On the (Quantum) Random Oracle Methodology: New Separations and More

Jiang Zhang[1], Yu Yu[2], Dengguo Feng[1], Shuqin Fan[1], and Zhenfeng Zhang[3]

[1] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[3] State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, China
{`jiangzhang09@gmail.com`, `yuyu@yuyu.hk`, `feng@tca.iscas.ac.cn`,
`shuqinfan78@163.com`, `zfzhang@tca.iscas.ac.cn`

Abstract. Motivated by the fact that in the quantum random oracle model (QROM) introduced by Boneh et al. (Asiacrypt 2011), honest parties (i.e., the cryptosystems) typically use the random oracle (RO) in a classical way while the adversary can send quantum queries to the RO, we first reformalize the classical RO (CRO) and the quantum RO (QRO) by adapting the indifferentiability framework of Maurer et al. (TCC 2004), and equipping a RO with a *private* and a (quantum) *public* interface for the honest parties and the adversary, respectively. Then, we give a new separation between the QROM and the ROM by showing that QRO is differentiable from CRO, which is technically different from Boneh et al.'s separation and is based on a new *information versus disturbance* lemma (that may be of independent interest).

We further abstract a class of BB-reductions in the ROM under the notion of committed-programming reduction (CPRed) for which the simulation of the RO can be easily quantized to handle quantum queries (from the adversary in the QROM). We show that 1) some well-known schemes such as the FDH signature and the Boneh-Franklin identity-based encryption are provably secure under CPReds; and 2) a CPRed associated with an instance-extraction algorithm implies a reduction in the QROM, which subsumes several recent results such as the security of the FDH signature by Zhandry (Crypto 2012) and the KEM variants from the Fujisaki-Okamoto transform by Jiang et al. (Crypto 2018).

We finally show that CPReds are incomparable to non-programming reductions (NPReds) and randomly-programming reductions (RPReds) formalized by Fischlin et al. (Asiacrypt 2010), which gives new insights into the abilities (e.g., observability and programmability) provided by the (Q)ROM, and the hardness of proving security in the QROM.

1 Introduction

In the random oracle model (ROM), all parties, including the adversary, are given access to an "idealized" random function (i.e., a random oracle). The ROM has been widely used to analyze the security of many (well-known) cryptosystems, but as pointed out by Boneh et al. [7], the classical ROM is problematic when considering quantum adversaries. This actually motivates them to introduce the quantum ROM (QROM) [7], where honest parties (e.g., the cryptosystems in post-quantum cryptography) typically use the random oracle (RO) in a classical way, but the adversary is explicitly allowed to send quantum queries to the RO. Boneh et al. [7] justified the necessity of the QROM by presenting an artificial identification protocol which is secure in the ROM but is insecure in the QROM. We observe that their results [7] raise several interesting problems.

First, the identification protocol for separating the QROM from the ROM in [7] is designed to directly utilize the gap in finding a collision of an n-bit output hash function between using the birthday attack with complexity $O(2^{n/2})$ in a classical way and using the Grover algorithm with complexity $O(2^{n/3})$ in a quantum way [18,10], which makes their arguments very sensitive to a "precise" timing model. In particular, they have to make sure that the running time of the artificial identification protocol is longer than $O(2^{n/3})$ "unit time" for a quantum adversary to run the Grover algorithm [18], but is shorter than $O(2^{n/2})$ "unit time" for a classical adversary to implement the birthday attack. *One might wonder if there are other separations between the QROM and the ROM.*

Second, it is well-known that a cryptosystem secure in the ROM might not be secure in the standard model [11,25,24]. Combining this with Boneh et al.'s separation [7] that there is a scheme which is secure in the ROM but is insecure in the QROM, *it is natural to ask what the relation between the QROM and the standard model is, or whether the security of a cryptosystem in the QROM could somehow guarantee the security of the cryptosystem in the real world.*

Third, following the introduction of the QROM [7], several papers [30,27,21] have been devoted to giving new security reductions in the QROM for some well-known schemes that were already proven secure in the ROM. However, most of them [7,30,27,21] rely on ad hoc techniques, which are usually specific on the concrete design of the schemes, rather than the inherent properties of existing reductions in the ROM.[4] *Is there a class of (black-box) reductions in the ROM, which can be amended to handle quantum adversaries in the QROM? If yes, what is the relation between this class and those known in the literature?*

1.1 Our Results

Motivated by the fact that in the QROM [7], honest parties (e.g., the cryptosytems) typically use the random oracle (RO) in a classical way while the adversary can make quantum queries to the RO, we reformalize the classical RO (CRO) and the quantum RO (QRO) by equipping a RO with two interfaces: a *private* one for the honest parties, and a (quantum) *public* one for the adversary. Specifically, we think of a CRO $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot))$ as being a primitive with two classical and equal interfaces (i.e., $\mathcal{O}^c_{priv}(\cdot) = \mathcal{O}^c_{pub}(\cdot)$), and a QRO $\mathcal{O}^q = (O^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot))$ as being a primitive where the private interface $O^q_{priv}(\cdot)$

[4] We note that the history-free reduction in [7] was introduced to abstract a class of existing BB-reductions restricted to signatures in the ROM.

is classical but the public interface $\mathcal{O}^q_{pub}(\cdot)$ accepts quantum queries from the adversary. Moreover, a scheme in the ROM (QROM) is treated as a (quantum) polynomial time construction which uses a CRO (QRO) as a component.

$$\begin{tikzcd}
\text{SM} \arrow[rr, Rightarrow, "{[11,25,24]}"] \arrow[dr, Rightarrow] & & \text{ROM} \arrow[ll, Rightarrow, bend right, "/" marking] \arrow[dl, Rightarrow, bend left, "/" marking, "{[7], \text{Sec. 3.1}}"'] \\
& \text{QROM} \arrow[ur, Rightarrow] \arrow[ul, Rightarrow, bend left, "/" marking, "\text{Sec.3.2}"] &
\end{tikzcd}$$

Fig. 1. Relations among the ROM, the QROM and the standard model (SM): $A \Rightarrow B$ (resp., $A \nRightarrow B$) means that the security in A always implies (resp., does not necessarily imply) that in B.

$$\begin{tikzcd}
\text{NPRed} \arrow[rr, Rightarrow, "{[13]}"] \arrow[dr, Rightarrow] \arrow[ddr, Rightarrow, bend right, "/" marking, "\text{Sec.5.1}"'] & & \text{RPRed} \arrow[ll, Rightarrow, bend right, "/" marking] \arrow[dl, Rightarrow] \arrow[ddl, Rightarrow, dashed, "/" marking, "\text{Sec.5.2}"] \\
& \text{FPRed} & \\
& \text{CPRed} \arrow[u, Rightarrow] \arrow[uul, Rightarrow, dashed, "/" marking] \arrow[uur, Rightarrow, bend right, "/" marking] &
\end{tikzcd}$$

Fig. 2. Relations among the NPRed, the RPRed, the FPRed and the CPRed: $X \Rightarrow Y$ (resp., $X \nRightarrow Y$) means that the existence of X implies (resp., does not necessarily imply) that of Y.

We then adapt the indifferentiability framework of Maurer et al. [24] to the quantum setting, and separate the QROM from the ROM by showing that QRO is differentiable from CRO. Technically, we construct an efficient algorithm $\mathcal{D}$ such that for all (even unbounded) algorithm $\mathcal{P}$ making at most a polynomial number of queries to a CRO $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot))$, $\mathcal{D}$ can distinguish a real QRO $\mathcal{O}^q = (O^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot))$ from the simulated one $\mathcal{O}' = (\mathcal{O}^c_{priv}(\cdot), \mathcal{P}^{\mathcal{O}^c_{pub}(\cdot)}(\cdot))$. In other words, no QPT adversary $\mathcal{P}$ in a CRO environment can cheat $\mathcal{D}$ that it is in a QRO environment by simulating a quantum interface for the CRO. Moreover, one can easily construct an (artificial) system such that it is secure in the ROM but is insecure in the QROM, by using $\mathcal{D}$ as a subroutine, and letting the system execute a malicious action (e.g., leaking the secret) if and only if $\mathcal{D}$ detects that the whole system is running in a QRO environment. For a full relation among the ROM, the QROM and the standard model (see Fig. 1), we also separate the standard model from the QROM by adapting the separation between the standard model and the ROM in [24].

We further abstract a class of black-box (BB) reductions in the ROM under the notion of committed-programming reduction (CPRed) for which the simulation of the RO can be easily quantized. We show that 1) some well-known schemes [4,17,29,19] such as the full-domain hash (FDH) signature [4] and the Boneh-Franklin identity-based encryption [8] are provable under CPReds, and 2) a CPRed associated with an instance-extraction algorithm implies a reduction in the QROM, which not only subsumes several recent results [7,34,21] such as the security of the FDH from trapdoor permutations (TDP) by Zhandry [34] and the "implicit-rejection" KEM variants from the Fujisaki-Okamoto transform by Jiang et al. [21], but also gives new security reductions for other schemes such as the "implicit-rejection" KEM variants from TDPs [29] in the QROM.

We finally show that CPReds are incomparable to both non-programming reductions (NPReds) and randomly-programming reductions (RPReds) formalized by Fischlin et al. [13] (see Fig. 2), which gives new insights into the abilities (e.g., observability and programmability) provided by the (Q)ROM, and the hardness of proving security in the QROM (e.g., a scheme which is provably secure under NPReds[5] might still not be provable in the QROM as a reduction in the QROM seems to be limited in both observability and programmability). Technically, we show that the OAEP encryption from TDPs [3] which was proven secure under NPReds by Fujisaki et al. [16] is not provable under CPReds, and that the FDH signature [4] which was shown not to be provable under RPReds by Fischlin et al. [13] is provably secure under CPReds. Since a NPRed always implies a RPRed, we obtain a full relation in Fig. 2, where full-programming reductions (FPReds) denote all BB-reductions in the ROM.

1.2 Technical Overview

QRO is differentiable from CRO. We first note that the two interfaces of a QRO/CRO only specify the way of accessing the same "random function", and the responses from both interfaces are always consistent. Specifically, let n, m be positive integers, and let $\mathcal{F}_{n,m} = \{f : \{0,1\}^n \to \{0,1\}^m\}$ be a family of functions. Then, given a CRO $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot)) \xleftarrow{\$} \mathcal{F}_{n,m}$, we always have $\mathcal{O}^c_{priv}(x) = \mathcal{O}^c_{pub}(x)$ for any $x \in \{0,1\}^n$. Moreover, if we encode x into a quantum state $|x, 0_m\rangle$, the measurement outcome on $\mathcal{O}^q_{pub}(|x, 0_m\rangle)$ is always equal to $(x, O^q_{priv}(x))$ for a QRO $\mathcal{O}^q = (O^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot)) \xleftarrow{\$} \mathcal{F}_{n,m}$.

Our goal is to construct a quantum polynomial time (QPT) algorithm $\mathcal{D}$ such that for all QPT algorithm $\mathcal{P}$ given access to (the public interface of) a CRO $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot))$, $\mathcal{D}$ can distinguish a QRO $\mathcal{O}^q = (O^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot))$ from a simulated one $\mathcal{O}' = (\mathcal{O}^c_{priv}(\cdot), \mathcal{P}^{\mathcal{O}^c_{pub}(\cdot)}(\cdot))$. Since there is a complexity gap for (quantum) algorithms to invert an n-bit input function (resp., find a collision of an m-bit output hash function) between using $O(2^n)$ (resp., $O(2^{m/2})$) classical queries and using $O(2^{n/2})$ (resp., $O(2^{m/3})$) quantum queries [18,10], one can construct a trivial distinguisher $\mathcal{D}$ by directly utilizing this complexity gap. Unfortunately, such a distinguisher does not suffice for our purpose as it has to make sub-exponentially many (i.e., $O(2^{n/2})$ or $O(2^{m/3})$) quantum queries, which may be useless for most schemes that are based on quantum sub-exponential time (or even polynomial time) hardness assumptions. Instead, we will construct an efficient distinguisher $\mathcal{D}$ which only makes at most two quantum queries.

Our starting point is that $\mathcal{D}$ can encode exponentially many classical queries into a single quantum query $|\phi\rangle = \sum_{x \in \{0,1\}^n} |x, 0_m\rangle$ in superposition, while $\mathcal{P}$ can only make a polynomial number of classical queries to $\mathcal{O}^c_{pub}(\cdot)$. In particular, if D sends a quantum query $|\phi\rangle$ to $\mathcal{P}$, then it is infeasible for $\mathcal{P}$ to compute a "valid" response $|\psi\rangle = \sum_{x \in \{0,1\}^n} \left|x, \mathcal{O}^c_{pub}(x)\right\rangle = \sum_{x \in \{0,1\}^n} |x, \mathcal{O}^c_{priv}(x)\rangle$ that

[5] Note that a NPRed can observe the RO queries and responses between the adversary and an external RO (accessible by all parties), but cannot program the RO responses.

should be consistent with $O^c_{priv}(\cdot)$. One main problem is that $\mathcal{D}$ cannot check if a response from $\mathcal{P}^{\mathcal{O}^c_{pub}(\cdot)}(\cdot)$ is valid or not, as it cannot extract all the classical pairs $\{(x, \mathcal{O}^c_{priv}(x))\}_{x\in\{0,1\}^n}$ from the state $|\psi\rangle$ by the quantum mechanics.

To get around the above obstacle, we establish a new *information versus disturbance* lemma (see Lemma 3) by adapting the security proof [6] of the well-known BB84 quantum key distribution [5]. Basically, the lemma gives a direct connection between the information that an algorithm obtains from a random quantum query and the disturbance that it causes to the query state, which may be of independent interest. For our goal, the lemma allows $\mathcal{D}$ to send a quantum query $|\phi\rangle = \sum_{x\in X} |x, 0_m\rangle$ which consists of a random subset $X \subseteq \{0,1\}^n$ of classical queries having the same pattern to $\mathcal{P}$, such that $\mathcal{P}$ needs to know sufficient information about X to make a classical query $x \in X$ to $\mathcal{O}^c_{pub}(\cdot)$. After obtaining the response $|\psi\rangle$ to $|\phi\rangle$ from $\mathcal{P}$, we then let $\mathcal{D}$ flip a random bit to either check that a measurement on $|\psi\rangle$ always result in a consistent pair $(x, \mathcal{O}^c_{pub}(x)) = (x, \mathcal{O}^c_{priv}(x))$ for some $x \in X$, or check that the query state $|\phi\rangle = \sum_{x\in X} |x, 0_m\rangle$ is not disturbed by using another quantum query. Since a QRO will always pass both checks, a successful $\mathcal{P}$ must perform some operations on $|\phi\rangle$ to obtain sufficient information about X without (significantly) disturbing the state $|\phi\rangle$, which is infeasible by our *information versus disturbance* lemma.

CPReds and Implications. Our above separation roughly says that one cannot efficiently simulate a QRO by using a classical RO, which motivates us to directly consider the class of BB-reductions in the ROM for which the simulation of RO can be easily quantized to handle quantum queries. Informally, we introduce the notion of committed-programming reduction (CPRed), and say that a BB-reduction in the ROM is a CPRed if it uses two sub-algorithms (with some common inputs) which do not interact with each other to simulate the RO and other oracles, respectively. In particular, the sub-algorithm for simulating other oracles cannot see the RO queries or affect the answers to the RO queries from the adversary. The term "committed-programming" captures the feature that the reduction must fix the strategy of simulating the RO (to keep consistency between the two sub-algorithms), and cannot adaptively program the RO to get out of a "bad situation" in simulating other oracles. One might think the above restrictions are somewhat strong. But some well-known schemes [4,17,29,19] such as the full-domain hash (FDH) signature [4] and the Boneh-Franklin identity-based encryption [8] are actually provably secure under CPReds.

Moreover, we show that if a CPRed $\mathcal{S}$ from problem P (e.g., the one-wayness of trapdoor permutations) to problem Q (e.g., the unforgeability of the FDH signature) has an associated instance-extraction algorithm, then it can be upgraded into a reduction in the QROM by quantizing the RO simulation. Informally, the instance-extraction algorithm can extract an instance of P (and produce an auxiliary string for the RO simulation) from an instance of Q, which is needed to ensure that the sub-algorithm for simulating the RO can be quantized to safely simulate a QRO, and that a real QRO can be smoothly replaced by the simulated QRO in the security proof. Intuitively, as the simulation of other oracles for a CPRed is oblivious to the RO queries from the adversary $\mathcal{A}$, it seems harmless

to allow $\mathcal{A}$ to quantumly access the RO. Note that the history-free reduction [7] directly contains a similar instance algorithm (restricted to the signatures) in the definition, we prefer not to do so since it only serves for getting a reduction in the QROM and is not necessary for a common BB-reduction in the ROM.
CPReds are incomparable to both NPReds and RPReds. We investigate the OAEP encryption from TDPs [3] which was proven secure under NPReds by Fujisaki et al. [16], and show that it is not provable under CPReds. The key point is that the reduction [16] for the CCA-security of the OAEP encryption crucially relies on the observability of the RO queries to answer the decryption queries, which is not achievable for CPReds (as the sub-algorithm for simulating the decryption oracle is not allowed to interact with the sub-algorithm for simulating the RO, and thus cannot see the RO queries from the adversary). We will use the two-oracle separation technique of Hsiao and Reyzin [20] to formally rule out the existence of CPReds for the CCA-security of the OAEP encryption.

We also investigate the FDH signature [4] which was shown not to be provable under RPReds by Fischlin et al. [13], and show that it is provable under CPReds. Our main observation is that existing reductions for the unforgability of the FDH signature need to program the RO response such that the corresponding preimage under some permutation is known (in any signing query). This is not achievable for a RPRed since it cannot directly choose the RO responses, but is achievable for a CPRed since the simulation of the RO can be done by using a sub-algorithm equipped with two random functions (which can be efficiently simulated by using k-wise independent functions [34]): one for guessing if a RO query from the adversary will be used in a signing query, and the other for picking a random preimage to program the RO response if the guess is "yes".

1.3 Related Work and Discussion

The random oracle methodology was first formalized by Bellare and Rogaway [2] and has been widely used to design and analyze many schemes such as the OAEP encryption [3] and the FDH signature [4]. Although most "honestly-designed" schemes in the ROM seem to keep the security in practice, but the soundness of this methodology has been questioned by the literatures [11,25,24]. The first separation between the ROM and the standard model was given by Canetti, Goldreich and Halevi [11], who showed that there exist signature and encryption schemes that are secure in the ROM, but for which any implementation of the RO results in insecure schemes. Later, Maurer et al. [24] introduced the notion of indifferentiability, and gave a more simple separation.

