of all possible pk generated by Gen. Now we define three fixed permutations on the set

$$\mathcal{C} \times \mathcal{M}_\perp \times \mathbf{D}_{q_\mathsf{G}} \times \mathbf{D}_{q_\mathsf{H}} \times S_{\mathsf{pk}} \times \{0,1\}^{k_0} \times \{0,1\}^{n+k_1} \times \{0,1\}^{n+k_1} \times \{0,1\}^{k_0} \times \mathcal{C}$$

as follows:

- Permutation $\mathsf{P}_{\mathsf{Search}}$: Transform the tuple $(c, y, D_\mathsf{G}, D_\mathsf{H}, \mathsf{pk}, r, g, s, h, c_1)$ into $(c, y \oplus \mathsf{Search}(\mathsf{pk}, c, D_\mathsf{G}, D_\mathsf{H}), D_\mathsf{G}, D_\mathsf{H}, \mathsf{pk}, r, g, s, h, c_1)$, where Search is the classical function defined in Eq. (11).
- Permutation P_G: Transform the tuple $(c, y, D_\mathsf{G}, D_\mathsf{H}, \mathsf{pk}, r, g, s, h, c_1)$ into $(c, y, D'_\mathsf{G}, D_\mathsf{H}, \mathsf{pk}, r, g, s, h, c_1)$, where $D'_\mathsf{G} = D_\mathsf{G} \cup (r, g)$ if $D_\mathsf{G}(r) = \perp$, $D'_\mathsf{G} = D_\mathsf{G} \backslash (r, g)$ if $D_\mathsf{G}(r) = g$, and $D'_\mathsf{G} = D_\mathsf{G}$ otherwise.
- Permutation P_H: Transform the tuple $(c, y, D_\mathsf{G}, D_\mathsf{H}, \mathsf{pk}, r, g, s, h, c_1)$ into $(c, y, D_\mathsf{G}, D'_\mathsf{H}, \mathsf{pk}, r, g, s, h, c_1)$, where $D'_\mathsf{H} = D_\mathsf{H} \cup (s, h)$ if $D_\mathsf{H}(r) = \perp$, $D'_\mathsf{H} = D_\mathsf{H} \backslash (s, h)$ if $D_\mathsf{H}(s) = h$, and $D'_\mathsf{H} = D_\mathsf{H}$ otherwise.

Then we define two fixed permutations P_0 and P_1 as follows:

[23] We say that this unitary operation is more concise, because it does not need an additional register to store the output value.

[24] In fact, as shown in Supplementary Material B, this set is used when we quantumly encode the set $\mathcal{M} \cup \{\perp\}$.

– Permutation $\mathsf{P_0}$: Transform the tuple $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, \mathsf{pk}, r, g, s, h, c_1)$ into the tuple $(c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}, \mathsf{pk}, r, g, s, h, c_1)$ if $c = c_1$. Otherwise, perform $\mathsf{P}_{\mathsf{Search}}$.
– Permutation $\mathsf{P_1}$: Transform the tuple $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, \mathsf{pk}, r, g, s, h, c_1)$ into the tuple $(c, y \oplus \perp, D_{\mathsf{G}}, D_{\mathsf{H}}, \mathsf{pk}, r, g, s, h, c_1)$ if $c = c_1$. Otherwise, sequentially perform $\mathsf{P_G}$, $\mathsf{P_H}$, $\mathsf{P}_{\mathsf{search}}$, $\mathsf{P_G}$, $\mathsf{P_H}$.

At this point, based on the above permutations $\mathsf{P_0}$ and $\mathsf{P_1}$, and parameters $(\mathsf{pk}, r^*, g^*, s^*, h^*, c^*)$ introduced in game $\mathbf{G_{14}}$ and game $\mathbf{G_{15}}$, we can conclude that the $\mathrm{U}^*_{\mathsf{Search}}$ utilized in game $\mathbf{G_{14}}$ must satisfy that $\mathrm{U}^*_{\mathsf{Search}}|c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle = |c', y', D'_{\mathsf{G}}, D'_{\mathsf{H}}\rangle$ if and only if

$$\mathsf{P_0}(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*) = (c', y', D'_{\mathsf{G}}, D'_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*).$$

Moreover we can conclude that the $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*} \circ \mathrm{U}^*_{\mathsf{Search}} \circ \mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}$ utilized in game $\mathbf{G_{15}}$ must satisfy $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*} \circ \mathrm{U}^*_{\mathsf{Search}} \circ \mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}|c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle = |c', y', D'_{\mathsf{G}}, D'_{\mathsf{H}}\rangle$ if and only if

$$\mathsf{P_1}(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*) = (c', y', D'_{\mathsf{G}}, D'_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*).$$

Here we design a new quantum oracle algorithm $\mathcal{A}_1$, which has oracle access to $\mathsf{P_0} : |x\rangle \mapsto |\mathsf{P_0}(x)\rangle$ and runs game $\mathbf{G_{14}}$. Specifically, when $\mathcal{A}$ queries qDec by $\sum |c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$[25] on the registers $\mathsf{IOD}_{q_{\mathsf{G}}}\mathsf{D}_{q_{\mathsf{H}}}$ during the generation of b', $\mathcal{A}_1$ first expands this state into $\sum |c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle \otimes |\mathsf{pk}, r^*, g^*, s^*, h^*, c^*\rangle$ and then forwards it to its oracle $\mathsf{P_0}$. Note that such expansion is feasible because all of the elements in tuple $(\mathsf{pk}, r^*, g^*, s^*, h^*, c^*)$ have been generated after the computation of c^*. In fact, by the two conclusions mentioned above, $\mathcal{A}_1$ perfectly simulates the procedure of $\mathcal{A}$ querying qDec during the generation of b', thus we have

$$\Pr[1 \leftarrow \mathbf{G_{14}}] = \Pr[1 \leftarrow \mathcal{A}_1^{\mathsf{P_0}}].$$

Similarly, if we change the oracle $\mathsf{P_0} : |x\rangle \mapsto |\mathsf{P_0}(x)\rangle$ to $\mathsf{P_1} : |x\rangle \mapsto |\mathsf{P_1}(x)\rangle$, we can obtain

$$\Pr[1 \leftarrow \mathbf{G_{15}}] = \Pr[1 \leftarrow \mathcal{A}_1^{\mathsf{P_1}}].$$

Assume that during the generation of b', $\mathcal{A}$ makes q'_{D} queries to qDec. Then it is obvious that $\mathcal{A}_1$ makes q'_{D} queries to oracle $\mathsf{P_0}$ or $\mathsf{P_1}$.

Now we construct game $\mathbf{G^1_{15}}$ as shown in Fig. 13, which chooses $i \leftarrow_{\$} [q'_{\mathsf{D}}]$, runs $\mathcal{A}$ in game $\mathbf{G_{15}}$ until (just before) its i-th query to qDec during the generation of b', measures the registers $\mathsf{IOD}_{q_{\mathsf{G}}}\mathsf{D}_{q_{\mathsf{H}}}$ to get the outcome $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})$, and then extends it to $w := (c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*)$. If $\mathsf{P_0}(w) \neq \mathsf{P_1}(w)$, game $\mathbf{G^1_{15}}$ outputs 1; otherwise outputs 0. Then by applying Theorem 6 and the fact that $q'_{\mathsf{D}} \leq q_{\mathsf{D}}$, we have

$$|\Pr[1 \leftarrow \mathbf{G_{14}}] - \Pr[1 \leftarrow \mathbf{G_{15}}]| \leq 2q_{\mathsf{D}}\sqrt{\Pr[1 \leftarrow \mathbf{G^1_{15}}]}. \tag{48}$$

Next, we construct games $\mathbf{G^2_{15}}$ to $\mathbf{G^9_{15}}$, which aim at providing an upper bound for the probability $\Pr[1 \leftarrow \mathbf{G^1_{15}}]$. All of these games are shown in Fig. 13.

[25] Note that $\mathcal{A}$ can only control registers IO and the coefficient of this state is omit, so this is actually an informal description. Without loss of generality, such informal (but

GAMES $\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$ to $\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$

1. $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}, r^* \leftarrow_\$ \{0,1\}^{k_0}, s^* \leftarrow_\$ \{0,1\}^{n+k_1}, t^* \leftarrow_\$ \{0,1\}^{k_0}, c^* = f_{\mathsf{pk}}(s^*,t^*)$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$
2. $(m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{G},\mathsf{H},\mathsf{qDec}}(\mathsf{pk})$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$
3. $b \leftarrow_\$ \{0,1\}$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$

   $g^* = (m_b^* || 0^{k_1}) \oplus s^*, h^* = r^* \oplus t^*$, perform $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$

4. $i \leftarrow_\$ [q'_{\mathsf{D}}]$, run $b' \leftarrow \mathcal{A}^{\mathsf{qDec},\mathsf{G},\mathsf{H}}(\mathsf{pk}, c^*)$ until i-th qDec query //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$

   measure registers $\mathsf{IOD}_{q_{\mathsf{G}}}\mathsf{D}_{q_{\mathsf{H}}}$ to get $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$

   compute $|c, y, D'_{\mathsf{G}}, D'_{\mathsf{H}}\rangle := \mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*} |c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$ //$\mathbf{G}^{\mathbf{2}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{4}}_{\mathbf{15}}$

   extend $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})$ to $w := (c, y, D_{\mathsf{G}}, D_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*)$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$

   extend $(c, y, D'_{\mathsf{G}}, D'_{\mathsf{H}})$ to $w := (c, y, D'_{\mathsf{G}}, D'_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*)$ //$\mathbf{G}^{\mathbf{2}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{4}}_{\mathbf{15}}$

5. **return** $[\![\mathsf{P}_0(w) \neq \mathsf{P}_1(w)]\!]$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{4}}_{\mathbf{15}}$

   return $[\![(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R]\!]$ //$\mathbf{G}^{\mathbf{5}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$

$\mathsf{G}(|x_{\mathsf{G}}, y_{\mathsf{G}}\rangle)$

if (m_0^*, m_1^*) has not been defined

- apply $\mathsf{CStO}_{\mathsf{G}}$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{2}}_{\mathbf{15}}$
- apply $\mathsf{CStO}_{\mathsf{G}} \circ \mathcal{O}^{SC}_{\{r^*\}}$ //$\mathbf{G}^{\mathbf{3}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{5}}_{\mathbf{15}}$
- apply $\mathsf{CStO}_{\mathsf{G}}$ //$\mathbf{G}^{\mathbf{6}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$
- **if** Find_{r^*} occurs, **return** $\perp$

else

- apply $\mathsf{CStO}^{r^*,g^*}_{\mathsf{G}}$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{6}}_{\mathbf{15}}$
- apply $\mathsf{CStO}^{r^*,g^*}_{\mathsf{G}} \circ \mathcal{O}^{SC}_{\{r^*,s^*\}}$ //$\mathbf{G}^{\mathbf{7}}_{\mathbf{15}}$
- apply $\mathsf{CStO}_{\mathsf{G}} \circ \mathcal{O}^{SC}_{\{r^*,s^*\}}$ //$\mathbf{G}^{\mathbf{8}}_{\mathbf{15}}$
- **if** Find_{r^*,s^*} occurs, **return** $\perp$
- apply $\mathsf{CStO}_{\mathsf{G}}$ //$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$