There are two main abilities that a black-box (BB) reduction can get from the ROM: the observability (i.e., the ability to see all the RO queries [1]) and the programmability (i.e., the ability to set the RO responses [13]). In 2010, Fischlin et al. [13] first formalized three notions of BB-reductions with different programmability in the ROM: fully-programming reduction (FPRed), randomly-programming reduction (RPRed) and non-programming reduction (NPRed). Later, Ananth and Bhaskar [1] considered the security of several existing schemes under BB-reductions without observability in the ROM.

Boneh et al. [7] first introduced the quantum ROM (QROM), and separate the QROM from the ROM by giving an "artificial" identification protocol which is provably secure in the ROM, but is insecure in the QROM. Their separation crucially relies on the speedup of the Grover quantum algorithm [18,10] over classical algorithms in finding a collision of hash functions (and a precise timing model), while we resort to a technically different way and construct an efficient algorithm which can distinguish a CRO environment from a QRO environment by directly using the quantum mechanics. Boneh et al. [7] also considered a class of BB-reductions (which are restricted to signatures) in the ROM, i.e., history-free reductions, and show that a signature scheme with a history-free reduction may still imply the security in the QROM.

Following [7], many researchers have devoted to giving security proofs for existing schemes [34,12,30,19,23,21] and designing new schemes [31,32,27] in the QROM. However, as commented in [13,30], it might be hard to prove the security of the OAEP encryption [3] in the QROM. We note that our results essentially provide some new insights into this "hypothesis" as we have showed that the OAEP encryption cannot be proven secure under CPReds, and that a CPRed with an instance-extraction algorithm implies a reduction in the QROM. Note that Targhi and Unruh [30] proved the security of a variant of the OAEP encryption in the QROM by adding an additional hash (modeled as a RO) to each ciphertext, which basically can be seen as a technique to "force the adversary to submit the input" of the RO to the reduction and thus force the adversary to provide a direct connection between the RO queries and the decryption queries.

2 Preliminaries

Let κ be the security parameter. The standard notations O, ω are used to classify the growth of functions. Denote log as the logarithm with base 2. A function $f(\kappa)$ is negligible in κ if for every positive c, we have $f(\kappa) < \kappa^{-c}$ for sufficiently large κ. By $\text{negl}(\kappa)$ we denote an arbitrary negligible function. The notation $\overset{\$}{\leftarrow}$ denotes randomly choosing elements from a distribution (or the uniform distribution over a finite set). Denote ϵ (resp., $\emptyset$) as an empty string (resp., set).

Let $\mathbb{C}$ be the set of complex numbers, and let $\mathbb{C}^N$ be the complex vector space of N dimension, where $N \geq 1$ is an integer. The bra-ket notations of $\langle\cdot|$ and $|\cdot\rangle$ are used to denote row and column vectors in $\mathbb{C}^N$, respectively. For any vector $v \in \mathbb{C}^N$, v^T denote the transpose of v, and v^* denotes the conjugate transpose of v. For any vectors $w = (w_0, \ldots, w_{N-1})^T, v = (v_0, \ldots, v_{N-1})^T \in \mathbb{C}^N$, the inner product between w and v is defined as $\langle w|v\rangle = \sum_{i=0}^{N-1} w_i^* v_i \in \mathbb{C}$.

2.1 Quantum Computation

We briefly recall some background for quantum computation, and refer to [26] for more information. Formally, a quantum system $\mathcal{Q}$ with N configurations labeled by $\{0, \ldots, N-1\}$ is associated to the Hilbert space $\mathcal{H}_N = \mathbb{C}^N$ with the inner product $\langle w|v\rangle = \sum_{i=0}^{N-1} w_i^* v_i \in \mathbb{C}$. A pure state of $\mathcal{Q}$ is specified by a column vector $|\phi\rangle \in \mathcal{H}_N$ of norm 1 (i.e., $\langle\phi|\phi\rangle = 1$), which assigns a (complex) weight to each configuration in $\{0, \dots, N-1\}$. The "computational basis" for $\mathcal{Q}$ is $\{|0\rangle, |1\rangle, \dots, |N-1\rangle\}$, where $|i\rangle$ assigns weight 1 to configuration i, and weight 0 to any other configuration $j \neq i$. By definition, $\{|0\rangle, |1\rangle, \dots, |N-1\rangle\}$ forms an orthonormal basis for $\mathcal{H}_N$, and any pure state $|\phi\rangle$ can be written as $|\phi\rangle = \sum_i \alpha_i |i\rangle$, where $\sum_i |\alpha_i|^2 = 1$. Given two quantum systems $\mathcal{Q}_0$ and $\mathcal{Q}_1$ over $\mathcal{H}_N$ and $\mathcal{H}_M$, respectively, the joint quantum system is defined via the tensor product $\mathcal{H}_N \otimes \mathcal{H}_M$, and the product state of $|\phi\rangle_0 \in \mathcal{H}_N$ and $|\phi\rangle_1 \in \mathcal{H}_M$ is denoted by $|\phi\rangle_0 \otimes |\phi\rangle_1$, or simply $|\phi_0, \phi_1\rangle$.

A qubit is a quantum system with $N = 2$ configurations labeled by $\{0, 1\}$. An n-qubit system is the joint quantum system of n qubits. The standard computational basis $\{|x\rangle\}_{x \in \{0,1\}^n}$ for an n-qubit system is given by $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$, where $x = x_1 \dots x_n$. Any classical bit-string $x \in \{0,1\}^n$ can be encoded into a quantum state $|x\rangle$, and any arbitrary pure n-qubit state $|\phi\rangle$ can be written as $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$, where $\sum_{x \in \mathcal{X}} |\alpha_x|^2 = 1$. A pure state $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$ is in *superposition* if $|\alpha_x|^2 < 1$ for all $x \in \{0,1\}^n$.

Quantum Measurement. Let $B = \{|x\rangle\}_{x \in \mathcal{X}}$ be any orthonormal basis of $\mathcal{H}_N$. Given a pure state $|\phi\rangle \in \mathcal{H}_N$, we can measure it in the basis B, obtaining a value x with probability $|\langle x|\phi\rangle|^2$. This operation induces a probability distribution $D_\phi(x) = |\langle x|\phi\rangle|^2$ over $\mathcal{X}$. After measurement, the state $|\phi\rangle$ collapses to $|x\rangle$, which will not change under subsequent measurements in the same basis B (but may still change under measurements in other basis). We can also perform partial measurement on a pure state in a joint quantum system. Formally, let $|\phi\rangle \in \mathcal{H}_N \otimes \mathcal{H}_M$ be a pure state of the joint quantum system $\mathcal{Q} = \mathcal{Q}_0 \otimes \mathcal{Q}_1$, and let $B_0 = \{|x\rangle\}_{x \in \mathcal{X}}$ (resp., $B_1 = \{|y\rangle\}_{y \in \mathcal{Y}}$) be an orthonormal basis of $\mathcal{H}_N$ (resp., $\mathcal{H}_M$). After a partial measurement on $|\phi\rangle$ in the basis B_0, we will obtain a value $x \in \mathcal{X}$ with probability $p_x = \sum_{y \in \mathcal{Y}} |\langle x, y|\phi\rangle|^2$, and the state $|\phi\rangle = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \alpha_{x,y} |x, y\rangle$ collapses to $\sum_{y \in \mathcal{Y}} \frac{\alpha_{x,y}}{\sqrt{p_x}} |x, y\rangle$.

Quantum Algorithms. A quantum algorithm $\mathcal{A}$ over a Hilbert space $\mathcal{H}_N$ with an orthonormal basis $\{|x\rangle\}_{x \in \mathcal{X}}$ is specified by a unitary transformation U, which takes an initial state $|\phi\rangle$ as input, and outputs a result obtained by performing a measurement on the final state $|\psi\rangle = U|\phi\rangle$. We say that a quantum algorithm $\mathcal{A}$ is efficient if U is composed of a polynomial number of universal basis gates (e.g., the Hadamard, CNOT, and $\pi/8$ gates). Let $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$ be any sets such that $\mathcal{Y}$ is the additive group of $\mathbb{Z}_2^\ell$ for some $\ell \in \mathbb{N}$. Let $B = \{|x, y, z\rangle\}_{x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}}$ be the corresponding orthonormal basis. For any function $f : \mathcal{X} \to \mathcal{Y}$, define U_f as the unitary transformation that maps $|x, y, z\rangle$ into $|x, y \oplus f(x), z\rangle$. By definition, the inverse transformation of U_f is itself. Let Id be the identity transformation.

Let O_f be an oracle that computes the unitary transformation U_f. An oracle quantum algorithm A^{O_f} with at most q queries to O_f is specified by a sequence of $q + 1$ unitary transformation $U_0, \cdots, U_q$. Specifically, given an initial state $|\phi\rangle$, the algorithm A^{O_f} outputs a result obtained by performing a measurement on the final state $|\psi\rangle = U_q U_f U_{q-1} \dots U_f U_0 |\phi\rangle$. Similarly, one can define oracle algorithm $A^{O_{f_1},\ldots,O_{f_k}}$ with access to a collection of oracles $\{O_{f_1},\ldots,O_{f_k}\}$, which is polynomially equivalent to an oracle algorithm B^O with access to a single oracle $O(k,x)=O_{f_k}(x)$. The following lemma is implicit in [33].

Lemma 1 (Algorithmic One-way to Hiding [33,19]). *Let $\mathcal{F}$ be the family of functions from $\mathcal{X}$ to $\mathcal{Y}$, and let $\mathcal{O}:\mathcal{X}\to\mathcal{Y}$ be a random oracle. Let D be any arbitrary probability distribution over $\mathcal{X}$. Let E be any arbitrary (probabilistic) algorithm which takes a pair $(x,y)\in\mathcal{X}\times\mathcal{Y}$ as inputs, outputs a bit string $inp\in\{0,1\}^*$. Consider an oracle algorithm $\mathcal{A}$ that makes at most q queries to $\mathcal{O}$. Let $\mathcal{B}$ be an oracle algorithm that on input $inp\in\{0,1\}^*$ does the following: pick $i\xleftarrow{\$}\{1,\ldots,q\}$, run $\mathcal{A}^{\mathcal{O}}(inp)$ until (just receiving) the i-th query, measure the argument of the query in the computational basis, output the measurement outcome. (When $\mathcal{A}$ makes less than i queries, $\mathcal{B}$ outputs $\perp$.) Let*

$$P^1_{\mathcal{A}} := \Pr\Big[b'=1 : \mathcal{O}\xleftarrow{\$}\mathcal{F}, x\xleftarrow{\$}D, y=\mathcal{O}(x), inp\leftarrow E(x,y), b'\xleftarrow{\$}\mathcal{A}^{\mathcal{O}}(inp)\Big]$$
$$P^2_{\mathcal{A}} := \Pr\Big[b'=1 : \mathcal{O}\xleftarrow{\$}\mathcal{F}, x\xleftarrow{\$}D,\ y\xleftarrow{\$}\mathcal{Y},\ inp\leftarrow E(x,y), b'\xleftarrow{\$}\mathcal{A}^{\mathcal{O}}(inp)\Big]$$
$$P_{\mathcal{B}} := \Pr\Big[x=x' : \mathcal{O}\xleftarrow{\$}\mathcal{F}, x\xleftarrow{\$}D,\ y\xleftarrow{\$}\mathcal{Y},\ inp\leftarrow E(x,y), x'\xleftarrow{\$}\mathcal{B}^{\mathcal{O}}(inp)\Big]$$

Then $|P^1_{\mathcal{A}}-P^2_{\mathcal{A}}|\le 2q\sqrt{P_{\mathcal{B}}}$.

We clarify that the original one-way to hiding lemma in [33,19] only considers the uniform distribution, i.e., $x\xleftarrow{\$}\mathcal{X}$, but the proof given in [33] essentially applies to any distribution D. Here is a brief explanation: if $\mathcal{A}$ does not use x in any RO queries, then it cannot distinguish $(x,\mathcal{O}(x))$ from the pair (x,y) with random $y\xleftarrow{\$}\mathcal{Y}$ (i.e., we already have $|P^1_{\mathcal{A}}-P^2_{\mathcal{A}}|\le \mathrm{negl}(\kappa)$), while if $\mathcal{A}$ uses x in some RO queries, then $\mathcal{B}$ can find x by randomly choosing and measuring one of the queries from $\mathcal{A}$. This fact is essentially independent from the choice of x (note that x is always chosen from the same distribution D in Lemma 1).

2.2 Indifferentiability

As a generalization of indistinguishability, the notion of indifferentiability was introduced by Maurer et al. [24] to deal with the setting where each primitive is assumed to have two interfaces (which may be different or not): *private* and *public*, modeling the access of the honest parties and the adversary, respectively. We adapt the notion of indifferentiability to the quantum setting. Formally, a primitive $\mathcal{I}=(\mathcal{I}_{priv}(\cdot),\mathcal{I}_{pub}(\cdot))$ is said to be (resp., quantum computationally) indifferentiable from another primitive $\mathcal{J}=(\mathcal{J}_{priv}(\cdot),\mathcal{J}_{pub}(\cdot))$ if for all (resp., QPT) distinguisher $\mathcal{D}(\cdot)$, there is a (resp., QPT) algorithm $\mathcal{P}(\cdot)$ such that

$$\left|\Pr\Big[\mathcal{D}(1^\kappa)^{\mathcal{I}_{priv}(\cdot),\mathcal{I}_{pub}(\cdot)}=1\Big]-\Pr\Big[\mathcal{D}(1^\kappa)^{\mathcal{J}_{priv}(\cdot),\mathcal{P}(\cdot)^{\mathcal{J}_{pub}(\cdot)}}=1\Big]\right|\le \mathrm{negl}(\kappa)$$

holds, where κ is the security parameter. Note that, unlike indistinguishability, indifferentiability is not symmetric, i.e., the fact that $\mathcal{I}$ is indifferentiable from $\mathcal{J}$ does not necessary imply that $\mathcal{J}$ is indifferentiable from $\mathcal{I}$. However, if the primitives have no public interfaces, then both notions are equivalent.

Definition 1 ([24]). *A cryptosystem $\mathcal{U} = (\mathcal{U}_{priv}(\cdot), \mathcal{U}_{pub}(\cdot))$ is said to be (resp., computationally) at least as secure as another cryptosystem $\mathcal{V} = (\mathcal{V}_{priv}(\cdot), \mathcal{V}_{pub}(\cdot))$ if for all (resp., QPT) algorithm $\mathcal{D}$ the following holds: For any (resp., QPT) adversary $\mathcal{A}$ accessing $\mathcal{U}$ there is another adversary $\mathcal{B}$ accessing $\mathcal{V}$, we have that*

$$\left|\Pr\left[\mathcal{D}(1^\kappa)^{\mathcal{U}_{priv}(\cdot), \mathcal{A}(\cdot)^{\mathcal{U}_{pub}(\cdot)}} = 1\right] - \Pr\left[\mathcal{D}(1^\kappa)^{\mathcal{V}_{priv}(\cdot), \mathcal{B}(\cdot)^{\mathcal{V}_{pub}(\cdot)}} = 1\right]\right| \leq \text{negl}(\kappa).$$

By definition the public interface of a primitive is always available to the adversary. In particular, if $\mathcal{C}(\mathcal{I})$ is a cryptosystem which uses primitive $\mathcal{I}$ as a component, the public interface $\mathcal{I}_{pub}(\cdot)$ of $\mathcal{I} = (\mathcal{I}_{priv}(\cdot), \mathcal{I}_{pub}(\cdot))$ is still available to the adversary attacking $\mathcal{C}(\mathcal{I})$. This is different from the notion of indistinguishability, where an adversary attacking a system $\mathcal{C}(\cdot)$ is not allowed to directly access its building blocks. The following lemma is obtained by directly adapting [24, Theorem 1] to the quantum setting (since [24, Theorem 1] is proved by simply renaming the interfaces of the involved systems).

Lemma 2 ([24]). *Let $\mathcal{I} = (\mathcal{I}_{priv}(\cdot), \mathcal{I}_{pub}(\cdot))$ and $\mathcal{J} = (\mathcal{J}_{priv}(\cdot), \mathcal{J}_{pub}(\cdot))$ be two arbitrary primitives. Then, $\mathcal{I}$ is (resp., quantum computationally) indifferentiable from $\mathcal{J}$ if and only if for all (resp., QPT) construction $\mathcal{C}(\cdot)$, we have that $\mathcal{C}(\mathcal{I})$ is (resp., quantum computationally) at least as secure as $\mathcal{C}(\mathcal{J})$.*

3 Separations of the (Quantum) Random Oracle Model

Note that the only difference between the ROM and the QROM is how the adversary uses the RO: the adversary in the ROM only makes classical queries to the RO, but that in the QROM can make quantum queries to the RO. This fact motivates us to formalize the (Q)ROM by using the indifferentiability framework in [24], and equipping the RO with a *private* and a *public* interface.

Let n, m be positive integers, and let $\mathcal{F}_{n,m} = \{f : \{0,1\}^n \to \{0,1\}^m\}$ be a family of functions. A RO $\mathcal{O}(\cdot)$ from $\{0,1\}^n$ to $\{0,1\}^m$ is a "black-box" random function uniformly chosen from $\mathcal{F}_{n,m}$, which can only be accessed via given interfaces. We say that $\mathcal{O}(\cdot) \overset{\$}{\leftarrow} \mathcal{F}_{n,m}$ is a classical RO (CRO) if it provides a private interface $\mathcal{O}_{priv}(\cdot)$ for the honest parties and an essentially the same public interface $\mathcal{O}_{pub}(\cdot)$ for the adversary (i.e., $\mathcal{O}_{pub}(\cdot) = \mathcal{O}_{priv}(\cdot) = \mathcal{O}(\cdot)$).

In contrast, we say that $\mathcal{O}(\cdot) \overset{\$}{\leftarrow} \mathcal{F}_{n,m}$ is a quantum RO (QRO) if it provides two different interfaces $\mathcal{O}_{priv}(\cdot)$ and $\mathcal{O}_{pub}(\cdot)$, where the private interface $\mathcal{O}_{priv}(\cdot)$ allows the honest parties to classically access $\mathcal{O}(\cdot)$ and returns $\mathcal{O}_{priv}(x) = \mathcal{O}(x)$ to a classical query $x \in \{0,1\}^n$, while the public interface $\mathcal{O}_{pub}(\cdot)$ enables the adversary to quantumly access $\mathcal{O}(\cdot)$ and returns a state $\mathcal{O}_{pub}(|x, z\rangle) = |x, z \oplus \mathcal{O}(x)\rangle$ to a quantum query $|x, z\rangle$ of $n + m$ qubits.

We will simply denote a RO as $\mathcal{O} = (\mathcal{O}_{priv}(\cdot), \mathcal{O}_{pub}(\cdot))$ or $\mathcal{O}(\cdot)$ if we do not distinguish whether it is classical or quantum. Otherwise, we directly denote a classical RO (CRO) as $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot))$ and a quantum RO (QRO) as $\mathcal{O}^q = (O^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot))$ by using superscripts "c" and "q". In this formalization, a cryptosystem in the (Q)ROM is a (quantum) polynomial time construction which uses a (QRO) CRO as a component.

3.1 The QROM is Stronger than the ROM

By Lemma 2, it suffices to show that the CRO is indifferentiable from the QRO, but the reverse does not hold. We will do this by using Theorem 1 and Theorem 2.

Theorem 1. *Let $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot)) \overset{\$}{\leftarrow} \mathcal{F}_{n,m}$ be a CRO, and let $\mathcal{O}^q = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot)) \overset{\$}{\leftarrow} \mathcal{F}_{n,m}$ be a QRO. Then, $\mathcal{O}^c$ is indifferentiable from $\mathcal{O}^q$. In particular, for all (even unbounded) distinguisher $\mathcal{D}$, there exists a QPT algorithm $\mathcal{P}$ such that*

$$\left|\Pr\left[\mathcal{D}(1^\kappa)^{\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot)} = 1\right] - \Pr\left[\mathcal{D}(1^\kappa)^{\mathcal{O}^q_{priv}(\cdot), \mathcal{P}(\cdot)^{\mathcal{O}^q_{pub}(\cdot)}} = 1\right]\right| = 0.$$

Proof. Note that the private interfaces of both $\mathcal{O}^c$ and $\mathcal{O}^q$ models the access of the honest parties to a "black-box" random function, and essentially have identical behaviors. Moreover, the public interface $\mathcal{O}^c_{pub}(\cdot)$ of $\mathcal{O}^c$ has the same behavior as its private interface $\mathcal{O}^c_{priv}(\cdot)$ by definition. To finish the proof, we construct a QPT oracle algorithm $\mathcal{P}(\cdot)$ which receives a classical query x, and returns a response y such that it is consistent with the one obtained by sending x to $\mathcal{O}^q(\cdot)$ via the private interface $\mathcal{O}^q_{priv}(\cdot)$. Specifically, when receiving a query $x \in \{0,1\}^n$ from the distinguisher $\mathcal{D}$, the algorithm $\mathcal{P}(\cdot)$ performs the followings:

– Prepare a state $|x, 0_m\rangle$ of $n+m$ qubits;
– Send a quantum query $|x, 0_m\rangle$ to its own oracle $\mathcal{O}^q(\cdot)$ via the public interface $\mathcal{O}^q_{pub}(\cdot)$, which will return a state $|x, \mathcal{O}^q(x)\rangle$ by definition;
– Measure the state $|x, \mathcal{O}^q(x)\rangle$ to obtain $\hat{x} \in \{0,1\}^n$ and $\hat{y} = \mathcal{O}^q(\hat{x})$. Return $\hat{y} \in \{0,1\}^m$ to the distinguisher $\mathcal{D}$.

By definition, $\hat{y}$ is consistent with the response obtained by querying $\hat{x}$ to $\mathcal{O}^q(\cdot)$ via the private interface $\mathcal{O}^q_{priv}(\cdot)$. Thus, both $\mathcal{O}^q_{priv}(\cdot)$ and $\mathcal{P}(\cdot)^{\mathcal{O}^q_{pub}(\cdot)}$ have the same behaviors in the distinguisher $\mathcal{D}$'s view, which is identical to the case when $\mathcal{D}$ is given access to $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot))$. This completes the proof. □

At first glance, one might think that a "reversal" variant (which first measures the quantum query, and then uses the outcome to make a classical query) of the algorithm $\mathcal{P}(\cdot)$ in the proof of Theorem 1 suffices to show that $\mathcal{O}^q$ is indifferentiable from $\mathcal{O}^c$. Unfortunately, we show that any (even unbounded) algorithm making only a polynomial number of classical queries will not work.

Theorem 2. *Let κ be the security parameter. Let $n \geq \omega(\log \kappa)$ and $m \geq 1$ be two integers. Let $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot)) \overset{\$}{\leftarrow} \mathcal{F}_{n,m}$ be a CRO, and let $\mathcal{O}^q = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot)) \overset{\$}{\leftarrow} \mathcal{F}_{n,m}$ be a QRO. Then, $\mathcal{O}^q$ is differentiable from $\mathcal{O}^c$. In particular, for sufficiently large n, there exists a QPT distinguisher $\mathcal{D}$ such that for all (even unbounded) quantum algorithm $\mathcal{P}(\cdot)$ making only a polynomial number ℓ of classical queries to its own oracle, we have that*

$$\left|\Pr\left[\mathcal{D}(1^\kappa)^{\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot)} = 1\right] - \Pr\left[\mathcal{D}(1^\kappa)^{\mathcal{O}^c_{priv}(\cdot), \mathcal{P}(\cdot)^{\mathcal{O}^c_{pub}(\cdot)}} = 1\right]\right| \geq \frac{1}{72}.$$

Note that Theorem 2 directly rules out the trivial distinguisher which runs the Grover algorithm [18,10] with sub-exponentially many, i.e., $O(2^{n/2})$ or $O(2^{m/3})$, quantum queries, as it is useless for most schemes that are based on quantum sub-exponential time (or even polynomial time) hardness assumptions. Actually, we will construct a QPT distinguisher $\mathcal{D}$ making at most two quantum queries.

Before proceeding to prove Theorem 2, we first give an information versus disturbance lemma, which might be of independent interest. Formally, let H^1 be the one-dimensional Hadamard transformation H (i.e., $H\left|0\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle)$ and $H\left|1\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$), and let H^0 be the identity transformation Id. Then, for any $s \in \{0,1\}^n$, we can define a quantum operation $H^s \stackrel{\text{def}}{=} H^{s_1} \otimes H^{s_2} \cdots \otimes H^{s_n}$ on n qubits: for any bit string $x \in \{0,1\}^n$ and $s \in \{0,1\}^n$, denote $\left|x\right\rangle_s \stackrel{\text{def}}{=} H^s \left|x\right\rangle$ as the quantum encoding $\left|x\right\rangle_s = \left|x_1\right\rangle_{s_1} \cdots \left|x_n\right\rangle_{s_n}$ of $x = x_1 \dots x_n$ by using bases indicated by $s = s_1 \dots s_n$, where $\left|x_i\right\rangle_{s_i} \stackrel{\text{def}}{=} \left|x_i\right\rangle$ if $s_i = 0$, and $\left|x_i\right\rangle_{s_i} \stackrel{\text{def}}{=} H\left|x_i\right\rangle$ otherwise. Clearly, we always have $H^s \left|x\right\rangle_s = \left|x\right\rangle$.

Lemma 3 (Information versus Disturbance). *Let x, s be two random variables uniformly distributed over $\{0,1\}^n$. Let $\mathcal{P}$ be any (unbounded) quantum algorithm which takes $\left|x\right\rangle_s$ as input, outputs a bit string $e \in \{0,1\}^{\mathrm{poly}(n)}$ and a quantum state $\left|\zeta\right\rangle$ of n-qubit, i.e., $(e, \left|\zeta\right\rangle) \leftarrow \mathcal{P}(\left|x\right\rangle_s)$. Let $x' \in \{0,1\}^n$ be a bit string obtained by first applying H^s on the state $\left|\zeta\right\rangle$, and then measuring the state $H^s\left|\zeta\right\rangle$ in the computational basis. Let $z = x \oplus x'$, and let $I(x,e)$ be the mutual information between x and e. Then, we have that*

$$I(x,e) \leq n(\alpha + \frac{1}{\alpha} \sum_{\mathsf{hw}(z) \geq 1} \Pr[z])$$

holds for any $\alpha > 0$, where $\mathsf{hw}(z)$ denotes the hamming-weight of $z \in \{0,1\}^n$.

Note that if $\left|\zeta\right\rangle = \left|x\right\rangle_s$, we always have $z = 0_n$ (i.e., $\mathsf{hw}(z) = 0$). The above lemma is basically a quantitative version of the claim that any attempt made by $\mathcal{P}$ to obtain useful information x from the state $\left|x\right\rangle_s$ will necessarily disturb the state. The proof of Lemma 3 is essentially adapted from the security proof [6] of BB84 protocol. Actually, one can imagine that there are two users "Alice" and "Bob" involved in the interactions with $\mathcal{P}$, where "Alice" prepares the input state to $\mathcal{P}$, and "Bob" checks if the input state is disturbed by $\mathcal{P}$. Moreover, the behaviors of "Alice" and "Bob" are similar to those of the real users in the used-bits-BB84 protocol (before the information reconciliation procedure) [6]. The major difference is that in our case "Alice" and "Bob" share the same random $x, s \in \{0,1\}^n$ (i.e., they will not leak any classical information of x, s to the "eavesdropper" $\mathcal{P}$), which allows us to obtain a "clean" information versus disturbance lemma (note that the eavesdropper in the BB84 protocol can also obtain information from a classical channel, which makes a similar result [6] for the BB84 protocol more complex and unsuitable for our purpose).

Proof. Without loss of generality, we can assume that $\mathcal{P}$ works as follows: given an input state $\left|x\right\rangle_s$ in the input register, it first prepares an ancillary register A

in a known state $|0\rangle_A$, and performs a unitary transformation U on the state

$$|0\rangle_A |x\rangle_s .$$

The resulting state $U|0\rangle_A |x\rangle_s$ can be expressed in a unique way as a sum

$$U|0\rangle_A |x\rangle_s = \sum_{\hat{x}} |U_{x,\hat{x}}\rangle_s |\hat{x}\rangle_s ,$$

where $|U_{x,\hat{x}}\rangle_s = {}_s\langle\hat{x}|U|0\rangle_A |x\rangle_s$ are non-normalized states of $\mathcal{P}$'s register A. Then, it performs some measurement M on the state in his ancillary register to obtain a classical information e and a "disturbed" state $|\zeta\rangle$ in the input register. Finally, it outputs e and $|\zeta\rangle$.

Let x' be the measurement outcome of $H^s|\zeta\rangle$. Let $z = x \oplus x'$. Then, we have

$$\Pr[z] = \sum_{x,s} \Pr[x,s] \Pr[z|x,s] = \frac{1}{2^{2n}} \sum_{x,s} \langle U_{x,x\oplus z}|U_{x,x\oplus z}\rangle_s . \tag{1}$$

Note that $\mathcal{P}$'s state in the ancillary register A after performing the unitary transformation U is fully determined by tracing-out the subsystem $|\hat{x}\rangle_s$ from the state $\sum_{\hat{x}} |U_{x,\hat{x}}\rangle_s |\hat{x}\rangle_s$, and it is

$$\rho_s^x = \sum_{\hat{x}} |U_{x,\hat{x}}\rangle_s {}_s\langle U_{x,\hat{x}}|.$$

We can purify the above state while giving more information to $\mathcal{P}$ by assuming she keeps the pure state

$$|\phi_x\rangle_s = \sum_{\hat{x}} |U_{x,\hat{x}}\rangle_s |x \oplus \hat{x}\rangle_s .$$

We have that $\rho_s^x = |\phi_x\rangle_s {}_s\langle\phi_x|$.

As shown in [6], it is sufficient to consider the symmetric attack, which is irrelevant to and will not be affected by the choices of $x \in \{0,1\}^n$ and $s \in \{0,1\}^n$. In fact, Biham et al. [6] showed that for any attack $\{U, M\}$, one can define a symmetric attack $\{U^{sym}, M^{sym}\}$ which is at least as good (for $\mathcal{P}$) as the original attack $\{U, M\}$. In particular, compared to the original attack, the symmetric one does not decrease the information obtained by $\mathcal{P}$ while keeping the same average error-rate caused by $\mathcal{P}$. For symmetric attack $\{U, M\}$, we have the following useful facts [6, Lemma 3.5]:

- $\langle U_{x,x\oplus z}|U_{x\oplus t,x\oplus z\oplus t}\rangle_s$ is independent of x;
- $\sum_{\hat{x}} \langle U_{x,\hat{x}}|U_{x\oplus t,\hat{x}\oplus t}\rangle_s$ is independent of x.

Define $\Phi_{t,s} \overset{\text{def}}{=} \langle\phi_x|\phi_{x\oplus t}\rangle_s = \sum_{\hat{x}} \langle U_{x,\hat{x}}|U_{x\oplus t,\hat{x}\oplus t}\rangle_s$, which is independent of x. Define

$$|\gamma_x\rangle_s \overset{\text{def}}{=} \frac{1}{2^n} \sum_t (-1)^{x\cdot t} |\phi_t\rangle_s , \quad d_{x,s}^2 \overset{\text{def}}{=} \langle\gamma_x|\gamma_x\rangle_s \text{ and } \hat{\gamma}_x \overset{\text{def}}{=} \gamma_x/d_{x,s},$$

where $x{\cdot}t = (x_1{\cdot}t_1)\oplus\cdots\oplus(x_n{\cdot}t_n) \in \{0,1\}$ for any bit vector $x = (x_1,\dots,x_n), t = (t_1,\dots,t_n) \in \{0,1\}^n$ and $d_{x,s} > 0$. We now slightly deviate from the main proof by showing several useful equations which will be used latter.

Firstly, by the fact that

$$\frac{1}{2^n}\sum_t (-1)^{(x\oplus\hat{x})\cdot t} = \begin{cases} 0,\ x \neq \hat{x}; \\ 1,\ \text{otherwise}, \end{cases}$$

we can rewrite

$$\left|\phi_t\right\rangle_s = \sum_{\hat{x}} (-1)^{\hat{x}\cdot t} \left|\gamma_{\hat{x}}\right\rangle_s = \sum_{\hat{x}} (-1)^{\hat{x}\cdot t} d_{\hat{x},s} \left|\hat{\gamma}_{\hat{x}}\right\rangle_s \tag{2}$$

Secondly, we can rewrite the equation $\left\langle \gamma_x | \gamma_x \right\rangle_s = \frac{1}{2^{2n}} \sum_{t,\hat{t}} (-1)^{x\cdot(t\oplus\hat{t})} \left\langle \phi_t | \phi_{\hat{t}} \right\rangle_s$ as $\left\langle \gamma_x | \gamma_x \right\rangle_s = \frac{1}{2^{2n}} \sum_{t,\hat{t}} (-1)^{x\cdot\hat{t}} \left\langle \phi_t \middle| \phi_{t\oplus\hat{t}} \right\rangle_s$ by using variable substitution, which in turn implies that

$$d_{x,s}^2 = \frac{1}{2^{2n}} \sum_{t,\hat{t}} (-1)^{x\cdot\hat{t}} \left\langle \phi_t \middle| \phi_{t\oplus\hat{t}} \right\rangle_s = \frac{1}{2^{2n}} \sum_{t,\hat{t},\hat{x}} (-1)^{x\cdot\hat{t}} \left\langle U_{t,\hat{x}} \middle| U_{t\oplus\hat{t},\hat{x}\oplus\hat{t}} \right\rangle_s. \tag{3}$$

Thirdly, for any $x \neq \hat{x}$, we have

$$\begin{aligned} \left\langle \gamma_x | \gamma_{\hat{x}} \right\rangle_s &= \tfrac{1}{2^{2n}} \textstyle\sum_{t,\hat{t}} (-1)^{x\cdot t} (-1)^{\hat{x}\cdot\hat{t}} \left\langle \phi_t | \phi_{\hat{t}} \right\rangle_s \\ &= \tfrac{1}{2^{2n}} \textstyle\sum_{t,\hat{t}} (-1)^{(x\oplus\hat{x})\cdot t} (-1)^{\hat{x}\cdot\hat{t}} \left\langle \phi_t \middle| \phi_{t\oplus\hat{t}} \right\rangle_s \\ &= \tfrac{1}{2^{2n}} \textstyle\sum_{t,\hat{t}} (-1)^{(x\oplus\hat{x})\cdot t} (-1)^{\hat{x}\cdot\hat{t}} \varPhi_{\hat{t},s} \\ &= \tfrac{1}{2^{2n}} \left(\textstyle\sum_t (-1)^{(x\oplus\hat{x})\cdot t}\right) \textstyle\sum_{\hat{t}} (-1)^{\hat{x}\cdot\hat{t}} \varPhi_{\hat{t},s}. \end{aligned}$$

Since $\sum_t (-1)^{(x\oplus\hat{x})\cdot t} = 0$, we have that

$$\left\langle \gamma_x | \gamma_{\hat{x}} \right\rangle_s = 0 \tag{4}$$

holds for any $x \neq \hat{x}$.