$\mathsf{H}(|x_{\mathsf{H}}, y_{\mathsf{H}}\rangle)$

if (m_0^*, m_1^*) has not been defined

- apply $\mathsf{CStO}_{\mathsf{H}}$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{3}}_{\mathbf{15}}$
- apply $\mathsf{CStO}_{\mathsf{H}} \circ \mathcal{O}^{SC}_{\{s^*\}}$ //$\mathbf{G}^{\mathbf{4}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{5}}_{\mathbf{15}}$
- apply $\mathsf{CStO}_{\mathsf{H}}$ //$\mathbf{G}^{\mathbf{6}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$
- **if** Find_{s^*} occurs, **return** $\perp$

else

- apply $\mathsf{CStO}^{s^*,h^*}_{\mathsf{H}}$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{6}}_{\mathbf{15}}$
- apply $\mathsf{CStO}^{s^*,h^*}_{\mathsf{H}} \circ \mathcal{O}^{SC}_{\{r^*,s^*\}}$ //$\mathbf{G}^{\mathbf{7}}_{\mathbf{15}}$
- apply $\mathsf{CStO}_{\mathsf{H}} \circ \mathcal{O}^{SC}_{\{r^*,s^*\}}$ //$\mathbf{G}^{\mathbf{8}}_{\mathbf{15}}$
- **if** Find_{r^*,s^*} occurs, **return** $\perp$
- apply $\mathsf{CStO}_{\mathsf{H}}$ //$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$

$\mathsf{qDec}(|c, y\rangle)$

if c^* has not been defined, apply $\mathrm{U}_{\mathsf{Search}}$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$

else apply $\mathrm{U}^{**}_{\mathsf{Search}}$ //$\mathbf{G}^{\mathbf{1}}_{\mathbf{15}}$

$\mathsf{qDec}(|c, y\rangle)$ //$\mathbf{G}^{\mathbf{2}}_{\mathbf{15}}$-$\mathbf{G}^{\mathbf{9}}_{\mathbf{15}}$

if c^* has not been defined, apply $\mathrm{U}_{\mathsf{Search}}$

else apply $\mathrm{U}^{*}_{\mathsf{Search}}$

Fig. 13. Games $\mathbf{G}^1_{15}$ to $\mathbf{G}^9_{15}$ in the proof of Lemma 6. Here different from game $\mathbf{G}_{14}$ and game $\mathbf{G}_{15}$, the games shown in this figure modifies the computation of $c^* = f_{\mathsf{pk}}(s^*, t^*)$ to be placed just after the generation of $(\mathsf{pk}, \mathsf{sk})$, which is acceptable since s^* and t^* are both uniformly sampled independent of the adversary.

Game $\mathbf{G^2_{15}}$: Compared with game $\mathbf{G^1_{15}}$, this game has the following three changes:

1. The computing of g^* and h^* and the performing of $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}$ just after the sampling of b are canceled.
2. The $\mathrm{U}^{**}_{\mathsf{Search}}$ used to simulate qDec is changed back to $\mathrm{U}^{*}_{\mathsf{Search}}$.
3. After getting the final measurement outcome $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})$, game $\mathbf{G^2_{15}}$ additionally computes $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*} |c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle = |c, y, D'_{\mathsf{G}}, D'_{\mathsf{H}}\rangle$ and then outputs $[\![\mathsf{P}_0(c, y, D'_{\mathsf{G}}, D'_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*) \neq \mathsf{P}_1(c, y, D'_{\mathsf{G}}, D'_{\mathsf{H}}, \mathsf{pk}, r^*, g^*, s^*, h^*, c^*)]\!]$.

Follow a similar analysis used to deduce Eq. (7) in the game hop from $\mathbf{G_{15}}$ to $\mathbf{G_{16}}$, the $\mathrm{U}^{r^*}_{g^*}$ and $\mathrm{U}^{s^*}_{h^*}$ additionally introduced in $\mathrm{U}^{**}_{\mathsf{Search}}$ can be canceled out, but the last performing of $\mathrm{U}^{r^*}_{g^*}$ and $\mathrm{U}^{s^*}_{h^*}$ cannot directly remove[26] and hence for consistency, the above third change additionally computes $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*} |c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$. That is to say, we have

$$\Pr[1 \leftarrow \mathbf{G^1_{15}}] = \Pr[1 \leftarrow \mathbf{G^2_{15}}]. \tag{49}$$

Game $\mathbf{G^3_{15}}$: Compared with game $\mathbf{G^2_{15}}$, this game has the following two changes:

1. Before $\mathcal{A}$ outputs (m^*_0, m^*_1), G is simulated by $\mathsf{CStO}_{\mathsf{G}} \circ \mathcal{O}^{SC}_{\{r^*\}}$, which is an operation that first queries the semi-classical oracle $\mathcal{O}^{SC}_{\{r^*\}}$ (Definition 1) on the input register of G and then performs $\mathsf{CStO}_{\mathsf{G}}$.
2. If Find_{r^*} occurs before $\mathcal{A}$ outputs (m^*_0, m^*_1), the game finally returns $\perp$. Here Find_{r^*} is the event that the measurement of $\mathcal{O}^{SC}_{\{r^*\}}$ ever has result 1.

Following a similar analysis used to deduce Eq. (14) in the game hop from $\mathbf{G_5}$ to $\mathbf{G_6}$ and use Theorem 1 with the form $|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}|$, we have

$$\begin{aligned} \left|\sqrt{\Pr[1 \leftarrow \mathbf{G^2_{15}}]} - \sqrt{\Pr[1 \leftarrow \mathbf{G^3_{15}}]}\right| &\leq \sqrt{(q_{\mathsf{G}}+1) \cdot \Pr[\mathsf{Find}_{r^*} : \mathbf{G^3_{15}}]} \\ &\overset{(a)}{\leq} 2(q_{\mathsf{G}}+1) \cdot \sqrt{2^{-k_0}}. \end{aligned} \tag{50}$$

Here (a) uses Lemma 3 and $\Pr[\mathsf{Find}_{r^*} : \mathbf{G^3_{15}}] = \Pr[\mathsf{Find}_{r^*} : \mathbf{G_6}]$. In fact, this equality is obvious since the first two steps of games $\mathbf{G^3_{15}}$ and $\mathbf{G_6}$ are identical.

Game $\mathbf{G^4_{15}}$: Compared with game $\mathbf{G^3_{15}}$, this game has the following two changes:

1. Before $\mathcal{A}$ outputs (m^*_0, m^*_1), H is simulated by $\mathsf{CStO}_{\mathsf{H}} \circ \mathcal{O}^{SC}_{\{s^*\}}$, which is an operation first queries the semi-classical oracle $\mathcal{O}^{SC}_{\{s^*\}}$ (Definition 1) on the input register of H and then performs $\mathsf{CStO}_{\mathsf{H}}$.