Fourthly, by the fact that $\sum_x (-1)^{x\cdot\hat{t}} = 0$ for any $\hat{t} \neq 0_n$, and that $|\phi_t\rangle$ is a pure state (which implies $\left\langle \phi_t | \phi_t \right\rangle_s = 1$ for any $t \in \{0,1\}^n$), we have that

$$\begin{aligned} \textstyle\sum_x d_{x,s}^2 &= \textstyle\sum_x \tfrac{1}{2^{2n}} \textstyle\sum_{t,\hat{t}} (-1)^{x\cdot\hat{t}} \left\langle \phi_t \middle| \phi_{t\oplus\hat{t}} \right\rangle_s \\ &= \tfrac{1}{2^{2n}} \textstyle\sum_{t,\hat{t}} \left(\textstyle\sum_x (-1)^{x\cdot\hat{t}}\right) \left\langle \phi_t \middle| \phi_{t\oplus\hat{t}} \right\rangle_s \\ &= \tfrac{1}{2^n} \textstyle\sum_t \left\langle \phi_t | \phi_t \right\rangle_s \\ &= 1. \end{aligned} \tag{5}$$

Next, we return back to the main proof. Let $I(x,e)$ be the mutual information of x and e. Let $I(x_i,e)$ be the mutual information between the i-th bit x_i of x

and e, where $i \in \{1, \ldots, n\}$. Since x is a random variable uniformly distributed over $\{0,1\}^n$, we have that

$$I(x,e) \leq \sum_i I(x_i, e). \tag{6}$$

Let $v_i \in \{0,1\}^n$ be the bit string whose j-th bit is nonzero if and only if $j = i$, we have $x_i = v_i \cdot x \in \{0,1\}$. For any bit $a \in \{0,1\}$, define

$$\begin{aligned}
\rho_a(v_i) &= \tfrac{1}{2^{2n-1}} \textstyle\sum_s \sum_{v_i \cdot x = a} \rho_s^x \\
&= \tfrac{1}{2^{2n-1}} \textstyle\sum_s \sum_{v_i \cdot x = a} |\phi_x\rangle_s {}_s\langle\phi_x| \\
&= \tfrac{1}{2^{2n-1}} \textstyle\sum_{s,t,\hat{t}} \sum_{v_i \cdot x = a} (-1)^{(t \oplus \hat{t}) \cdot x} d_{t,s} d_{\hat{t},s} |\hat{\gamma}_t\rangle_s {}_s\langle\hat{\gamma}_{\hat{t}}|,
\end{aligned}$$

where the last equation is due to Equation (2). Note that in order to distinguish the i-th bit x_i of x, $\mathcal{P}$ has to distinguish the two states $\rho_0(v_i)$ and $\rho_1(v_i)$. A good measure for the distinguishability of $\rho_0(v_i)$ and $\rho_1(v_i)$ is the optimal mutual information that one could get if one needs to guess the bit a by performing an optimal measurement to distinguish between the two density matrices, when the two are given with equal probability of half. This information is called the Shannon Distinguishability [14], denote as $SD(\rho_0(v_i), \rho_1(v_i))$. Due to the optimality of SD, we get

$$I(x_i, e) \leq SD(\rho_0(v_i), \rho_1(v_i)),$$

which is then bounded by the trace norm of $\rho_0(v_i) - \rho_1(v_i)$ [14]. Since

$$\begin{aligned}
\rho_0(v_i) - \rho_1(v_i) &= (-1)^0 \rho_0(v_i) + (-1)^1 \rho_1(v_i) \\
&= \tfrac{1}{2^{2n-1}} \textstyle\sum_{s,x,t,\hat{t}} (-1)^{(t \oplus \hat{t} \oplus v_i) \cdot x} d_{t,s} d_{\hat{t},s} |\hat{\gamma}_t\rangle_s {}_s\langle\hat{\gamma}_{\hat{t}}| \\
&= \tfrac{1}{2^{2n-1}} \textstyle\sum_{s,t,\hat{t}} \left(\sum_x (-1)^{(t \oplus \hat{t} \oplus v_i) \cdot x} \right) d_{t,s} d_{\hat{t},s} |\hat{\gamma}_t\rangle_s {}_s\langle\hat{\gamma}_{\hat{t}}| \\
&= \tfrac{1}{2^{n-1}} \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} |\hat{\gamma}_t\rangle_s {}_s\langle\hat{\gamma}_{t \oplus v_i}|,
\end{aligned}$$

we have

$$\begin{aligned}
SD(\rho_0(v_i), \rho_1(v_i)) &\leq \tfrac{1}{2} Tr |\rho_0(v_i) - \rho_1(v_i)| \\
&\leq \tfrac{1}{2^n} Tr \left| \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} |\hat{\gamma}_t\rangle_s {}_s\langle\hat{\gamma}_{t \oplus v_i}| \right| \\
&= \tfrac{1}{2^{n+1}} Tr \left| \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} (|\hat{\gamma}_t\rangle_s {}_s\langle\hat{\gamma}_{t \oplus v_i}| + |\hat{\gamma}_{t \oplus v_i}\rangle_s \langle\hat{\gamma}_t|_s) \right| \\
&\leq \tfrac{1}{2^n} \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} (\tfrac{1}{2} Tr| \, |\hat{\gamma}_t\rangle_s \langle\hat{\gamma}_{t \oplus v_i}|_s + |\hat{\gamma}_{t \oplus v_i}\rangle_s {}_s\langle\hat{\gamma}_t| \, |) \\
&= \tfrac{1}{2^n} \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} \sqrt{1 - (\mathrm{Im}(\langle\hat{\gamma}_t|\hat{\gamma}_{t \oplus v_i}\rangle_s))^2} \\
&= \tfrac{1}{2^n} \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} \\
&= \tfrac{1}{2^n} \textstyle\sum_s \left(\sum_{\mathsf{hw}(t) \geq 1} d_{t,s} d_{t \oplus v_i, s} + \sum_{\mathsf{hw}(t) = 0} d_{t,s} d_{t \oplus v_i, s} \right) \\
&\leq \tfrac{1}{2^n} \textstyle\sum_s \left(\sum_{\mathsf{hw}(t) \geq 1} d_{t,s} d_{t \oplus v_i, s} + \sum_{\mathsf{hw}(t \oplus v_i) \geq 1} d_{t,s} d_{t \oplus v_i, s} \right),
\end{aligned}$$

where Im is the imaginary part, and the last third equality holds due to the fact that $\langle\hat{\gamma}_t|\hat{\gamma}_{t\oplus v_i}\rangle_s = 0$ by Equation (4). For any positive $\alpha > 0$, we have

$$\begin{aligned}SD(\rho_0(v_i),\rho_1(v_i)) &\le \tfrac{1}{2^n}\textstyle\sum_s \left(2\sum_{\mathsf{hw}(t)\ge 1} d_{t,s}d_{t\oplus v_i,s}\right)\\ &= \tfrac{1}{2^n}\textstyle\sum_s \left(\frac{1}{\alpha}\sum_{\mathsf{hw}(t)\ge 1} 2\alpha d_{t,s}d_{t\oplus v_i,s}\right)\\ &\le \tfrac{1}{2^n}\textstyle\sum_s \left(\frac{1}{\alpha}\sum_{\mathsf{hw}(t)\ge 1}(d_{t,s}^2+\alpha^2 d_{t\oplus v_i,s}^2)\right)\\ &= \tfrac{1}{2^n}\textstyle\sum_s \left(\frac{1}{\alpha}\sum_{\mathsf{hw}(t)\ge 1} d_{t,s}^2+\alpha\sum_{\mathsf{hw}(t)\ge 1} d_{t\oplus v_i,s}^2\right)\\ &\le \tfrac{1}{2^n}\textstyle\sum_s \left(\frac{1}{\alpha}\sum_{\mathsf{hw}(t)\ge 1} d_{t,s}^2+\alpha\right)\end{aligned}$$

The third inequality follows from the Cauchy-Schwarz inequality, and the last one holds because $\sum_{\mathsf{hw}(t)\ge 1} d_{t\oplus v_i,s}^2 \le \sum_t d_{t,s}^2 = 1$ by Equation (5). This means that

$$I(x_i,e) \le SD(\rho_0(v_i),\rho_1(v_i)) \le \alpha + \frac{1}{\alpha 2^n}\sum_s \sum_{\mathsf{hw}(t)\ge 1} d_{t,s}^2$$

holds for any $\alpha > 0$.

By Equation (6), we have

$$I(x,e) \le \sum_i I(x_i,e) \le n\left(\alpha + \frac{1}{\alpha 2^n}\sum_s \sum_{\mathsf{hw}(t)\ge 1} d_{t,s}^2\right) \tag{7}$$

We finish the proof by bounding $I(x,e)$ using $\Pr[z]$. By Equation (1), we have

$$\Pr[z] = \sum_{x,s}\Pr[x,s]\Pr[c|x,s] = \frac{1}{2^{2n}}\sum_{x,s}\langle U_{x,x\oplus z}|U_{x,x\oplus z}\rangle_s .$$

For any $s\in\{0,1\}^n$, let $\bar{s} = s\oplus 1_n \in \{0,1\}^n$ be the bit string obtained by flipping each bit of $s\in\{0,1\}^n$. Since the change of basis between s and $\bar{s}$ is expressed by $|x'\rangle_{\bar{s}} = \sum_x 2^{-n/2}(-1)^{x'\cdot x}|x\rangle_s$ and $|x\rangle_s = \sum_{x'} 2^{-n/2}(-1)^{x\cdot x'}|x'\rangle_{\bar{s}}$, we have that

$$|U_{x',\hat{x}'}\rangle_{\bar{s}} = \frac{1}{2^n}\sum_{x,\hat{x}}(-1)^{x'\cdot x}(-1)^{\hat{x}'\cdot\hat{x}}|U_{x,\hat{x}}\rangle_s .$$

This implies that

$$\begin{aligned}\Pr[z] &= \tfrac{1}{2^{2n}}\textstyle\sum_{t,\bar{s}}\langle U_{t,t\oplus z}|U_{t,t\oplus z}\rangle_{\bar{s}}\\ &= \tfrac{1}{2^{4n}}\textstyle\sum_{t,\bar{s}}\sum_{x,\hat{x}}\sum_{x',\hat{x}'}(-1)^{t\cdot x}(-1)^{(t\oplus z)\cdot\hat{x}}(-1)^{t\cdot x'}(-1)^{(t\oplus z)\cdot\hat{x}'}\langle U_{x,\hat{x}}|U_{x',\hat{x}'}\rangle_s\\ &= \tfrac{1}{2^{4n}}\textstyle\sum_{\bar{s},x,\hat{x},x',\hat{x}'}\left(\sum_t(-1)^{t\cdot(x\oplus x'\oplus\hat{x}\oplus\hat{x}')}\right)(-1)^{z\cdot(\hat{x}\oplus\hat{x}')}\langle U_{x,\hat{x}}|U_{x',\hat{x}'}\rangle_s .\end{aligned}$$

Since $\sum_t(-1)^{t\cdot(x\oplus x'\oplus\hat{x}\oplus\hat{x}')} \ne 0 \Leftrightarrow x\oplus x'\oplus\hat{x}\oplus\hat{x}' = 0$, by setting $\hat{t} = x\oplus x' = \hat{x}\oplus\hat{x}'$ and using Equation (3), we have that

$$\Pr[z] = \frac{1}{2^{3n}}\sum_{\bar{s},x,\hat{x},\hat{t}}(-1)^{z\cdot\hat{t}}\langle U_{x,\hat{x}}|U_{x\oplus\hat{t},\hat{x}\oplus\hat{t}}\rangle_s = \frac{1}{2^n}\sum_{\bar{s}} d_{z,s}^2 = \frac{1}{2^n}\sum_s d_{z,s}^2. \tag{8}$$

Combining Equations (7) and (8), we obtain

$$I(x,e) \leq n\left(\alpha + \frac{1}{\alpha}\sum_{\mathsf{hw}(z)\geq 1}\Pr[z]\right).$$

This completes the proof. $\square$

Now, we are ready to prove Theorem 2. Technically, our distinguisher $\mathcal{D}$ will first send a quantum query such that 1) it is a superposition of a random subset $X \subseteq \{0,1\}^n$ of classical queries having the same pattern; 2) any attempt of $\mathcal{P}$ to obtain the information about X will necessarily disturb the query state; and 3) it is infeasible for $\mathcal{P}$ to make a correctly patterned classical query $x \in X$ without knowing sufficient information about X. After receiving a response from $\mathcal{P}$, $\mathcal{D}$ will flip a random bit to either check the consistency of the response (i.e., $\mathcal{P}$ has to make a correctly patterned classical query $x \in X$), or detect the disturbance of the query state (i.e., $\mathcal{P}$ cannot perform operations such as measurements that are not unitary) by using another quantum query. Clearly, $\mathcal{D}$ wins if it obtains a wrong response from the "oracle" or it detects that the "oracle" is not unitary (since both will never happen if it is given access to a real QRO).

Proof. Note that the distinguisher $\mathcal{D}$ is given access to an oracle $\tilde{\mathcal{O}} = (\tilde{\mathcal{O}}_{priv}(\cdot), \tilde{\mathcal{O}}_{pub}(\cdot))$, which is either a real QRO (i.e., $\tilde{\mathcal{O}} = \mathcal{O}^q = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot)) \xleftarrow{\$} \mathcal{F}_{n,m}$) or a simulated one (i.e., $\tilde{\mathcal{O}} = (\mathcal{O}^c_{priv}(\cdot), \mathcal{P}(\cdot)^{\mathcal{O}^c_{pub}(\cdot)})$) for some algorithm $\mathcal{P}(\cdot)$ and CRO $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot)) \xleftarrow{\$} \mathcal{F}_{n,m}$. Moreover, given a query $x \in \{0,1\}^n$, the private interface $\tilde{\mathcal{O}}_{priv}(\cdot)$ will return a bit string $y \in \{0,1\}^m$, while given a quantum query $|x,z\rangle$, the public interface $\tilde{\mathcal{O}}_{pub}(\cdot)$ will return a state $|\psi\rangle$.

For any bit string $s = s_1 \dots s_n \in \{0,1\}^n$, define an associated set $S = \{i_1, \dots, i_{\mathsf{hw}(s)}\}$ such that $i_j \in S$ if and only if the i_j-th bit s_{i_j} of s is 1, where $\mathsf{hw}(s)$ is the Hamming weight of s and $i_1 < \cdots < i_{\mathsf{hw}(s)}$. Let $\bar{s} \in \{0,1\}^n$ be the bit string obtained by flipping each bit of s, i.e., $\bar{s} = s \oplus 1_n$. Denote $x^s = x_{i_1} \dots x_{i_{\mathsf{hw}(s)}} \in \{0,1\}^{\mathsf{hw}(s)}$ as the $\mathsf{hw}(s)$-bit substring of $x \in \{0,1\}^n$ indexed by S. Corresponding, denote $x^{\bar{s}} \in \{0,1\}^{\mathsf{hw}(\bar{s})} = \{0,1\}^{n-\mathsf{hw}(s)}$ as the substring of $x \in \{0,1\}^n$ by deleting the bits indexed by S. Now, we are ready to give the description of the distinguisher $\mathcal{D}$, which will make at most two quantum queries to $\tilde{\mathcal{O}}(\cdot)$ via the public interface $\tilde{\mathcal{O}}_{pub}(\cdot)$, and one classical query to $\tilde{\mathcal{O}}(\cdot)$ via the private interface $\tilde{\mathcal{O}}_{priv}(\cdot)$. Specifically, $\mathcal{D}$ works as follows:

1. (**Create a Random Pattern**) Uniformly choose bit-strings $x, s \xleftarrow{\$} \{0,1\}^n$ at random, and prepare a quantum state $|x\rangle |0_m\rangle$ of $n+m$ qubits. Then, apply H^s on the state $|x\rangle$ to obtain $|x\rangle_s = H^s |x\rangle$. Let $X \subseteq \{0,1\}^n$ be the set $X = \{\tilde{x} \in \{0,1\}^n : \tilde{x}^{\bar{s}} = x^{\bar{s}}\}$ determined by (x,s), we can rewrite

$$|x\rangle_s = H^s |x\rangle = \frac{1}{\sqrt{2^{\mathsf{hw}(s)}}}\sum_{\tilde{x}\in X}(-1)^{x^s \cdot \tilde{x}^s}|\tilde{x}\rangle.$$

.

2. Send a quantum query $|\phi\rangle = |x\rangle_s |0_m\rangle$ to the oracle $\tilde{\mathcal{O}}(\cdot)$ via the public interface $\tilde{\mathcal{O}}_{pub}(\cdot)$, and obtain a state $|\psi\rangle$ consisting of $n+m$ qubits;
3. Pick a bit $b \xleftarrow{\$} \{0,1\}$ at random, and do the following computations:
   3.1 If $b = 0$, measure the state $|\psi\rangle$ to obtain a pair of $\tilde{x} \in \{0,1\}^n$ and $\tilde{y} \in \{0,1\}^m$. If $\tilde{x}^{\bar{s}} \neq x^{\bar{s}}$ (i.e., $\tilde{x} \notin X$), output 0 and abort. Otherwise, send a query $\tilde{x}$ to the oracle $\tilde{\mathcal{O}}(\cdot)$ via the private interface $\tilde{\mathcal{O}}_{priv}(\cdot)$, and obtain a result $\tilde{y}'$. If $\tilde{y}' \neq \tilde{y}$, output 0 and abort.
   3.2 Else if $b = 1$, send a quantum query with state $|\psi\rangle$ to the oracle $\tilde{\mathcal{O}}(\cdot)$ via the public interface $\tilde{\mathcal{O}}_{pub}(\cdot)$, and obtain a state $|\zeta\rangle$ of $n+m$ qubits. Apply H^s to the first n-qubit of the state $|\zeta\rangle$, and obtain a state $|\zeta'\rangle$. Measure the state $|\zeta'\rangle$ to obtain a pair of $\hat{x} \in \{0,1\}^n$ and $\hat{y} \in \{0,1\}^m$. If $\hat{x} \neq x$ or $\hat{y} \neq 0_m$, output 0 and abort.
4. If no abort has happened in step 3, output 1 and abort.

Now, it is enough to show the following two claims: 1) if $\tilde{\mathcal{O}}$ is a real QRO, namely, $\tilde{\mathcal{O}} = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot))$, $\mathcal{D}$ will always output 1; and 2) if $\tilde{\mathcal{O}}$ is a simulated one, namely, $\tilde{\mathcal{O}} = (\mathcal{O}^c_{priv}(\cdot), \mathcal{P}(\cdot)^{\mathcal{O}^c_{pub}(\cdot)})$ for some CRO $\mathcal{O}^c = (\mathcal{O}^c_{priv}(\cdot), \mathcal{O}^c_{pub}(\cdot)) \xleftarrow{\$} \mathcal{F}_{n,m}$ and algorithm $\mathcal{P}(\cdot)$ making only a polynomial number ℓ of classical queries to its own oracle, we have

$$\Pr\left[\mathcal{D}(1^\kappa)^{\mathcal{O}^c_{priv}(\cdot), \mathcal{P}(\cdot)^{\mathcal{O}^c_{pub}(\cdot)}} = 1\right] \leq \frac{71}{72}.$$

We first consider the case that $\tilde{\mathcal{O}}$ is a real QRO, namely, $\tilde{\mathcal{O}} = \mathcal{O}^q = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot)) \xleftarrow{\$} \mathcal{F}_{n,m}$. Note that $\mathcal{D}$ will first send a quantum query with state $|\phi\rangle$ to $\tilde{\mathcal{O}}(\cdot) = \mathcal{O}^q(\cdot)$ via the public interface $\mathcal{O}^q_{pub}(\cdot)$ in step 2, and obtain a state

$$|\psi\rangle = \frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\tilde{x} \in X} (-1)^{x^s \cdot \tilde{x}^s} |\tilde{x}, \mathcal{O}^q(\tilde{x})\rangle .$$

If $b = 0$, measuring the state $|\psi\rangle$ will always obtain a pair of $\tilde{x} \in \{0,1\}^n$ and $\tilde{y} \in \{0,1\}^m$ such that $\tilde{x} \in X$ (i.e., $\tilde{x}^{\bar{s}} = x^{\bar{s}}$) and $\tilde{y} = \mathcal{O}^q(\tilde{x})$, which means that $\mathcal{D}$ will never output 0 in step 3.1. As for $b = 1$ in step 3.2, $\mathcal{D}$ will first send back $|\psi\rangle$ to the oracle $\tilde{\mathcal{O}}(\cdot) = \mathcal{O}^q(\cdot)$ via the public interface $\mathcal{O}^q_{pub}(\cdot)$, and obtain a state

$$|\zeta\rangle = \frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\tilde{x} \in X} (-1)^{x^s \cdot \tilde{x}^s} |\tilde{x}\rangle |0_m\rangle .$$

Then, it will apply $H^s \otimes \mathsf{Id}_m$ on the state $|\zeta\rangle$, and obtain a state $|\zeta'\rangle = |x\rangle |0_m\rangle$. Thus, measuring the state $|\zeta'\rangle$ will result in a pair of $\hat{x} = x$ and $\hat{y} = 0_m$. This shows that $\mathcal{D}$ will always output 1 in both cases if $\tilde{\mathcal{O}}$ is a real QRO.

We now consider the case that $\tilde{\mathcal{O}}$ is a simulated QRO, namely, $\tilde{\mathcal{O}} = (\mathcal{O}^c_{priv}(\cdot), \mathcal{P}(\cdot)^{\mathcal{O}^c_{pub}(\cdot)})$. Note that after obtaining the response $|\psi\rangle$ from $\mathcal{P}(\cdot)^{\mathcal{O}^c_{pub}(\cdot)}$ to the query $|\phi\rangle$, the distinguisher $\mathcal{D}(\cdot)$ will first uniformly pick a bit $b \xleftarrow{\$} \{0,1\}$ at random, and then perform different checks depending on the value of b. Let ϑ_0

and ϑ_1 be the probabilities that $|\psi\rangle$ passes the checks for $b = 0$ and $b = 1$, respectively. Then, the probability that $\mathcal{D}(\cdot)$ will output 1 is $(\vartheta_0 + \vartheta_1)/2$.

Note that if $\vartheta_1 \leq 35/36$, the proof is already finished as $(\vartheta_0+\vartheta_1)/2 \leq 71/72$. Thus, it suffices to consider that $\vartheta_1 > 35/36$. Let $z = \hat{x} \oplus x$ be the difference between the bit string $\hat{x} \in \{0,1\}^n$ obtained in step 3.2 and the random string $x \in \{0,1\}^n$ chosen in step 1. By definition, we have that $\vartheta_1 \leq \Pr[z = 0_n]$. We now give a bound on ϑ_0. Let E be the event that $\mathcal{P}$ makes a classical query $\tilde{x} \in \{0,1\}^n$ such that $\tilde{x}^{\bar{s}} = x^{\bar{s}}$. Note that if E does not happen, the probability that $\mathcal{P}$ passes the check in step 3.1 is at most $1/2^m$. Thus, we have $\vartheta_0 \leq 1/2^m + \Pr[E]$. Moreover, let $\delta = 1/2 - \nu$ for some $0 < \nu < 1/2$, by the law of total probability we have that $\Pr[E]$

$$= \Pr[E\,|\mathsf{hw}(\bar{s}) \leq \delta n] \cdot \Pr[\mathsf{hw}(\bar{s}) \leq \delta n] + \Pr[E\,|\mathsf{hw}(\bar{s}) > \delta n] \cdot \Pr[\mathsf{hw}(\bar{s}) > \delta n]$$
$$\leq \Pr[\mathsf{hw}(\bar{s}) \leq \delta n] + \Pr[E|\mathsf{hw}(\bar{s}) > \delta n].$$

Since s is uniformly chosen from $\{0,1\}^n$, we have that $\Pr[\mathsf{hw}(\bar{s}) \leq \delta n] \leq e^{-n\nu^2}$ by the Chernoff Bounds. Furthermore, let e be the classical information about x that $\mathcal{P}$ obtains from the interactions, and let $I(x,e)$ be the mutual information between x and e. Then, we have that $I(x^{\bar{s}}, e) \leq I(x,e)$, and

$$\Pr[E\,|\mathsf{hw}(\bar{s}) > \delta n] \leq \frac{\ell}{2^{\mathsf{hw}(\bar{s}) - I(x^{\bar{s}},e)}} \leq \frac{\ell}{2^{\delta n - I(x,e)}},$$

where $\ell = \mathrm{poly}(\kappa)$ is the number of classical queries made by $\mathcal{P}$ (since $I(x,e)$ bounds the information $\mathcal{P}$ obtained about x from above, see Section B). Thus, we have that

$$\vartheta_0 \leq \frac{1}{2^m} + e^{-n\nu^2} + \frac{\ell}{2^{\delta n - I(x,e)}}.$$

Note that if we only care about how much information $\mathcal{P}$ obtains about x from the input state $|\phi\rangle$, we can simplify the analysis by temporarily treating the output register of the query (which is initialized with zeros) as a part of the ancillary register of $\mathcal{P}$ and omit the check of $\hat{y}$. Since the choice of b is random and independent from x and s, the algorithm $\mathcal{P}$ cannot obtain more information about (x,s) when $b = 0$. This means that we can use Lemma 3 to establish a connection between $I(x,e)$ and ϑ_1 (because in this simplified case, the input to $\mathcal{P}$ and the check in $b = 1$ are essentially the same to the that in Lemma 3):

$$I(x,e) \leq n(\alpha + \frac{1}{\alpha} \sum_{\mathsf{hw}(z) \geq 1} \Pr[z]) = n(\alpha + \frac{1}{\alpha}(1 - \Pr[z = 0_n])) \leq n(\alpha + \frac{1}{\alpha}(1 - \vartheta_1)),$$

where $\alpha > 0$ is an arbitrary real. Since $\vartheta_1 > 35/36$, by setting $\alpha = 1/6$ we have $I(x,e) \leq n(\alpha + (1 - \vartheta_1)/\alpha) \leq n/3$. Moreover, by setting $\nu = 1/8$ we have that

$$\vartheta_0 \leq \frac{1}{2^m} + e^{-n\nu^2} + \frac{\ell}{2^{\delta n - I(x,e)}} \leq \frac{1}{2^m} + e^{-n/64} + \frac{\ell}{2^{n/24}} \leq \frac{1}{2} + \mathrm{negl}(\kappa)$$

holds for sufficiently large $n = \omega(\log \kappa)$. Thus, for $\vartheta_1 > 35/36$, we also have

$$\Pr\Big[\mathcal{D}(1^\kappa)^{\mathcal{O}_{priv}(\cdot),\mathcal{P}(\cdot)^{\mathcal{O}_{pub}(\cdot)}} = 1\Big] = \frac{\vartheta_0 + \vartheta_1}{2} \leq \frac{3}{4} + \mathrm{negl}(\kappa) \leq \frac{71}{72}$$

for sufficiently large $n = \omega(\log \kappa)$. This completes the proof. $\square$

Remark 1 (On the Implications of Theorem 2). First, one can already use a tighter bound $1/68 > 1/72$ for a reasonable choice of $(n, m, \ell) = (256, 128, 2^{32})$. Second, by running the above distinguisher $\mathcal{D}$ a polynomial number of times, one can obtain an algorithm $\mathcal{D}'$ which always outputs 1 when it is in the QROM, and outputs 0 with probability negligibly close to 1 when it is in the ROM. Third, by using algorithm $\mathcal{D}'$ as a sub-routine, one can construct an artificial system $\mathcal{C}'$ from another system $\mathcal{C}$ (e.g., a signature scheme) that is secure in the ROM, such that $\mathcal{C}'$ first runs algorithm $\mathcal{D}'$ and reveals some secret information (e.g., a signing key) of $\mathcal{C}$ if $\mathcal{D}'$ outputs 1, otherwise performs normally as $\mathcal{C}$ does if $\mathcal{D}'$ outputs 0 (just like the proof of Theorem 3). Clearly, $\mathcal{C}'$ is secure in the ROM (as $\mathcal{D}'$ will almost always output 0, and $\mathcal{C}$ is secure in the ROM), but it is insecure in the QROM (since $\mathcal{D}'$ will always output 1 in the QROM). Note that this artificial construction is actually implied by Lemma 2. We also note that the system $\mathcal{C}'$ will send out a polynomial number of random quantum queries (as $\mathcal{D}'$ will do), which may be unlikely to happen in a real system. But as an artificial system designed for separations, we believe it is non-trivial and meaningful.

3.2 On the Failure of the QROM

It is well-known that there exist schemes that are secure in the ROM, but for which any implementation of the RO results in insecure schemes [11,25,24]. Given that the QROM is stronger than the ROM, one might wonder if the QROM has some essentially different properties such that a security proof in the QROM could somehow guarantee the security of a cryptosytem in the real world. Now, we adapt the techniques in [24] to show that the answer is negative. Formally,

Theorem 3. *If there is an efficient post-quantum signature scheme which is secure in the QROM, then there exists another efficient signature scheme that is also secure in the QROM, but for which any implementation of the QRO using arbitrary efficiently computable function is insecure.*

The key observation behind our proof is that a QRO $\mathcal{O}^q(\cdot) \xleftarrow{\$} \mathcal{F}_{n,m}$ cannot be fully determined by a bit string of length less than $m \cdot 2^n$ (as each n-bit classical input will be mapped into a random m-bit string), but an efficiently computable function always has a succinct representation which can be uniquely encoded into a polynomial number of bits. Thus, any efficient implementation of the QRO actually allows the adversary to get a succinct representation of "the QRO", which is unlikely to happen for an adversary in the QROM.

For our purpose, we restrict our attention to post-quantum signature schemes where the (quantum) adversary is only allowed to make classical signing queries, and do not consider the case that the adversary can make quantum signing queries [9]. Now, we are ready to give the formal proof of Theorem 3.

Proof. Let $\mathcal{SIG} = (\mathsf{KeyGen}^{\mathcal{O}_1(\cdot)}(\cdot), \mathsf{Sign}^{\mathcal{O}_1(\cdot)}(\cdot,\cdot), \mathsf{Verify}^{\mathcal{O}_1(\cdot)}(\cdot,\cdot,\cdot))$ be an efficient post-quantum signature in the QROM, where $\mathcal{O}_1(\cdot) \xleftarrow{\$} \mathcal{F}_{n,m}$ is a QRO. Let κ be the security parameter. We modify $\mathcal{SIG}$ to obtain a new signature scheme $\widetilde{\mathcal{SIG}} = (\mathsf{KeyGen}^{\mathcal{O}_1(\cdot)}(\cdot), \widetilde{\mathsf{Sign}}^{\mathcal{O}_1(\cdot),\mathcal{O}_2(\cdot)}(\cdot,\cdot), \widetilde{\mathsf{Verify}}^{\mathcal{O}_1(\cdot)}(\cdot,\cdot,\cdot))$, which uses another QRO $\mathcal{O}_2(\cdot) :\xleftarrow{\$} \mathcal{F}_{\omega(\log\kappa),1}$. Specifically, given the signing key sk and a message $\mu \in \{0,1\}^*$ as inputs, the signing algorithm $\widetilde{\mathsf{Sign}}^{\mathcal{O}_1(\cdot),\mathcal{O}_2(\cdot)}(sk,\mu)$ first computes $b \leftarrow \mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu)$. If $b = 1$, the algorithm $\widetilde{\mathsf{Sign}}^{\mathcal{O}_1(\cdot),\mathcal{O}_2(\cdot)}(sk,\mu)$ directly outputs sk. Otherwise, it runs $\mathsf{Sign}^{\mathcal{O}_1(\cdot)}(sk,\mu)$ and outputs whatever $\mathsf{Sign}^{\mathcal{O}_1(\cdot)}(sk,\mu)$ returns. Accordingly, given the verification key vk, a message $\mu \in \{0,1\}^*$ and a signature σ as inputs, the algorithm $\widetilde{\mathsf{Verify}}^{\mathcal{O}_1(\cdot)}(vk,\mu,\sigma)$ returns 1 if σ is a valid signing key for vk, otherwise returns whatever $\mathsf{Verify}^{\mathcal{O}_1(\cdot)}(vk,\mu,\sigma)$ outputs.

Clearly, the signature scheme $\widetilde{\mathcal{SIG}}$ is correct. Moreover, if the sub-routine $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu)$ always outputs 0 for any message $\mu \in \{0,1\}^*$, then the new scheme $\widetilde{\mathcal{SIG}}$ essentially has the same functionality and security as the original scheme $\mathcal{SIG}$. We now construct a sub-routine $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\cdot)$ with two useful properties: 1) for a real QRO $\mathcal{O}_2(\cdot) \xleftarrow{\$} \mathcal{F}_{\omega(\log\kappa),1}$, the probability that there is a $\mu \in \{0,1\}^*$ such that $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu)$ outputs 1 is negligible; and 2) for any efficient implementation of $\mathcal{O}_2(\cdot)$, it is easy to find a special $\mu^* \in \{0,1\}^*$ such that $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu^*)$ always outputs 1. Formally, $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\cdot)$ works as follows:

1. Given an input $\mu \in \{0,1\}^*$, parse μ as a pair (π, t) consisting of an encoding of a universal turing machine π and a unary encoding of some integer t. Let ℓ_μ and ℓ_π be the length of μ and π, respectively (i.e., $t + \ell_\pi = \ell_\mu$). Then, do the following computation:
   (a) Set $\ell = 2\ell_\pi + \kappa$, and simulate at most t steps of the turing machine π to obtain $\alpha_1 = \pi(1), \cdots, \alpha_\ell = \pi(\ell)$. If the simulation stops before obtaining $\alpha_j = \pi(j)$ for some $j \le \ell$, set $\alpha_j = \alpha_{j+1} = \cdots = \alpha_\ell = 0$.
   (b) Send ℓ queries with $x = 1, \cdots, \ell$ to (the private interface of) the oracle $\mathcal{O}_2(\cdot)$, and obtain the responses $\beta_1 = \mathcal{O}_2(1), \ldots, \beta_\ell = \mathcal{O}_2(\ell)$;
   (c) If $\alpha_i = \beta_i$ for all $i \in \{1, \ldots, \ell\}$, output 1, else output 0.

Obviously, $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\cdot)$ runs in polynomial time. Moreover, for any fixed $\pi \in \{0,1\}^*$, the probability that $\pi(x) = \mathcal{O}_2(x)$ is at most $1/2$ over the random choice of $\mathcal{O}_2(\cdot) \xleftarrow{\$} \mathcal{F}_{\omega(\log\kappa),1}$. Since $\mathcal{D}(\mu)^{\mathcal{O}_2(\cdot)}$ will compare the values of $\mathcal{O}_2(\cdot)$ and $\pi(\cdot)$ at inputs $x \in \{1, \ldots, \ell\}$, for any fixed μ (and thus fixed π) the probability that $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu)$ outputs 1 is at most $1/2^\ell$. Taken over all possible π of a fixed length ℓ_π, the probability that $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu)$ outputs 1 is at most $2^{\ell_\pi}/2^\ell = 2^{-(\ell_\pi+\kappa)}$. Thus, the probability that there exists an $\mu \in \{0,1\}^*$ such that $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu)$ outputs 1 is at most $\sum_{\ell_\pi=1}^{\infty} 2^{-(\ell_\pi+\kappa)} \le 2^{-\kappa}$. Since this probability only depends on the choice of the QRO $\mathcal{O}_2(\cdot) \xleftarrow{\$} \mathcal{F}_{\omega(\log\kappa),1}$ and is independent of the choice of message $\mu \in \{0,1\}^*$, it is infeasible even for an adversary given quantum access to $\mathcal{O}_2(\cdot)$ to output a message $\mu \in \{0,1\}^*$ such that $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu)$ outputs 1 with non-negligible probability. This shows that the signature scheme $\widetilde{\mathcal{SIG}}$ is secure in the QROM.

Now, consider that $\mathcal{O}_2(\cdot)$ is implemented by using a function $f \in \mathcal{F}_{\omega(\log\kappa),1}$ which can be computed in polynomial time. Let π_f be an encoding of a universal turing machine that efficiently computes $f(\cdot)$. Let $\ell' = 2\ell_{\pi_f} + \kappa$, where ℓ_{π_f} is the length of π_f. Let t' be the maximal number of steps that the turing machine $\pi_f(\cdot)$ is required for computing $\pi_f(1), \ldots, \pi_f(\ell')$, and let μ^* be the concatenation of π_f and a unary encoding of t'. Since $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu^*)$ can compute $\pi_f(1), \ldots, \pi_f(\ell')$ in less than t' steps, and $\pi_f(x) = f(x) = \mathcal{O}_2(x)$ holds for any input $x \in \{0,1\}^{\omega(\log\kappa)}$, the sub-routine $\mathcal{D}^{\mathcal{O}_2(\cdot)}(\mu^*) = \mathcal{D}^{f(\cdot)}(\mu^*)$ will always output 1. Thus, for any efficiently computable function $f \in \mathcal{F}_{\omega(\log\kappa),1}$ that implements $\mathcal{O}_2(\cdot)$, there is an adversary which can obtain the signing key sk of $\widetilde{\mathcal{SIG}}$ by making a single signing query. This completes the proof. □

4 Committed-Programming Reductions

Let $\mathcal{O}(\cdot) \xleftarrow{\$} \mathcal{F}_{n,m}$ be a RO. A cryptographic problem equipped with the RO $\mathcal{O}(\cdot)$ consisting of three algorithms $\mathrm{P} = (\mathsf{IGen}^{\mathcal{O}(\cdot)}, \mathsf{Orcl}^{\mathcal{O}(\cdot)}, \mathsf{Vrfy}^{\mathcal{O}(\cdot)})$. Specifically, the instance generator $\mathsf{IGen}^{\mathcal{O}(\cdot)}$ is a polynomial time algorithm which takes as input the security parameter κ, outputs an instance x and a secret value s_x, i.e., $(x, s_x) \leftarrow \mathsf{IGen}^{\mathcal{O}(\cdot)}(1^\kappa)$. The stateful oracle algorithm $\mathsf{Orcl}^{\mathcal{O}(\cdot)}(s_x, \cdot)$ takes as input a query $q \in \{0,1\}^*$, returns a response r to q. The deterministic verification algorithm $\mathsf{Vrfy}^{\mathcal{O}(\cdot)}$ takes as inputs an instance x, a secret value s_x and a candidate solution w, returns 1 if w is a correct solution of x, and 0 otherwise. Finally, we note that the oracle $\mathsf{Orcl}^{\mathcal{O}(\cdot)}(s_x, \cdot)$ may accept many different types of queries (i.e., it is not necessarily restricted to a single functionality), and any polynomial number of ROs can also be easily simulated by using a single one, e.g., "a query (i, h) to a RO" can be treated as "a query h to the i-th RO".