---

convenient) description used here is acceptable and will not cause any ambiguity in the subsequent proof.

[26] In the game hop from $\mathbf{G_{15}}$ to $\mathbf{G_{16}}$, the last $\mathrm{U}^{r^*}_{g^*}$ and $\mathrm{U}^{s^*}_{h^*}$ can be directly remove without affecting $\mathcal{A}$'s output because $\mathcal{A}$ does not measure the database registers. However, the final measurement of game $\mathbf{G^1_{15}}$ needs to measure the the database registers, which makes that we cannot directly remove the last performing of $\mathrm{U}^{r^*}_{g^*}$ and $\mathrm{U}^{s^*}_{h^*}$.

2. If Find_{s^*} occurs before $\mathcal{A}$ outputs (m_0^*, m_1^*), the game finally returns $\perp$. Here Find_{s^*} is the event that the measurement of $\mathcal{O}^{SC}_{\{s^*\}}$ ever has result 1.

Following a similar analysis used to deduce Eq. (18) in the game hop from $\mathbf{G_8}$ to $\mathbf{G_9}$ and use Theorem 1 with the form $|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}|$, we have

$$\begin{aligned}\left|\sqrt{\Pr[1 \leftarrow \mathbf{G^3_{15}}]} - \sqrt{\Pr[1 \leftarrow \mathbf{G^4_{15}}]}\right| &\leq \sqrt{(q_{\mathsf{H}}+1)\cdot \Pr[\mathsf{Find}_{s^*} : \mathbf{G^4_{15}}]} \\ &\overset{(b)}{\leq} 2(q_{\mathsf{H}}+1)\cdot\sqrt{2^{-n-k_1}}.\end{aligned} \tag{51}$$

Here (b) uses Lemma 3 and $\Pr[\mathsf{Find}_{s^*} : \mathbf{G^4_{15}}] = \Pr[\mathsf{Find}_{s^*} : \mathbf{G_9}]$. In fact, this equality is obvious since the first two steps of games $\mathbf{G^4_{15}}$ and $\mathbf{G_9}$ are identical.

In game $\mathbf{G^4_{15}}$, under the condition that Find_{r^*} and Find_{s^*} both do not occur, there will be no record of $(r^*, D_{\mathsf{G}}(r^*))$ or $(s^*, D_{\mathsf{H}}(s^*))$ in the database state before $\mathcal{A}$ outputs (m_0^*, m_1^*). Furthermore, during the generation of b', we note that the unitary operations $\mathsf{CStO}^{r^*,g^*}_{\mathsf{G}}/\mathsf{CStO}^{s^*,h^*}_{\mathsf{H}}$ used to simulate G/H will not update the database pair $(r^*, D_{\mathsf{G}}(r^*))$ or $(s^*, D_{\mathsf{H}}(s^*))$, so until the final measurement of game $\mathbf{G^4_{15}}$ there will be no record of $(r^*, D_{\mathsf{G}}(r^*) \neq \perp)$ or $(s^*, D_{\mathsf{H}}(s^*) \neq \perp)$ in the database state. That is to say, for the outcome $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})$ of the measurement on qDec's query state, we have $D_{\mathsf{G}}(r^*) = \perp$ and $D_{\mathsf{H}}(s^*) = \perp$. Then after performing $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}|c, y, D_{\mathsf{G}}, D_{\mathsf{H}}\rangle$ to get $|c, y, D'_{\mathsf{G}}, D'_{\mathsf{H}}\rangle$, one can check that

$$(c, y, D'_{\mathsf{G}}, D'_{\mathsf{H}}) = (c, y, D_{\mathsf{G}} \cup (r^*, s^* \oplus (m_b^* || 0^{k_1})), D_{\mathsf{H}} \cup (s^*, r^* \oplus t^*)).$$

As for the final judgment criteria $\mathsf{P}_0(w) \neq \mathsf{P}_1(w)$ of game $\mathbf{G^4_{15}}$, by the above observation, we can derive an implication relationship as:

$$\mathsf{P}_0(w) \neq \mathsf{P}_1(w) \implies \begin{pmatrix} (c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \text{ satisfies} \\ \exists (s, D_{\mathsf{H}}(s) \neq \perp) \text{ s.t. } c = f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) \\ \text{or } \exists (r, D_{\mathsf{G}}(r) \neq \perp) \text{ s.t. } c = f_{\mathsf{pk}}(s^*, r \oplus r^* \oplus t^*) \end{pmatrix}$$

Game $\mathbf{G^5_{15}}$: Compared with game $\mathbf{G^4_{15}}$, this game has the following two changes:

1. After the final measurement on the query state of qDec, cancel the performing of $\mathrm{U}^{s^*}_{h^*} \circ \mathrm{U}^{r^*}_{g^*}$.
2. Alter the judgment criteria for game outputting 1 from $\mathsf{P}_0(w) \neq \mathsf{P}_1(w)$ to $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R$, where

$$R := \left\{ (c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) : \begin{array}{l} \exists (s, D_{\mathsf{H}}(s) \neq \perp) \text{ s.t. } c = f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s)) \\ \text{or } \exists (r, D_{\mathsf{G}}(r) \neq \perp) \text{ s.t. } c = f_{\mathsf{pk}}(s^*, r \oplus r^* \oplus t^*) \end{array} \right\}. \tag{52}$$

According to the analysis given above, we have

$$\Pr[1 \leftarrow \mathbf{G^4_{15}}] \leq \Pr[1 \leftarrow \mathbf{G^5_{15}}]. \tag{53}$$

Game $\mathbf{G^6_{15}}$: Before $\mathcal{A}$ outputs (m_0^*, m_1^*), the queries to the semi-classical oracles $\mathcal{O}^{SC}_{\{r^*\}}$ and $\mathcal{O}^{SC}_{\{s^*\}}$ performed in game $\mathbf{G^5_{15}}$ are deleted, meanwhile the two flags

"Find_{r^*} occurs, game finally returns $\perp$" and "Find_{s^*} occurs, game finally returns $\perp$" are both removed.

In fact, the changes introduced in game $\mathbf{G}^{\mathbf{6}}_{\mathbf{15}}$ is a combination of those in games $\mathbf{G}^{\mathbf{3}}_{\mathbf{15}}$ and $\mathbf{G}^{\mathbf{4}}_{\mathbf{15}}$, but are applied in the opposite direction. Hence similar to the analysis given in game hop from $\mathbf{G}^{\mathbf{2}}_{\mathbf{15}}$ to $\mathbf{G}^{\mathbf{3}}_{\mathbf{15}}$ and $\mathbf{G}^{\mathbf{3}}_{\mathbf{15}}$ to $\mathbf{G}^{\mathbf{4}}_{\mathbf{15}}$, we have

$$\left|\sqrt{\Pr[1 \leftarrow \mathbf{G}^{\mathbf{5}}_{\mathbf{15}}]} - \sqrt{\Pr[1 \leftarrow \mathbf{G}^{\mathbf{6}}_{\mathbf{15}}]}\right| \leq 2(q_{\mathsf{G}}+1)\cdot\sqrt{2^{-k_0}} + 2(q_{\mathsf{H}}+1)\cdot\sqrt{2^{-n-k_1}}. \quad (54)$$

Game $\mathbf{G}^{\mathbf{7}}_{\mathbf{15}}$: Compared with game $\mathbf{G}^{\mathbf{6}}_{\mathbf{15}}$, this game has the following two changes:

1. during the generation of b', G is simulated by $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*} \circ \mathcal{O}^{SC}_{\{r^*,s^*\}}$, which is an operation first queries the semi-classical oracle $\mathcal{O}^{SC}_{\{r^*,s^*\}}$ (Definition 1) on the input register of G, and then performs $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$. Similarly, H is simulated by $\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*} \circ \mathcal{O}^{SC}_{\{r^*,s^*\}}$.
2. If Find_{r^*,s^*} occurs during the generation of b', the game finally returns $\perp$. Here Find_{r^*,s^*} is the event the measurement of $\mathcal{O}^{SC}_{\{r^*,s^*\}}$ ever has result 1.

We first construct a quantum oracle algorithm $\mathcal{A}_2$ as follows, which has oracle access to a constant zero function F.

– $\mathcal{A}_2$ runs game $\mathbf{G}^{\mathbf{6}}_{\mathbf{15}}$. Specifically, when $\mathcal{A}$ queries oracle G (resp. H) by a (informal) query state $\sum |x,y\rangle_{\mathsf{X_G Y_G}}$ (resp. $\sum |x,y\rangle_{\mathsf{X_H Y_H}}$) after it outputs (m_0^*, m_1^*), $\mathcal{A}_2$ will perform an additional operation: first expand this query state into $\sum |x,y\rangle \otimes_{\mathsf{X_G Y_G}} |0\rangle_{\mathsf{O_F}}$ (resp. $\sum |x,y\rangle \otimes_{\mathsf{X_H Y_H}} |0\rangle_{\mathsf{O_F}}$) and then forward the input/output register $\mathsf{X_G/O_F}$ (resp. $\mathsf{X_H/O_F}$) to its function oracle F.

Obviously $\mathcal{A}_2$ makes at most $q_{\mathsf{G}} + q_{\mathsf{H}}$ queries to its oracle, and such additional queries do not disturb $\mathcal{A}$'s computation since F is a constant zero function. Hence,

$$\Pr[1 \leftarrow \mathbf{G}^{\mathbf{6}}_{\mathbf{15}}] = \Pr\left[b = 1 : \begin{array}{l} b \leftarrow \mathcal{A}_2^{\mathsf{F}}(r^*, s^*, z), z \leftarrow D_{\mathcal{A}_2} \\ r^* \leftarrow_{\$} \{0,1\}^{k_0}, s^* \leftarrow_{\$} \{0,1\}^{n+k_1} \end{array}\right].$$

Here we particularly write r^*, s^* and assume (r^*, s^*, z) as the classical input of $\mathcal{A}_2$ without loss of generality, where $r^* \leftarrow_{\$} \{0,1\}^{k_0}$, $s^* \leftarrow_{\$} \{0,1\}^{n+k_1}$ and z is sampled according to a distribution $D_{\mathcal{A}_2}$. Then, if we change F into $\mathsf{F}\backslash\{r^*, s^*\}$, i.e., the oracle that first queries $\mathcal{O}^{SC}_{\{r^*,s^*\}}$ on the input register of F and then queries F, it is easy to check that

$$\Pr[1 \leftarrow \mathbf{G}^{\mathbf{7}}_{\mathbf{15}}] = \Pr\left[(b = 1) \wedge \neg\mathsf{Find}_{r^*,s^*} : \begin{array}{l} b \leftarrow \mathcal{A}_2^{\mathsf{F}\backslash\{r^*,s^*\}}(r^*, s^*, z), z \leftarrow D_{\mathcal{A}_2} \\ r^* \leftarrow_{\$} \{0,1\}^{k_0}, s^* \leftarrow_{\$} \{0,1\}^{n+k_1} \end{array}\right].$$

Let D be a joint distribution of $(\mathsf{E}, \mathsf{F}, \{r^*, s^*\}, z)$, where E, F are both constant zero functions, $r^* \leftarrow_{\$} \{0,1\}^{k_0}$, $s^* \leftarrow_{\$} \{0,1\}^{n+k_1}$ and $z \leftarrow D_{\mathcal{A}_2}$. Now by applying Theorem 1 with the form of $|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}|$ under this joint distribution D,

we have

$$\begin{aligned}
&\left|\sqrt{\Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{6}}]} - \sqrt{\Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{7}}]}\right| \\
&\leq \sqrt{(q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \Pr[\mathsf{Find}_{r^*,s^*} : \mathcal{A}_2^{\mathsf{F}\setminus\{r^*,s^*\}}(z), (\mathsf{E}, \mathsf{F}, \{r^*, s^*\}, z) \leftarrow D]} \\
&= \sqrt{(q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \Pr[\mathsf{Find}_{r^*,s^*} : \mathbf{G}_{\mathbf{15}}^{\mathbf{7}}]},
\end{aligned} \tag{55}$$

The above final equation follows from the definition of $\mathcal{A}_2$ and game $\mathbf{G}_{\mathbf{15}}^{\mathbf{7}}$.