Definition 2 (Hard Problem). *Let $\mathcal{O}(\cdot) \xleftarrow{\$} \mathcal{F}_{n,m}$ be a RO. A cryptographic problem $\mathrm{P} = (\mathsf{IGen}^{\mathcal{O}(\cdot)}, \mathsf{Orcl}^{\mathcal{O}(\cdot)}, \mathsf{Vrfy}^{\mathcal{O}(\cdot)})$ is said to be hard with respect to a threshold function $t(\cdot)$, if for all polynomial time algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ in the game with a challenger $\mathcal{C}$ who provides inputs to $\mathcal{A}$ and answers $\mathcal{A}$'s oracle queries is negligible in the security parameter κ:*

$$\mathrm{Adv}_{\mathrm{P},\mathcal{A}}(1^\kappa) = \Pr\left[\begin{array}{c} \mathcal{O}(\cdot) \xleftarrow{\$} \mathcal{F}_{n,m} \\ (x, s_x) \leftarrow \mathsf{IGen}^{\mathcal{O}(\cdot)}(1^\kappa) \\ w \leftarrow \mathcal{A}^{\mathsf{Orcl}^{\mathcal{O}(\cdot)}(s_x,\cdot), \mathcal{O}(\cdot)}(1^\kappa, x) \end{array} : \mathsf{Vrfy}^{\mathcal{O}(\cdot)}(x, s_x, w) = 1\right] - t(\kappa).$$

Typically, we have $t(\kappa) = 0$ for computational problems, and $t(\kappa) = 1/2$ for decisional problems. In this paper, we only consider *falsifiable* problems, which require that IGen, Orcl and Vrfy are polynomial time algorithms. Since a standard cryptographic problem can be seen as a problem $\mathrm{P} = (\mathsf{IGen}^{\mathcal{O}(\cdot)}, \mathsf{Orcl}^{\mathcal{O}(\cdot)}, \mathsf{Vrfy}^{\mathcal{O}(\cdot)})$ where the three algorithms do not use the RO $\mathcal{O}(\cdot)$, Definition 2 captures many cryptographic problems such as SIS and CCA/CPA-secure PKE (in the ROM).

A black-box (BB) reduction $\mathcal{R}$ from a problem P_1 to another problem P_2 (in the ROM) is a polynomial time oracle algorithm such that $\mathcal{R}^{\mathcal{A}}$ solves P_1 whenever $\mathcal{A}$ solves P_2. A BB-reduction $\mathcal{R}$ in the ROM can exploit various properties (e.g., programmability) of the RO in communication with $\mathcal{A}$. By restricting the programmability, Fischlin et al. [13] formalized three notions of BB-reductions in the ROM: fully-programming reduction (FPRed), randomly-programming reduction (RPRed) and non-programming reduction (NPRed). In brief, the FPRed formalizes the standard conception of BB-reductions in the ROM which have full control over the RO, while the NPRed considers the setting where all parties are given access to an external RO (i.e., non-programmable RO [25]) that is not controlled by any party (but the reduction is allowed to learn all the RO queries made by $\mathcal{A}$). The RPRed can only program the RO via an interface not fully controlled by itself, and is formalized via the notion of randomly programmable RO. Please refer to Section A for a formal definition of RPReds.

4.1 Definitions

In Definition 3, we define a splittable property for a BB-reduction in the ROM, which will be used to formalize the notion of committed-programming reduction in Definition 4. In both definitions, we will omit the superscript $\mathcal{O}(\cdot)$ and denote problem $\mathrm{P} = (\mathsf{IGen}, \mathsf{Orcl}, \mathsf{Vrfy})$ for simplicity of exposition.

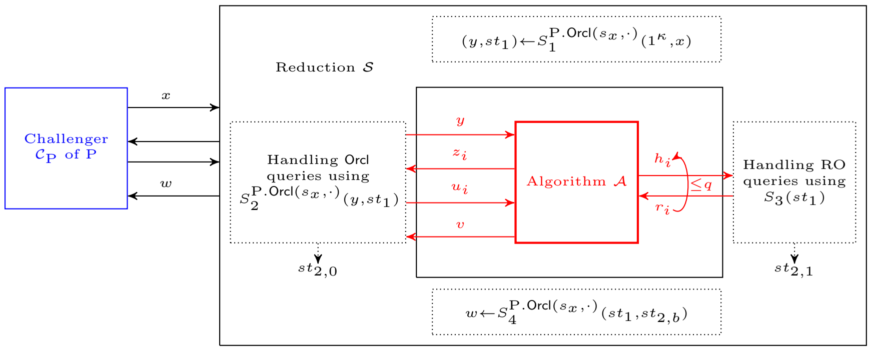
Basically, the splittable property says that the simulation of $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ and the RO $\mathcal{O}(\cdot)$ can be done by using two sub-algorithms that do not interact with each other. Note that for fixed algorithm $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ and query z, when and how $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ uses the RO $\mathcal{O}(\cdot)$ to compute a response to z is also fixed, one can count the RO queries and use an index function (taking z as input) to specify any subset of the RO queries required in this computation. Let q be a positive integer. We parameterize a splittable reduction with a triple (b, F, I), where $b \in \{0,1\}$ is a bit, $F(\cdot,\cdot)$ is a deterministic function (used to program the RO responses), and $I(\cdot)$ is an index function (which may be a dummy one, i.e., $I(\cdot) = \emptyset$).

Definition 3 (Splittability). *Using the notations above, a BB-reduction $\mathcal{S}$ from problem $\mathrm{P} = (\mathsf{IGen}, \mathsf{Orcl}, \mathsf{Vrfy})$ to $\mathrm{Q} = (\mathsf{IGen}^{\mathcal{O}(\cdot)}, \mathsf{Orcl}^{\mathcal{O}(\cdot)}, \mathsf{Vrfy}^{\mathcal{O}(\cdot)})$ in the ROM is (b, F, I)-splittable if for any $\mathcal{A}$ solving Q with at most q RO queries, $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$ can be split into four sub-algorithms (as depicted in Fig. 3):*

1. *Given a security parameter κ and an instance x of P as inputs, $\mathcal{S}$ first runs sub-algorithm $\mathcal{S}_1^{\mathrm{P}.\mathsf{Orcl}(s_x,\cdot)}(1^\kappa, x)$ to generate an instance y of Q and a state st_1, i.e., $(y, st_1) \leftarrow \mathcal{S}_1^{\mathrm{P}.\mathsf{Orcl}(s_x,\cdot)}(1^\kappa, x)$, which are then used as inputs to run sub-algorithms $\mathcal{S}_2^{\mathrm{P}.\mathsf{Orcl}(s_x,\cdot)}(y, st_1)$ and $\mathcal{S}_3(st_1)$ to interact with $\mathcal{A}$;*
2. *The sub-algorithm $\mathcal{S}_2^{\mathrm{P}.\mathsf{Orcl}(s_x,\cdot)}(y, st_1)$ takes an instance y of Q and a state st_1 as inputs, invokes a single instance of $\mathcal{A}$ with input y:*
   (a) Whenever receiving a $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ query z_i from $\mathcal{A}$, let $\{h_{i,j}\}_{j\in\{1,\dots,q_i\}}$ be all possible RO queries (indexed by the order of the query) required for computing a response u_i to z_i. Then, a set $\{r_j\}_{j\in\{1,\dots,q_i\}}$ is used as the RO responses to generate u_i (which may be $\perp$) such that $r_j = F(st_1, h_{i,j})$

for all $j \notin I(z_i)$, and r_j is randomly chosen with consistency (i.e., the same r_j is chosen for the same $h_{i,j}$) for all $j \in I(z_i) \subseteq \{1, \ldots, q_i\}$;[6]

(b) *Whenever $\mathcal{A}$ aborts or outputs a solution v of y at some time, $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}$ outputs a state $st_{2,0}$ (which may be an empty string ϵ) and aborts;*



$(y, st_1) \leftarrow S_1^{\mathsf{P.Orcl}(s_x,\cdot)}(1^\kappa, x)$
Reduction $\mathcal{S}$
x
Challenger $\mathcal{C}_{\mathrm{P}}$ of P
w
Handling Orcl queries using $S_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$
y
z_i
u_i
v
Algorithm $\mathcal{A}$
h_i
$\leq q$
r_i
Handling RO queries using $S_3(st_1)$
$st_{2,0}$
$st_{2,1}$
$w \leftarrow S_4^{\mathsf{P.Orcl}(s_x,\cdot)}(st_1, st_{2,b})$

Fig. 3. A (b, F, I)-splittable reduction $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$ from P to Q, where $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_4$ can have access to $\mathrm{P}.\mathsf{Orcl}(s_x, \cdot)$, but $\mathcal{S}_3$ does not access $\mathrm{P}.\mathsf{Orcl}(s_x, \cdot)$.

3. *The sub-algorithm $\mathcal{S}_3(st_1)$ takes a state st_1 as input, sets $k^* = q + 1$ and $st_{2,1} = \epsilon$ if $b = 0$, else chooses $k^* \xleftarrow{\$} \{1, \ldots, q\}$ if $b = 1$. Whenever receiving a RO query h from $\mathcal{A}$, the sub-algorithm $\mathcal{S}_3$ directly outputs $st_{2,1} = h$ and aborts if h is the k^*-th RO query, otherwise returns $r = F(st_1, h)$ to $\mathcal{A}$;*
4. *Finally, $\mathcal{S}$ returns $w \leftarrow \mathcal{S}_4^{\mathsf{P.Orcl}(s_x,\cdot)}(st_1, st_{2,b})$ as his own solution of x.*

Clearly, the parameter $b \in \{0, 1\}$ is used to distinguish the two typical usages of the RO in existing reductions: 1) for $b = 0$, $\mathcal{S}$ will directly use the output of $\mathcal{A}$ to solve its own problem; while 2) for $b = 1$, $\mathcal{S}$ relies on a critical RO query from $\mathcal{A}$ to solve its own problem (and the output of $\mathcal{A}$ is ignored). Note that the sub-algorithms $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$ and $\mathcal{S}_3(st_1)$ are the exact parts of $\mathcal{S}$ directly interacting with the algorithm $\mathcal{A}$ (as depicted in Fig. 3), and solely try to answer the queries from (a single instance of) $\mathcal{A}$, a splittable reduction $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$ will not rewind $\mathcal{A}$. Moreover, $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$ and $\mathcal{S}_3(st_1)$ only share the common input st_1, and will not interact with each other after being invoked. We finally note that $\mathcal{S}_3$ does not have access to $\mathrm{P}.\mathsf{Orcl}(s_x, \cdot)$, which is necessary for isolating the behaviors of $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}$ and $\mathcal{S}_3$.

We now consider two events that may help the adversary distinguish the game simulated by $\mathcal{S}$ from the real one. Let E_0 be the event that $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$ returns a failure symbol $u_i = \perp$ to some query z_i from $\mathcal{A}$. Let E_1 be the event that $\mathcal{A}$ makes a "bad" RO query $h_{i,j}$ to $\mathcal{S}_3(st_1)$ such that $h_{i,j}$ is the j-th RO query required for computing a response to some $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ query z_i and $j \in I(z_i) \neq \emptyset$.

[6] In the case that $j \in I(z_i)$, the algorithm $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}$ may not know the values $(h_{i,j}, r_j)$ since both might come from it oracle $\mathrm{P}.\mathsf{Orcl}(s_x, \cdot)$

Definition 4 (Committed-Programming Reduction). *Using the notations of Definition 3 and above, a BB-reduction $\mathcal{S}$ from problem* $\mathrm{P} = (\mathsf{IGen}, \mathsf{Orcl}, \mathsf{Vrfy})$ *to problem* $\mathrm{Q} = (\mathsf{IGen}^{\mathcal{O}(\cdot)}, \mathsf{Orcl}^{\mathcal{O}(\cdot)}, \mathsf{Vrfy}^{\mathcal{O}(\cdot)})$ *in the ROM is a (b, F, I)-committed-programming reduction (CPRed) if for any (even unbounded) algorithm $\mathcal{A}$ solving* Q *with non-negligible advantage ϑ_A by making* $q_1 = \mathrm{poly}(\kappa)$ $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ *queries and* $q_2 = \mathrm{poly}(\kappa)$ *RO queries, $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$ is (b, F, I)-splittable, and*

- *For $b = 0$, we have that $I(\cdot) = \emptyset$ (which implies that* $\Pr[E_1] = 0$*), and that the distribution of $\{u_i\}$ produced by $\mathcal{S}_2^{\mathrm{P}.\mathsf{Orcl}(s_x,\cdot)}(y, st_1)$, conditioned on E_0 not happening, is statistically close to the output distribution of the real oracle* $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$. *Moreover,*
  - *the probability that E_0 does not happen is noticeable,*
  - *the advantage that $\mathcal{A}$ outputs a solution v of y to $\mathcal{S}_2^{\mathrm{P}\mathsf{Orcl}(s_x,\cdot)}(y, st_1)$, conditioned on E_0 not happening, is negligibly close to ϑ_A, and*
  - *any solution v of y can be used by $\mathcal{S}_4^{\mathrm{P}.\mathsf{Orcl}(s_x,\cdot)}$ to find a solution w of x with non-negligible advantage.*
- *For $b = 1$, we have that* $\Pr[E_0] = 0$*, and that the distribution of $\{u_i\}$ produced by $\mathcal{S}_2^{\mathrm{P}.\mathsf{Orcl}(s_x,\cdot)}(y, st_1)$, conditioned on E_1 not happening, is identical to the output distribution of the real oracle* $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$. *Moreover,*
  - *the advantage that $\mathcal{A}$ outputs a solution v of y to $\mathcal{S}_2^{\mathrm{P}\mathsf{Orcl}(s_x,\cdot)}(y, st_1)$ is negligible whether the event E_1 happens or not (which implies that* $\Pr[E_1]$ *is negligibly close to ϑ_A), and*
  - *any "bad" RO query $h_{i,j}$ for some $j \in I(z_i)$ to $\mathcal{S}_3$ can be used by $\mathcal{S}_4^{\mathrm{P}.\mathsf{Orcl}(s_x,\cdot)}$ to find a solution w of x with non-negligible advantage.*

Many well-known schemes are probably secure under CPReds. We now take the full-domain hash (FDH) signature [4] as an example to explain how a CPRed works. Let $\mathsf{Perm}(\cdot,\cdot) : \{0,1\}^\ell \to \{0,1\}^\ell$ be some family of trapdoor permutations (TDP). The goal of a reduction for the unforgeability of the FDH signature is, given an function index s^* and a string $y^* \in \{0,1\}^\ell$, to output a preimage $x^* \in \{0,1\}^\ell$ such that $y^* = \mathsf{Perm}(s^*, x^*)$. Then, for any adversary $\mathcal{A}$ making q_r RO queries and q_s signing queries, there is a $(0, F, \emptyset)$-CPRed $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$:

- $(s^*, st_1 = (f_1, f_2, s^*, y^*)) \leftarrow \mathcal{S}_1(1^\kappa, (s^*, y^*))$, where $f_1 : \{0,1\}^* \to \{1, \ldots, p\}$ and $f_2 : \{0,1\}^* \to \{0,1\}^\ell$ are two random functions (which can be replaced with efficiently computable k-wise independent functions that are equivalent to random functions up to k queries [22]) and p is some positive integer;
- $\mathcal{S}_2(s^*, st_1)$ uses s^* as the challenge public key to invoke $\mathcal{A}$. For a signing query $z_i \in \{0,1\}^\ell$ from $\mathcal{A}$, $\mathcal{S}_2$ returns a failure symbol $u_i = \perp$ if $f_1(z_i) = 1$, and $u_i = f_2(z_i) \in \{0,1\}^\ell$ otherwise. Whenever $\mathcal{A}$ outputs a forgery $v = (z^*, u^*)$ and abort, $\mathcal{S}_2$ outputs $st_{2,0} = (z^*, u^*)$ if $f_1(z^*) = 1 \wedge \mathsf{Perm}(s^*, u^*) = y^*$, and $st_{2,0} = \epsilon$ otherwise;
- $\mathcal{S}_3(st_1)$ sets $st_{2,1} = \epsilon$ and uses $F(st_1, \cdot)$ to answer a RO query h from $\mathcal{A}$:

$$F(st_1, h) = \begin{cases} y^*, & \text{if } f_1(h) = 1; \\ \mathsf{Perm}(s^*, f_2(h)), & \text{otherwise.} \end{cases}$$

– $\mathcal{S}_4(st_1, st_{2,0})$ returns $w^* = u^*$ if $st_{2,0} = (z^*, u^*)$, and $\perp$ otherwise.

We will further investigate the FDH signature and its associated CPRed in Section 5.2 and **??**. Besides, the variant PSF-FDH where the TDP is replaced with preimage sampleable trapdoor functions (PSF), e.g., the one given by Gentry et al. [17], is also provably EUF-CMA secure under a $(0, F, \emptyset)$-CPRed, which can be confirmed by inspection of the reductions given in [17] and [7].

We also note that the Boneh-Franklin identity-based encryption (IBE) [8], the "implicit rejection" variant of the KEM from TDPs due to Shoup [29], and the "implicit rejection" [19] variants of the KEM from the Fujisaki-Okamoto (FO) transform [15] are provably secure under $(1, F, I)$-CRPeds for some deterministic function F (which can be implemented using k-wise independent functions) and index function I. We take the IND-CPA security of the Boneh-Franklin IBE [8] as an example to elaborate a bit more on the role of I. Informally, apart from the RO queries, the adversary can make two types of other queries: the "user key extraction" query id whose response requires one RO query (for hashing the user identity id), and the "challenge" query (id^*, μ_0^*, μ_1^*) whose response requires two RO queries (one for hashing the challenge user identity id^*, and the other for masking the challenge plaintext μ_δ^* for some $\delta \in \{0, 1\}$). The security proof in [8] implies a $(1, F, I)$-CRPed $\mathcal{S}$ with the following index function I:

$$I(query\ type, query\ content) = \begin{cases} \emptyset, & \text{if } \textit{query type}=\text{"user key extraction"}; \\ 2, & \text{if } \textit{query type}=\text{"challenge"}, \end{cases}$$

where each input of I is split into two parts to specify the query type and the query content for convenience. In this case, the sub-algorithm $\mathcal{S}_2$ of $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$ will only replace the response to the second RO query, i.e., the one for masking μ_δ^*, in responding the "challenge" query with a random one.

4.2 Relation to Reductions in the QROM

In this section, we show that any problem having a CPRed associated with an instance-extraction algorithm implies a reduction in the QROM. Informally, the instance-extraction algorithm can extract an instance of problem P (and produce an auxiliary string for the RO simulation) from an instance of Q, which is needed to ensure that the sub-algorithm (of the CPRed) for simulating the RO can be quantized to simulate a QRO, and that one can smoothly replace a real QRO with a simulated one in the security proof. We note that the history reduction given in [7] also requires similar (and possibly stronger) algorithm.