Game $\mathbf{G}_{\mathbf{15}}^{\mathbf{8}}$: This game is the same as game $\mathbf{G}_{\mathbf{15}}^{\mathbf{7}}$, except that the simulation of G and H after the generation of (m_0^*, m_1^*) is changed. In fact, the $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$ is replaced by $\mathsf{CStO}_{\mathsf{G}}$, and the $\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*}$ is replaced by $\mathsf{CStO}_{\mathsf{H}}$.

Note that the situation where game $\mathbf{G}_{\mathbf{15}}^{\mathbf{7}}$ returns 1 and game $\mathbf{G}_{\mathbf{15}}^{\mathbf{8}}$ returns 1 both imply that the event Find_{r^*,s^*} does not occur, which means that $\mathcal{A}$ only queries G (resp. H) by a (informal) query state in the form $\sum_{x \neq r^*} |x, y\rangle_{\mathsf{X_G Y_G}}$ (resp. $\sum_{x \neq s^*} |x, y\rangle_{\mathsf{X_H Y_H}}$). So if Find_{r^*,s^*} does not occur, the change from game $\mathbf{G}_{\mathbf{15}}^{\mathbf{7}}$ to game $\mathbf{G}_{\mathbf{15}}^{\mathbf{8}}$ cannot be detected by the adversary. Hence, we have

$$\Pr[\mathsf{Find}_{r^*,s^*} : \mathbf{G}_{\mathbf{15}}^{\mathbf{7}}] = \Pr[\mathsf{Find}_{r^*,s^*} : \mathbf{G}_{\mathbf{15}}^{\mathbf{8}}], \tag{56}$$

and

$$\Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{7}}] = \Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{8}}]. \tag{57}$$

Game $\mathbf{G}_{\mathbf{15}}^{\mathbf{9}}$: During the generation of b', the query to the semi-classical oracle $\mathcal{O}_{\{r^*,s^*\}}^{SC}$ performed in game $\mathbf{G}_{\mathbf{15}}^{\mathbf{8}}$ is deleted, meanwhile the flag "Find_{r^*,s^*} occurs, game finally returns $\perp$" is removed.

Using a similar analysis of the game hop from $\mathbf{G}_{\mathbf{15}}^{\mathbf{6}}$ to $\mathbf{G}_{\mathbf{15}}^{\mathbf{7}}$, we have

$$\left|\sqrt{\Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{8}}]} - \sqrt{\Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{9}}]}\right| \leq \sqrt{(q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \Pr[\mathsf{Find}_{r^*,s^*} : \mathbf{G}_{\mathbf{15}}^{\mathbf{8}}]}. \tag{58}$$

Next we compute the upper bound of $\Pr[\mathsf{Find}_{r^*,s^*} : \mathbf{G}_{\mathbf{15}}^{\mathbf{8}}]$ and $\Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{9}}]$.

For the $\Pr[\mathsf{Find}_{r^*,s^*} : \mathbf{G}_{\mathbf{15}}^{\mathbf{8}}]$, we assume without loss of generality that, the query number of $\mathcal{O}_{\{r^*,s^*\}}^{SC}$ in game $\mathbf{G}_{\mathbf{15}}^{\mathbf{8}}$ is exactly $q_{\mathsf{G}} + q_{\mathsf{H}}$ (if it does not, we can modify game $\mathbf{G}_{\mathbf{15}}^{\mathbf{8}}$ by adding a number of useless queries to $\mathcal{O}_{\{r^*,s^*\}}^{SC}$ in the end). Hence we can directly construct a quantum oracle algorithm $\mathcal{A}_3(\mathsf{pk}, c^*)$, which queries $\mathcal{O}_{\{r^*,s^*\}}^{SC}$ with times $q_{\mathsf{G}} + q_{\mathsf{H}}$ and uses its input (pk, c^*) to run game $\mathbf{G}_{15}^{8}$. Now we have

$$\Pr[\mathsf{Find}_{r^*,s^*} : \mathbf{G}_{\mathbf{15}}^{\mathbf{8}}] = \Pr\left[\mathsf{Find}_{r^*,s^*} : \begin{array}{c} \mathcal{A}_3^{\mathcal{O}_{\{r^*,s^*\}}^{SC}}(\mathsf{pk}, c^*), (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen} \\ s^* \leftarrow_{\$} \{0,1\}^{n+k_1}, t^* \leftarrow_{\$} \{0,1\}^{k_0} \\ r^* \leftarrow_{\$} \{0,1\}^{k_0}, c^* = f_{\mathsf{pk}}(s^*, t^*) \end{array}\right]. \tag{59}$$

Let D be a joint distribution of $(\{r^*, s^*\}, \mathsf{pk}, c^*)$, where $r^*, t^* \leftarrow_{\$} \{0,1\}^{k_0}$, $s^* \leftarrow_{\$} \{0,1\}^{n+k_1}$, pk is sampled by $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$ and $c^* = f_{\mathsf{pk}}(s^*, t^*)$. By applying

Theorem 2 with this joint distribution D, we can construct a quantum algorithm $\mathcal{B}_1$ which first chooses $j \leftarrow_{\$} [q_{\mathsf{G}} + q_{\mathsf{H}}]$, runs $\mathcal{A}_3^{\mathcal{O}_{\emptyset}^{SC}}(\mathsf{pk}, c^*)$ until its j-th query to $\mathcal{O}_{\emptyset}^{SC}$, measures the input register to get the output x, satisfying:

$$\begin{aligned}
&\Pr\left[\mathsf{Find}_{r^*,s^*} : \begin{array}{c} \mathcal{A}_3^{\mathcal{O}_{\{r^*,s^*\}}^{SC}}(\mathsf{pk}, c^*), (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen} \\ s^* \leftarrow_{\$} \{0,1\}^{n+k_1}, t^* \leftarrow_{\$} \{0,1\}^{k_0} \\ r^* \leftarrow_{\$} \{0,1\}^{k_0}, c^* = f_{\mathsf{pk}}(s^*, t^*) \end{array}\right] \\
&\leq 4(q_{\mathsf{G}} + q_{\mathsf{H}}) \cdot \Pr[x \in \{r^*, s^*\} : x \leftarrow \mathcal{B}_1(\mathsf{pk}, c^*), (\{r^*, s^*\}, \mathsf{pk}, c^*) \leftarrow D].
\end{aligned} \tag{60}$$

Further we can compute

$$\begin{aligned}
&\Pr[x \in \{r^*, s^*\} : x \leftarrow \mathcal{B}_1(\mathsf{pk}, c^*), (\{r^*, s^*\}, \mathsf{pk}, c^*) \leftarrow D] \\
&\leq \Pr[x = r^* : x \leftarrow \mathcal{B}_1(\mathsf{pk}, c^*), (\{r^*, s^*\}, \mathsf{pk}, c^*) \leftarrow D] \\
&\quad + \Pr[x = s^* : x \leftarrow \mathcal{B}_1(\mathsf{pk}, c^*), (\{r^*, s^*\}, \mathsf{pk}, c^*) \leftarrow D] \\
&\overset{(a)}{=} 2^{-k_0} + \mathrm{Adv}_{\mathsf{TDI},\mathcal{B}_1}^{\mathsf{PD\text{-}OW}}.
\end{aligned} \tag{61}$$

Here (a) follows from two facts: first, under the joint distribution D, (pk, c^*) is independent of r^*, and the other is that, under the joint distribution D, $\mathcal{B}_1$ can be viewed as a PD-OW adversary against TDI if it aims at searching s^*. Since $\mathcal{B}_1$ needs to use the compressed standard oracle technique to simulate G and H, and uses $\mathsf{U}_{\mathsf{Search}}$, $\mathsf{U}_{\mathsf{Search}}^*$ to simulate qDec, the running time of $\mathcal{B}_1$ can be upper bounded by $\mathcal{T}_{\mathcal{A}} + \mathcal{O}(q_{\mathsf{G}}^2 + q_{\mathsf{H}}^2) + \mathcal{O}(q_{\mathsf{D}} q_{\mathsf{G}} q_{\mathsf{H}}) \cdot \mathcal{T}_f$. Now by combining Eq. (59) to Eq. (61), we obtain

$$\Pr[\mathsf{Find}_{r^*,s^*} : \mathbf{G}_{\mathbf{15}}^{\mathbf{8}}] \leq 4(q_{\mathsf{G}} + q_{\mathsf{H}}) \cdot 2^{-k_0} + 4(q_{\mathsf{G}} + q_{\mathsf{H}}) \cdot \mathrm{Adv}_{\mathsf{TDI},\mathcal{B}_1}^{\mathsf{PD\text{-}OW}}. \tag{62}$$

For the $\Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{9}}]$, we first rewrite game $\mathbf{G}_{\mathbf{15}}^{\mathbf{9}}$ as a search algorithm $\mathcal{A}_4(\mathsf{pk}, c^*, s^*, t^*)$, where the sampling of $(\mathsf{pk}, c^*, s^*, t^*)$ is the same as that in game $\mathbf{G}_{\mathbf{15}}^{\mathbf{9}}$. As for the r^*, which is uniformly sampled from $\{0,1\}^{k_0}$, is independent of $\mathcal{A}_4$'s view since the running of game $\mathbf{G}_{\mathbf{15}}^{\mathbf{9}}$ does not dependent on it. Then we have

$$\Pr[1 \leftarrow \mathbf{G}_{\mathbf{15}}^{\mathbf{9}}] = \Pr[(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R : (c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \leftarrow \mathcal{A}_4(\mathsf{pk}, c^*, s^*, t^*)]. \tag{63}$$

Here set R is defined in Eq. (52), and $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R$ actually means that

- $\exists (s, D_{\mathsf{H}}(s) \neq \perp)$ s.t. $c = f_{\mathsf{pk}}(s, r^* \oplus D_{\mathsf{H}}(s))$,
- or $\exists (r, D_{\mathsf{G}}(r) \neq \perp)$ s.t. $c = f_{\mathsf{pk}}(s^*, r \oplus r^* \oplus t^*)$.

For the first case, we can directly recover r^* by first computing $f_{\mathsf{sk}}^{-1}(c)$ to get $(s, r^* \oplus D_{\mathsf{H}}(s))$ and then extracting r^* from $r^* \oplus D_{\mathsf{H}}(s)$. For the second case, we can also recover r^* with a noticeable probability. That is, first compute $f_{\mathsf{sk}}^{-1}(c)$ to get $(s^*, r \oplus r^* \oplus t^*)$, and then use t^* with a uniformly-chosen r from $\{r : D_{\mathsf{G}}(r) \neq \perp\}$ to extract r^* from $r \oplus r^* \oplus t^*$. That is to say, if the $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})$ generated by $\mathcal{A}_4$ belongs to set R, we can recover r^* from $(c, y, D_{\mathsf{G}}, D_{\mathsf{H}})$ even

though $\mathcal{A}_4$'s view is independent of r^*. Recall that r^* is uniformly sampled from $\{0,1\}^{k_0}$, so through a simple probability analysis we have

$$\Pr[(c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \in R : (c, y, D_{\mathsf{G}}, D_{\mathsf{H}}) \leftarrow \mathcal{A}_4(\mathsf{pk}, c^*, s^*, t^*)] \leq (q_{\mathsf{G}}+1)\cdot 2^{-k_0}. \quad (64)$$

Combine the above equation with Eq. (63), we get $\Pr[1 \leftarrow \mathbf{G_{15}^9}] \leq (q_{\mathsf{G}}+1)\cdot 2^{-k_0}$.