Theorem 4. *Using the notations of Definition 4, if* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$ *is a* (b, F, I)*-CPRed from problem* P *to* $\mathrm{Q} = (\mathsf{IGen}^{\mathcal{O}(\cdot)}, \mathsf{Orcl}^{\mathcal{O}(\cdot)}, \mathsf{Vrfy}^{\mathcal{O}(\cdot)})$*, and there is an extra instance-extraction algorithm* $\mathrm{Q}.\mathsf{Extract}(\cdot,\cdot)$ *for problem* Q *such that for any QRO* $\mathcal{O}^q = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot)) \xleftarrow{\$} \mathcal{F}_{n,m}$, $(y, s_y) \leftarrow \mathrm{Q}.\mathsf{IGen}^{\mathcal{O}^q_{priv}(\cdot)}(1^\kappa)$ *and* $(x, s_x, st_1) \leftarrow \mathrm{Q}.\mathsf{Extract}(y, s_y)$*, we have that:*

– *Given y, no algorithm making q_3 quantum queries can distinguish the oracle $\mathcal{O}_{quant}(\cdot) : |h_1, h_2\rangle \to |h_1, F(st_1, h_1) \oplus h_2\rangle$ from $\mathcal{O}^q_{pub}(\cdot)$ with non-negligible advantage over the randomness used by $\mathsf{Q.IGen}^{\mathcal{O}^q_{priv}(\cdot)}$ and $\mathsf{Q.Extract}$;*
– *The distribution of (x, s_x, y, st_1) is statistically close to that of (x', s'_x, y', st'_1) generated by running $(x', s'_x) \leftarrow \mathsf{P.IGen}(1^\kappa)$ and $(y', st'_1) \leftarrow \mathcal{S}_1^{\mathsf{P.Orcl}(s'_x,\cdot)}(1^\kappa, x)$, over the randomness used by $\mathsf{Q.IGen}^{\mathcal{O}^q_{priv}(\cdot)}$, $\mathsf{Q.Extract}$, $\mathsf{P.IGen}$ and $\mathcal{S}_1^{\mathsf{P.Orcl}(s'_x,\cdot)}$,*

then for any quantum polynomial time algorithm $\mathcal{A}$ attacking Q using at most q_1 queries to $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ and q_2 QRO queries such that $dq_1 + q_2 \leq q_3$, there is a reduction $\mathcal{S}'$ from P to Q in the QROM, where d be the maximum number of RO queries that is required for computing a response to any $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query.

One can check that the schemes mentioned after Definition 4 have CPReds associated with natural instance-extraction algorithms (note that Zhandry [34] showed that k-wise independent functions can be used to simulate a (Q)RO, and that a (Q)RO can be used as a random function even in the quantum setting [27]). Thus, we have that there are reductions from the security of the FDH [4], the PSF-FDH [17], and the "implicit-rejection" KEM variants from TDPs [29] and the FO transform [15,19] to their underlying problems in the QROM. We clarify that Theorem 4 provides guarantee on the security of a scheme only when its underlying problems are hard (against quantum polynomial time adversaries). In particular, Theorem 4 implies that there is a reduction from the IND-ID-CPA security of the BF-IBE [8] to the hardness of the BDH problem in the QROM, but it provides no guarantee on the security of BF-IBE against quantum adversaries since the BDH problem can be efficiently solved by using the Shor algorithm [28]. We finally note that Theorem 4 not only subsumes the security reductions for the FDH [4] given in [34], for the PSF-FDH [17] given in [7], and for the "implicit rejections" KEMs from the FO transform [15,19] given in [21], but also gives new security reductions for other schemes such as the "implicit-rejection" KEM variants from TDPs [29] in the QROM.

We now give a formal proof of Theorem 4, which are mainly based on the following two facts: 1) $\mathcal{S}$ is a (b, F, I)-CPRed, the simulation of the RO and that of the $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ are relatively independent; and 2) the simulation of the RO can be safely quantized by the assumption in Theorem 4.

Proof. Let $\mathcal{A}$ be any quantum algorithm that solves problem Q with non-negligible advantage ϑ_A, by making at most $q_1 = \text{poly}(\kappa)$ queries to $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ and $q_2 = \text{poly}(\kappa)$ QRO queries. We first give the description of the reduction $\mathcal{S}'$. Formally, given the security parameter κ and an instance x of P as inputs, $\mathcal{S}'$ first computes $(y, st_1) \leftarrow \mathcal{S}_1^{\mathsf{P.Orcl}(s_x,\cdot)}(1^\kappa, x)$ and sets $st_{2,0} = st_{2,1} = \epsilon$. If $b = 1$, it uniformly chooses an integer $k^* \xleftarrow{\$} \{1, \ldots, q_2\}$, else sets $k^* = q_2 + 1$. Define the quantum oracle $\mathcal{O}_{quant}(\cdot) : |h_1, h_2\rangle \to |h_1, F(st_1, h_1) \oplus h_2\rangle$. Then, $\mathcal{S}'$ invokes $\mathcal{A}$ with input y, answers the $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ queries from $\mathcal{A}$ by running $st_{2,0} \leftarrow \mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$ and handles a QRO query $|\phi\rangle = |h_1, h_2\rangle$ from $\mathcal{A}$ by using $\mathcal{O}_{quant}(\cdot)$ as follows: if $|\phi\rangle$ is the k^*-th QRO query, then measure the argument of the query and set $st_{2,1}$ to be the measurement outcome, else return $|\psi\rangle = \mathcal{O}_{quant}(|\phi\rangle)$ to $\mathcal{A}$. After obtaining $st_{2,b} \neq \epsilon$, $\mathcal{S}'$ computes and outputs $w \leftarrow \mathcal{S}_4^{\mathsf{P.Orcl}(s_x,\cdot)}(st_1, st_{2,b})$ as the solution to its own challenge x of problem P.

It suffices to show that $\mathcal{S}'$ is a valid reduction in the QROM. We will finish the proof by using a sequence of five main games G_i for $i \in \{0, \ldots, 4\}$, where game G_0 is the real game and game G_4 is essentially the game simulated by $\mathcal{S}'$. For technical reason, we also define a sequence of sub-games $G_{1,j}$ between the two main games G_1 and G_2 for $j = \{0, \ldots, dq_1\}$ such that $G_{1,0} = G_1$, $G_{1,dq_1} = G_2$, and $G_{1,j}$ is modified from $G_{1,j-1}$ by replacing at most one real RO response with a uniformly random one (to apply Lemma 1). We outline the changes between two consecutive games in Table 1. The game sequences is formally given below.

Table 1. Outline of the changes between two consecutive games

Games	Changes w.r.t. Previous Game	Note
G_0	**Instance:** pick a QRO $\mathcal{O}^q(\cdot) = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot))$ compute $(y, s_y) \leftarrow \mathsf{Q.IGen}^{\mathcal{O}^q_{priv}(\cdot)}(1^\kappa)$ Q.Orcl **query** z: compute $u = \mathsf{Q.Orcl}^{\mathcal{O}^q_{priv}(\cdot)}(s_y, z)$ **QRO. query** $\|\phi\rangle$: compute $\mathcal{O}^q_{pub}(\|\phi\rangle)$	real game
G_1	For each $\mathsf{Q.Orcl}^{\mathcal{O}^q_{priv}(\cdot)}$ query z, computes $I(z) \subseteq \{1, 2, \ldots, d\}$	a conceptual change of G_0
$G_{1,j}$	For $1 \leq j \leq dq_1$, define $k_j = \lfloor \frac{j-1}{d} \rfloor + 1$ and $\ell_j = j \mod d$, and for the k_j-th $\mathsf{Q.Orcl}^{\mathcal{O}^q_{priv}(\cdot)}$ query z_{k_j}, replace $\mathcal{O}^q_{priv}(h_{k_j,\ell_j})$ with a random and consistent r_{k_j,ℓ_j} if h_{k_j,ℓ_j} is ℓ_j-th RO query needed in computing the response to z_{k_j} and $\ell_j \in I(z_{k_j})$	Let $G_{1,0} = G_1$, then games $G_{1,j-1}$ and $G_{1,j}$ are connected by the One-way to Hiding Lemma 1
G_2	For each $\mathsf{Q.Orcl}^{\mathcal{O}^q_{priv}(\cdot)}$ query z, replace $\mathcal{O}^q_{priv}(h_\ell)$ with a random and consistent r_ℓ if h_ℓ is ℓ-th RO query needed in computing the response to z and $\ell \in I(z)$	a conceptual change of G_{1,dq_1}
G_3	Compute $(x, s_x, st_1) \leftarrow \mathsf{Q.Extract}(y, s_y)$, and handle the queries to $\mathcal{O}^q_{priv}(\cdot)$ and $\mathcal{O}^q_{pub}(\cdot)$ by using $F(st_1, \cdot)$ and $\mathcal{O}_{quant}(\cdot)$: $\|h_1, h_2\rangle \rightarrow \|h_1, F(st_1, h_1) \oplus h_2\rangle$	$\mathcal{O}' = (F(st_1, \cdot), \mathcal{O}_{quant}(\cdot))$ is indistinguishable from $\mathcal{O}^q(\cdot)$ in game G_2 for any $q_3 \geq dq_1 + q_2$ queries
G_4	Compute $(x, s_x) \leftarrow \mathsf{P.IGen}(1^\kappa)$ and $(y, st_1) \leftarrow \mathcal{S}_1^{\mathsf{P.Orcl}(s_x,\cdot)}(x)$, and handle the Q.Orcl query by using $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$	(x, s_x, y, st_1) is statistically close to that in G_3

Game G_0: This game is the real security game of problem Q in the QROM. Concretely, the challenger $\mathcal{C}_{\mathsf{Q}}$ picks a QRO $\mathcal{O}^q(\cdot) = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot)) \xleftarrow{\$} \mathcal{F}_{n,m}$, and computes $(y, s_y) \leftarrow \mathsf{Q.IGen}^{\mathcal{O}^q_{priv}(\cdot)}(1^\kappa)$. Then, it invokes $\mathcal{A}$ with input y:

– When receiving a QRO query $|\phi\rangle = |h_1, h_2\rangle$ from $\mathcal{A}$, send $|\phi\rangle$ to $\mathcal{O}^q(\cdot)$ via the public interface $\mathcal{O}^q_{pub}(\cdot)$, and return whatever received to $\mathcal{A}$;
– When receiving a $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query z from $\mathcal{A}$, compute $u = \mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}(s_y, z)$ by using s_y. Whenever the computation stops to make a classical RO query h, send h to $\mathcal{O}^q(\cdot)$ via the private interface $\mathcal{O}^q_{priv}(\cdot)$, and use the response $\mathcal{O}^q(h)$ to continue the computation. Finally, return u to $\mathcal{A}$.
– When $\mathcal{A}$ outputs a solution v of y, compute and output $\mathsf{Q.Vrfy}^{\mathcal{O}^q_{priv}(\cdot)}(y, s_y, v)$.

Let Succ_i be the event that the algorithm $\mathcal{A}(y)$ succeeds to output a solution v of y (i.e., $\mathrm{Q}.\mathsf{Vrfy}(y, s_y, v) = 1$), where $i \in \{0, \ldots, 4\}$. By definition, the advantage of $\mathcal{A}$ in game G_0 is equal to $\Pr[\mathsf{Succ}_0] - t(\kappa)$, i.e., $\Pr[\mathsf{Succ}_0] = \vartheta_A + t(\kappa)$, where $t(\cdot)$ is the threshold function for the hardness of problem Q in Definition 2.

Game G_1: This game is almost identical to game G_0 except that $\mathcal{C}_\mathrm{Q}$ always internally computes the index function $I(z)$ for any $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ query z from $\mathcal{A}$.

Lemma 4. $\Pr[\mathsf{Succ}_0] = \Pr[\mathsf{Succ}_1]$.

Proof. This lemma follows because game G_1 is a conceptual change of G_0. $\square$

Let $G_{1,0} = G_1$ and $\mathsf{Succ}_{1,0} = \mathsf{Succ}_1$. For $1 \leq j \leq dq_1$, define $k_j = \lfloor \frac{j-1}{d} \rfloor + 1$ and $\ell_j = j \mod d$. Now, we define a sequence of sub-games $G_{1,j}$ for $1 \leq j \leq dq_1$. Informally, game $G_{1,j}$ is identical to game $G_{1,j-1}$ except that when computing the response to the k_j-th $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ query z_{k_j} from $\mathcal{A}$, the challenger $\mathcal{C}_\mathrm{Q}$ replaces $\mathcal{O}^q_{priv}(h_{k_j,\ell_j})$ in game $G_{1,j-1}$ with a random and consistent r_{k_j,ℓ_j} (chosen from an appropriate domain) in game $G_{1,j}$ if h_{k_j,ℓ_j} is the ℓ_j-th RO query required by $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ in computing the response to z_{k_j} and $\ell_j \in I(z_{k_j}) \subseteq \{1, \ldots, d\}$. Clearly, $\mathcal{A}$ can only distinguish game $G_{1,j}$ from $G_{1,j-1}$ by making a QRO query using h_{k_j,ℓ_j}. Let $\mathsf{Succ}_{1,j}$ be the event that $\mathcal{A}(y)$ outputs a solution v of y in game $G_{1,j}$, where $j \in \{1, \ldots, dq_1\}$. Let L be an initially empty list in the following games, which is used to keep consistency of the RO responses.

Game $G_{1,j}$: This game is almost identical to game $G_{1,j-1}$ except that the challenger $\mathcal{C}_\mathrm{Q}$ handles the $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ queries from $\mathcal{A}$ as follows:

– When receiving a $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ query z from $\mathcal{A}$, use s_y to compute the response $u = \mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}(s_y, z)$ and count the RO queries needed by the computation. Whenever the computation stops to make a RO query h, if there is a pair $(h, r) \in L$ (i.e., the value $\mathcal{O}^q_{priv}(h)$ has been previously replaced with a uniformly random r), then use the same r to continue the computation for consistency. Otherwise, if z is the k_j-th $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ query from $\mathcal{A}$, h is the ℓ_j-th RO query required by this computation and $\ell_j \in I(z)$, then choose a uniformly random r to continue the computation and append (h, r) to the list L, else continue the computation as in game $G_{1,j-1}$, where $k_j = \lfloor \frac{j-1}{d} \rfloor + 1$ and $\ell_j = j \mod d$. Finally, return u to $\mathcal{A}$.

Let $\mathcal{B}_{1,j}$ be an algorithm which interacts with $\mathcal{A}$ as the challenger in game $G_{1,j}$ except that it chooses an integer $k^* \xleftarrow{\$} \{1, \ldots, q_2\}$, runs $\mathcal{A}$ until just after receiving the k^*-th QRO query $|\phi\rangle = |h_1, h_2\rangle$ from $\mathcal{A}$, measures the argument of the query in the computational basis, and outputs the measurement result $\hat{h}$ ($\mathcal{B}_{1,j}$ outputs $\perp$ if $\mathcal{A}$ makes less than k^* RO queries). Let $\delta_{1,j}$ be the probability that $\hat{h} = h_{k_j,\ell_j} \wedge \ell_j \in I(z_{k_j})$, where z_{k_j} is the k_j-th $\mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}$ query from $\mathcal{A}$, and h_{k_j,ℓ_j} is the ℓ_j-th RO query required for computing $u = \mathrm{Q}.\mathsf{Orcl}^{\mathcal{O}(\cdot)}(s_y, z_{k_j})$. Clearly, $\delta_{1,j}$ is only useful if $\mathcal{S}$ is a $(1, F, I)$-CPRed, since $I(\cdot) = \emptyset$ otherwise.

Lemma 5. *If* $\mathcal{S}$ *is a* $(0, F, \emptyset)$*-CPRed, then* $\Pr[\mathsf{Succ}_{1,j-1}] = \Pr[\mathsf{Succ}_{1,j}]$*, else* $|\Pr[\mathsf{Succ}_{1,j-1}] - \Pr[\mathsf{Succ}_{1,j}]| \leq 2q_2\sqrt{\delta_{1,j}}$.

Proof. Note that if $\mathcal{S}$ is a $(0, F, \emptyset)$-CPRed, the change from game $G_{1,j-1}$ to game $G_{1,j}$ is conceptual, i.e., we always have that $\Pr[\mathsf{Succ}_{1,j-1}] = \Pr[\mathsf{Succ}_{1,j}]$. Otherwise, if $\mathcal{S}$ is a $(1, F, I)$-CPRed, the only difference between $G_{1,j-1}$ and $G_{1,j}$ is that the challenger $\mathcal{C}_{\mathrm{Q}}$ might replace the value $\mathcal{O}^q_{priv}(h_{k_j,\ell_j})$ in game $G_{1,j-1}$ with a uniformly random r_{k_j,ℓ_j} in game $G_{1,j}$ if z_{k_j} is the k_j-th $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query from $\mathcal{A}$, h_{k_j,ℓ_j} is the ℓ_j-th RO query required for computing $u_{k_j} = \mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}(s_y, z_{k_j})$ and $\ell_j \in I(z_{k_j})$. By Lemma 1, we have that $|\Pr[\mathsf{Succ}_{1,j-1}] - \Pr[\mathsf{Succ}_{1,j}]| \leq 2q_2\sqrt{\delta_{1,j}}$ holds. This completes the proof of Lemma 5. $\square$

Game G_2: Let $\mathcal{O}^q(\cdot) = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot))$ be a real QRO, the challenger $\mathcal{C}_{\mathrm{Q}}$ in this game computes $(y, s_y) \leftarrow \mathsf{Q.IGen}^{\mathcal{O}^q_{priv}(\cdot)}(1^\kappa)$, and invokes $\mathcal{A}$ with y:

– When receiving a quantum RO query $|\phi\rangle = |h_1, h_2\rangle$ from $\mathcal{A}$, send $|\phi\rangle$ to $\mathcal{O}^q(\cdot)$ via the public interface $\mathcal{O}^q_{pub}(\cdot)$, and return whatever obtained to $\mathcal{A}$;
– When receiving a $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query z from $\mathcal{A}$, compute $u = \mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}(s_y, z)$ by using s_y and count the RO queries required by the computation. Whenever the computation stops to make a RO query h, if there is a pair $(h, r) \in L$, then use the same r to continue the computation. Otherwise, if h is the ℓ-th RO query required by this computation and $\ell \in I(z)$, then choose a uniformly random r to continue the computation and append (h, r) to the list L, else use $\mathcal{O}^q_{priv}(h)$ to continue the computation. Finally, return u to $\mathcal{A}$;
– When $\mathcal{A}$ outputs a solution v of y, compute and output $\mathsf{Q.Vrfy}^{\mathcal{O}^q_{priv}(\cdot)}(y, s_y, v)$.

Let $\mathcal{B}_i$ be an algorithm which interacts with $\mathcal{A}$ as the challenger in game G_i for $i \in \{2, 3, 4\}$ except that it chooses an integer $k^* \xleftarrow{\$} \{1, \ldots, q_2\}$, runs $\mathcal{A}$ until just after receiving the k^*-th QRO query $|\phi\rangle = |h_1, h_2\rangle$ from $\mathcal{A}$, measures the argument of the query and outputs the measurement outcome $\hat{h}$ ($\mathcal{B}_i$ outputs $\perp$ if $\mathcal{A}$ makes less than k^* RO queries). Let δ_i be the probability that $\hat{h} = h_{k_j,\ell_j} \wedge \ell_j \in I(z_{k_j})$ for any $1 \leq j \leq dq_1$, where z_{k_j} is the k_j-th $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query from $\mathcal{A}$, and h_{k_j,ℓ_j} is the ℓ_j-th RO query required for computing $u_{k_j} = \mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}(s_y, z_{k_j})$.