Finally, by combining Eq.(48) to Eq.(64), we obtain

$$\begin{aligned}|\Pr[1 \leftarrow \mathbf{G_{14}}] - \Pr[1 \leftarrow \mathbf{G_{15}}]| \leq & 18 q_{\mathsf{D}}(q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \sqrt{2^{-k_0}} \\ & + 8 q_{\mathsf{D}}(q_{\mathsf{H}} + 1) \cdot \sqrt{2^{-n-k_1}} \\ & + 8 q_{\mathsf{D}}(q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \sqrt{\mathrm{Adv}^{\mathsf{PD\text{-}OW}}_{\mathsf{TDI},\mathcal{B}_1}},\end{aligned}$$

and we complete the proof.

D.9 Proof of Lemma 7

During the generation of b' in game $\mathbf{G_{15}}$ (Fig. 9), G is simulated by the $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$ defined in Eq. (22), which will directly XOR g^* on the output register by CNOT_{g^*}. This means that $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$ will not update the record of database D_{G} on the pair $(r^*, D_{\mathsf{G}}(r^*))$. Based on this fact, we can conclude that $\mathrm{U}_{g^*}^{r^*}$ commutes with $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$ since $\mathrm{U}_{g^*}^{r^*}$ defined in Eq. (15) only updates the record of D_{G} on the pair $(r^*, D_{\mathsf{G}}(r^*))$. Moreover, $\mathrm{U}_{g^*}^{r^*}$ also commutes with $\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*}$ and the unitary operation which simulates H during the generation of b' in game $\mathbf{G_{15}}$, since they act on different database registers. Therefore, we have that $\mathrm{U}_{g^*}^{r^*}$ commutes with both $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$ and $\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*}$, and by a similar analysis we have that $\mathrm{U}_{h^*}^{s^*}$ commutes with both $\mathsf{CStO}_{\mathsf{G}}^{r^*,g^*}$ and $\mathsf{CStO}_{\mathsf{H}}^{s^*,h^*}$. In addition, the adversary's operations performed before outputting b' also commute with $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$, since the adversary cannot perform operations on the database registers.

Note that $\mathrm{U}_{g^*}^{r^*}$ must commute with $\mathrm{U}_{h^*}^{s^*}$ since they act on different database registers, hence we find that the $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ added in $\mathrm{U}_{\mathsf{Search}}^{**}$ actually commute with all the other operations performed during the generation of b' in game $\mathbf{G_{15}}$. Now we can cancel out these two operations as follows, since they are both involutions:

- $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ performed before the generation of c^* can cancel out with the first two $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ performed in the first implementation of $\mathrm{U}_{\mathsf{Search}}^{**}$.
- The second $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ performed in the implementation of $\mathrm{U}_{\mathsf{Search}}^{**}$ can cancel out with the $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ performed firstly in the implementation of the next $\mathrm{U}_{\mathsf{Search}}^{**}$.
- As for the second $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ performed in the final implementation of $\mathrm{U}_{\mathsf{Search}}^{**}$, by the commutativity analysis given in the previous paragraph, we can move these two operations to the end of the game and then delete them without affecting the adversary's output.

So after the above cancelation of $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$, the performing of $\mathrm{U}_{g^*}^{r^*}$ and $\mathrm{U}_{h^*}^{s^*}$ just before the generation of c^* is canceled and $\mathrm{U}_{\mathsf{Search}}^{**}$ will degrade into $\mathrm{U}_{\mathsf{Search}}^{*}$. Meanwhile, g^* and h^* are no longer used by the game and hence can be removed. That is, game $\mathbf{G_{16}}$ given above is actually a rewrite of game $\mathbf{G_{15}}$, hence

$$\Pr[1 \leftarrow \mathbf{G_{15}}] = \Pr[1 \leftarrow \mathbf{G_{16}}].$$

D.10 Proof of Lemma 8

By mimicking the analysis of the game hop from $\mathbf{G}_{\mathbf{15}}^{\mathbf{6}}$ to $\mathbf{G}_{\mathbf{15}}^{\mathbf{7}}$, $\mathbf{G}_{\mathbf{15}}^{\mathbf{7}}$ to $\mathbf{G}_{\mathbf{15}}^{\mathbf{8}}$ and $\mathbf{G}_{\mathbf{15}}^{\mathbf{8}}$ to $\mathbf{G}_{\mathbf{15}}^{\mathbf{9}}$ in the proof of Lemma 6, we can transform game $\mathbf{G_{16}}$ to game $\mathbf{G_{17}}$, and the corresponding bound is

$$|\Pr[1 \leftarrow \mathbf{G_{16}}] - \Pr[1 \leftarrow \mathbf{G_{17}}]| \leq 4(q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \left(\sqrt{2^{-k_0}} + \sqrt{\mathrm{Adv}_{\mathsf{TDI},\mathcal{B}_2}^{\mathsf{PD\text{-}OW}}}\right),$$

where the PD-OW adversary $\mathcal{B}_2$ first chooses $j \leftarrow_{\$} [q_{\mathsf{G}}+q_{\mathsf{H}}]$[27], runs the adversary $\mathcal{A}$ in game $\mathbf{G}_{17}$ until its j-th query to G or H during the generation of b', and then measures the query register to get its output x. The running time of $\mathcal{B}_2$ can be upper bounded by $\mathcal{T}_{\mathcal{A}} + \mathcal{O}(q_{\mathsf{G}}^2 + q_{\mathsf{H}}^2) + \mathcal{O}(q_{\mathsf{D}} q_{\mathsf{G}} q_{\mathsf{H}}) \cdot \mathcal{T}_f$.

[27] Here we have assumed without loss of generality that, the total query number of G and H during the generation of b' in game $\mathbf{G_{17}}$ is exactly $q_{\mathsf{G}} + q_{\mathsf{H}}$ (if it does not, we can modify game $\mathbf{G_{17}}$ by adding a number of useless queries to G and H in the end).