Lemma 6. $\Pr[\mathsf{Succ}_2] = \Pr[\mathsf{Succ}_{1,dq_1}]$. *Moreover, if $\mathcal{S}$ is a $(1, F, I)$-CPRed, then $\delta_2 \geq \delta_{1,j}$ holds for any $1 \leq j \leq dq_1$.*

Proof. This first claim follows from the fact that games G_2 and G_{1,dq_1} are identical. Note that if $\mathcal{A}$ does not make a QRO query using any element in $\{\{h_{k_j,\ell_j}\}_{\ell_j \in I(z_{k_j})}\}_{1 \leq j \leq dq_1}$, the view of $\mathcal{A}$ in games $G_{1,j}$ for $0 \leq j \leq dq_1$ is identically distributed. Moreover, before $\mathcal{A}$ making a QRO query using one of the elements in $\{\{h_{k_j,\ell_j}\}_{\ell_j \in I(z_{k_j})}\}_{1 \leq j \leq dq_1}$, we always have that the view of $\mathcal{A}$ in games $G_{1,j}$ for $0 \leq j \leq dq_1$ is identically distributed, which implies that $\delta_2 \geq \delta_{1,j}$ for any $1 \leq j \leq dq_1$. This completes the proof of Lemma 6. $\square$

Game G_3: This game is almost identical to game G_2 except that the challenger $\mathcal{C}_{\mathrm{Q}}$ replaces the real QRO $\mathcal{O}^q(\cdot)$ with a simulated one using $\mathsf{F}(\cdot, \cdot)$. Specifically, $\mathcal{C}_{\mathrm{Q}}$ first computes $(x, s_x, st_1) \leftarrow \mathsf{Q.Extract}(y, s_y)$ and defines the quantum oracle $\mathcal{O}_{quant}(\cdot) : |h_1, h_2\rangle \rightarrow |h_1, F(st_1, h_1) \oplus h_2\rangle$. Then, it invokes $\mathcal{A}$ with y:

– When receiving a quantum RO query $|\phi\rangle = |h_1, h_2\rangle$ from $\mathcal{A}$, compute and return $|\psi\rangle = \mathcal{O}_{quant}(|\phi\rangle) = |h_1, F(st_1, h_1) \oplus h_2\rangle$ to $\mathcal{A}$;
– When receiving a $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query z from $\mathcal{A}$, answer this query z the same as in game G_2 except that the value $\mathcal{O}^q_{priv}(h)$ which is needed for computing the response in game G_2 is replaced with $F(st_1, h)$ in game G_3;
– When $\mathcal{A}$ outputs a solution v of y, compute and output $\mathsf{Q.Vrfy}^{F(st_1,\cdot)}(y, s_y, v)$.

Lemma 7. $|\Pr[\mathsf{Succ}_3] - \Pr[\mathsf{Succ}_2]| \leq \text{negl}(\kappa)$. *Moreover, if $\mathcal{S}$ is a $(1, F, I)$-CPRed, then* $|\delta_3 - \delta_2| \leq \text{negl}(\kappa)$.

Proof. The only difference between games G_2 and G_3 is that the QRO $\mathcal{O}^q = (\mathcal{O}^q_{priv}(\cdot), \mathcal{O}^q_{pub}(\cdot))$ in game G_2 is replaced with $(F(st_1, \cdot), \mathcal{O}_{quant}(\cdot))$ in game G_3. Note that the challenger $\mathcal{C}_{\mathrm{Q}}$ will make at most dq_1 queries to $F(st_1, \cdot)$, and the adversary will make at most q_2 queries to $\mathcal{O}_{quant}(\cdot)$. Since for any $(y, s_y) \leftarrow \mathsf{Q.IGen}^{\mathcal{O}(\cdot)}(1^\kappa)$ and $(x, s_x, st_1) \leftarrow \mathsf{Q.Extract}(y, s_y)$, no algorithm making $q_3 \geq dq_1 + q_2$ QRO queries can distinguish $\mathcal{O}_{quant}(\cdot) : |h_1, h_2\rangle \rightarrow |h_1, F(st_1, h_1) \oplus h_2\rangle$ from $\mathcal{O}^q_{pub}(\cdot)$ by our assumption in Theorem 4, the view of $\mathcal{A}$ in games G_3 and G_2 are indistinguishable. This completes the proof of Lemma 7. □

Game G_4: This game is almost identical to game G_3 except that $\mathcal{C}_{\mathrm{Q}}$ generates (x, s_x, y, st_1) by running $(x, s_x) \leftarrow \mathsf{P.IGen}(1^\kappa)$ and $(y, st_1) \leftarrow \mathcal{S}_1^{\mathsf{P.Orcl}(s_x,\cdot)}(x)$, and handles the $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query by using $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$.

Let E_0 be the event that $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$ returns a failure symbol $\perp$ to some $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query in game G_4. We have the following lemma.

Lemma 8. *If $\mathcal{S}$ is a $(0, F, \emptyset)$-CPRed, then* $\Pr[\mathsf{Succ}_4|\neg E_0] \geq \Pr[\mathsf{Succ}_3] - \text{negl}(\kappa)$, *else* $|\Pr[\mathsf{Succ}_4] - \Pr[\mathsf{Succ}_3]| \leq \text{negl}(\kappa)$ *and* $|\delta_4 - \delta_3| \leq \text{negl}(\kappa)$.

Proof. There are two differences between games G_4 and G_3: 1) the way of generating (x, s_x, y, st_1), and 2) the way of handling the $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ queries. Specifically, the tuple (x, s_x, y, st_1) in game G_3 is generated by using $(y, s_y) \leftarrow \mathsf{Q.IGen}^{\mathcal{O}^q_{priv}(\cdot)}(1^\kappa)$ and $(x, s_x, st_1) \leftarrow \mathsf{Q.Extract}(y, s_y)$, while that in game G_4 is generated by using $(x, s_x) \leftarrow \mathsf{P.IGen}(1^\kappa)$ and $(y, st_1) \leftarrow \mathcal{S}_1^{\mathsf{P.Orcl}(s_x,\cdot)}(1^\kappa, x)$. Moreover, each $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ query in game G_3 is handled by using the real oracle $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}(s_y, \cdot)$, while that in game G_4 is handled by using $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$. By the assumption on $\mathsf{Q.Extract}$ in Theorem 4, the distribution of (x, s_x, y, st_1) are statistically indistinguishable in both games.

Now, we consider the case that $\mathcal{S}$ is a $(0, F, \emptyset)$-CPRed. By the requirement on $\mathcal{S}_2$ in Definition 4, the distribution of the $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ responses generated by $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$, conditioned on E_0 not happening, in game G_4 is statistically close to the output distribution of the real oracle $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$, which is then indistinguishable from that generated in game G_3. In other words, if E_0 does not happen, the view of $\mathcal{A}$ in both games G_4 and G_3 is indistinguishable, which implies that $\Pr[\mathsf{Succ}_4|\neg E_0] \geq \Pr[\mathsf{Succ}_3] - \text{negl}(\kappa)$.

As for the case that $\mathcal{S}$ is a $(1, F, I)$-CPRed, the $\mathsf{Q.Orcl}^{\mathcal{O}(\cdot)}$ responses generated in game G_3 are essentially identical to that generated by $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$ by the requirement on $\mathcal{S}_2$ in Definition 3. Thus, games G_4 and G_3 are indistinguishable in the view of $\mathcal{A}$. In other words, we have $|\Pr[\mathsf{Succ}_4] - \Pr[\mathsf{Succ}_3]| \leq \mathrm{negl}(\kappa)$ and $|\delta_4 - \delta_3| \leq \mathrm{negl}(\kappa)$. This completes the proof of Lemma 8. □

It is easy to check that the view of $\mathcal{A}$ in game G_4 is identical to the one simulated by $\mathcal{S}'$. Thus, it suffices to analyze the behavior of $\mathcal{A}$ in game G_4. First, if $\mathcal{S}$ is a $(0, F, \emptyset)$-CPRed, by the law of total probability we have that $\Pr[\mathsf{Succ}_4] = \Pr[\mathsf{Succ}_4|E_0] \cdot \Pr[E_0] + \Pr[\mathsf{Succ}_4|\neg E_0] \cdot \Pr[\neg E_0]$. Since the behavior of $\mathcal{A}$ is unpredictable after E_0 happens, we now use a simplifying assumption that conditioned on E_0 happening, the advantage that $\mathcal{A}$ will output a solution v of y to $\mathcal{S}_2^{\mathsf{P.Orcl}(s_x,\cdot)}(y, st_1)$ is negligible, namely, $\Pr[\mathsf{Succ}_4|E_0] = t(\kappa) + \mathrm{negl}(\kappa)$. Note that one can always modify $\mathcal{A}$ to satisfy this simplifying assumption by running a trivial solving algorithm for problem Q after receiving a failure symbol $\perp$ from $\mathcal{S}_2^{\mathsf{POrcl}(s_x,\cdot)}(y, st_1)$. Thus, the probability that $\mathcal{A}$ can output a solution v of y to $\mathcal{S}_2^{\mathsf{POrcl}(s_x,\cdot)}(y, st_1)$ is

$$\Pr[\mathsf{Succ}_4] \geq t(\kappa) + \Pr[\neg E_0] \cdot \vartheta_A - \mathrm{negl}(\kappa)$$

by Lemma 4∼8, which means that the advantage that $\mathcal{A}$ can output a solution v of y is negligibly close to $\Pr[\neg E_0] \cdot \vartheta_A$. Since $\Pr[\neg E_0]$ is noticeable for any (even unbounded) algorithm, and $\mathcal{S}'$ will compute $w \leftarrow \mathcal{S}_4^{\mathsf{P.Orcl}(s_x,\cdot)}(st_1, v)$ after obtaining a solution v of y, we have that $\mathcal{S}'$ can output a solution w of x with non-negligible advantage by the requirement in Definition 4.

Second, if $\mathcal{S}$ is a $(1, F, I)$-CPRed, we have $\Pr[\mathsf{Succ}_4] = t(\kappa) + \mathrm{negl}(\kappa)$ by the requirement in Definition 4. Combining this with Lemma 4∼8, we have that $\Pr[\mathsf{Succ}_0] \leq \Pr[\mathsf{Succ}_4] + 2q_2 \sum_{j=1}^{dq_1} \sqrt{\delta_{1,j}} + \mathrm{negl}(\kappa) \leq t(\kappa) + 2dq_1q_2\sqrt{\delta_4} + \mathrm{negl}(\kappa)$, which implies that

$$\vartheta_A \leq 2dq_1q_2\sqrt{\delta_4} + \mathrm{negl}(\kappa).$$

Note that $\mathcal{S}'$ will compute $w \leftarrow \mathcal{S}_4^{\mathsf{P.Orcl}(s_x,\cdot)}(st_1, \hat{h})$ by first randomly measuring one of the QRO queries from $\mathcal{A}$ to obtain $\hat{h}$. The probability that $\hat{h}$ is a "bad" RO query is equal to δ_4 by definition, which is non-negligible. Thus, we have that $\mathcal{S}'$ can output a solution w of x with non-negligible advantage by the requirement in Definition 4. This completes the proof. □

5 Relation to Other BB-Reductions in the ROM

In this section, we investigate the relation of CPReds to other BB-reductions formalized by Fischlin et al. [13], i.e., fully-programming reduction (FPRed), randomly-programming reduction (RPRed), and non-programming reduction (NPRed). By definition, a NPRed implies a RPRed (i.e., NPRed ⇒ RPRed), which in turn implies a FPRed (i.e., RPRed ⇒ FPRed). Since the notion of FPRed essentially formalizes the standard concept of BB-reductions in the ROM, we immediately have that a CPRed implies a FPRed (i.e., CPRed ⇒ FPRed).

By carefully examining the OAEP encryption [3] and the FDH signature [4], we show that NPRed $\not\Rightarrow$ CPRed and CPRed $\not\Rightarrow$ RPRed, which implies that CPReds are incomparable to both NPReds and RPReds (see Fig. 2). We begin by first recalling the definition of trapdoor permutation (TDP).

A trapdoor permutation family $\Pi_{\mathrm{TDP}} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ with respect to length function $\ell(\cdot)$ consists of three algorithms. The parameter generator $\mathsf{Gen}(1^\kappa)$ takes a security parameter κ as input, outputs an index-trapdoor pair (s, td), i.e., $(s, td) \leftarrow \mathsf{Gen}(1^\kappa)$. The evaluation algorithm $\mathsf{Eval}(s, x)$ takes an index s and a string $x \in \{0,1\}^{\ell(\kappa)}$ as inputs, outputs a string $y \in \{0,1\}^{\ell(\kappa)}$, i.e., $y \leftarrow \mathsf{Eval}(s, x)$. The inversion algorithm $\mathsf{Inv}(td, y)$ takes a trapdoor td and a string $y \in \{0,1\}^{\ell(\kappa)}$ as inputs, outputs a string $x \in \{0,1\}^{\ell(\kappa)}$, i.e., $x \leftarrow \mathsf{Inv}(td, y)$.

For correctness, we require that for all $(s, td) \leftarrow \mathsf{Gen}(1^\kappa)$ and all $x \in \{0,1\}^{\ell(\kappa)}$, the equation $\mathsf{Inv}(td, \mathsf{Eval}(s, x)) = x$ always holds. Moreover, we say that $\Pi_{\mathrm{TDP}} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ is *one-way* if for all PPT algorithms $\mathcal{A}$, we have that

$$\Pr\Big[(s, td) \leftarrow \mathsf{Gen}(1^\kappa), x \xleftarrow{\$} \{0,1\}^{\ell(\kappa)}, x' \leftarrow \mathcal{A}(s, \mathsf{Eval}(s, x)) : x' = x\Big] = \mathrm{negl}(\kappa).$$

A trapdoor permutation $\Pi_{\mathrm{TDP}} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ is said to be *partial-domain one-way* with respect to $(\ell(\cdot), \ell_0(\cdot))$ if for all PPT algorithms $\mathcal{A}$, we have that

$$\Pr\left[\begin{array}{r}(s, td) \leftarrow \mathsf{Gen}(1^\kappa),\, x_0 \xleftarrow{\$} \{0,1\}^{\ell_0(\kappa)}, x_1 \xleftarrow{\$} \{0,1\}^{\ell(\kappa)-\ell_0(\kappa)}\\ x_0' \leftarrow \mathcal{A}(s, \mathsf{Eval}(s, x_0 \| x_1)) : x_0' = x_0\end{array}\right] = \mathrm{negl}(\kappa).$$

5.1 NPRed $\not\Rightarrow$ CPRed

In this subsection, we show that the OAEP encryption in [3] which was provably secure under NPReds is not provable under CPReds. Formally, let κ be the security parameter. Let $n_1, n_2, n_3 \geq \kappa$ be any positive integers. Let $\ell(\cdot)$ and $\ell_0(\cdot)$ be two arbitrary integer functions such that $\ell(\kappa) = \ell_0(\kappa) + n_3 = n_1 + n_2 + n_3$. Let $\Pi_{\mathrm{TDP}} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ be a trapdoor partial-domain one-way permutation family with respect to functions $(\ell(\cdot), \ell_0(\cdot))$. Let $G : \{0,1\}^{n_3} \to \{0,1\}^{n_1+n_2}$ and $H : \{0,1\}^{n_1+n_2} \to \{0,1\}^{n_3}$ be two hash functions. The PKE scheme $\Pi_{\mathrm{OAEP}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ given in [3] works as follows:

- $\mathsf{KeyGen}(1^\kappa)$: take a security parameter κ as input, compute and output the public and secret key pair $(pk, sk) = (s, td) \leftarrow \Pi_{\mathrm{TDP}}.\mathsf{Gen}(1^\kappa)$;
- $\mathsf{Enc}(pk, \mu)$: take the public key $pk = s$ and a message $\mu \in \{0,1\}^{n_1}$ as inputs, first append n_2 zeros to μ and obtain $\hat{\mu} = \mu \| 0_{n_2} \in \{0,1\}^{n_1+n_2}$. Then uniformly choose $r \xleftarrow{\$} \{0,1\}^{n_3}$, compute $x_0 = \hat{\mu} \oplus G(r)$, $x_1 = r \oplus H(x_0)$ and $y = \Pi_{\mathrm{TDP}}.\mathsf{Eval}(s, x_0 \| x_1)$. Finally, output the ciphertext $c = y$;
- $\mathsf{Dec}(sk, c)$: take the secret key $sk = td$ and a ciphertext $c = y \in \{0,1\}^{\ell(\kappa)}$ as inputs, first compute $x_0 \| x_1 = \Pi_{\mathrm{TDP}}.\mathsf{Inv}(td, y)$, $r = x_1 \oplus H(x_0)$ and $\hat{\mu} = x_0 \oplus G(r)$, where $x_0, \hat{\mu} \in \{0,1\}^{n_1+n_2}$ and $x_1 \in \{0,1\}^{n_3}$. Then, parse $\hat{\mu} = \mu \| \mu_0$ into $\mu \in \{0,1\}^{n_1}$ and $\mu_0 \in \{0,1\}^{n_2}$. If $\mu_0 \neq 0_{n_2}$, return $\perp$. Else return μ.

We have the following lemma which is implicit in [16,13].

Lemma 9. *Let $n_1, n_2, n_3 \geq \kappa$ be any positive integers. Let $(\ell(\cdot), \ell_0(\cdot))$ be two arbitrary integer functions such that $\ell(\kappa) = \ell_0(\kappa) + n_3 = n_1 + n_2 + n_3$. If the trapdoor permutation family Π_{TDP} is partial-domain one-way with respect to $(\ell(\cdot), \ell_0(\cdot))$, the PKE scheme Π_{OAEP} is provably CCA-secure under NPReds.*

As noted in [13], the reduction for the PKE scheme Π_{OAEP} given in [16] does not need to program the RO, and thus is a NPRed. Specifically, assuming that the trapdoor permutation family $\Pi_{\mathrm{TDP}} = (\mathsf{Gen}, , \mathsf{Eval}, \mathsf{Inv})$ is partial-domain one-way, Π_{PKE} is provably CCA-secure under NPReds.

We note that the reduction in [16] crucially relies on the knowledge of all RO query/response pairs made by the adversary to answer the decryption queries, which is not allowed for a CPRed (since the simulation of the decryption oracle is oblivious to the RO queries made by the adversary). Actually, we can show that the scheme Π_{OAEP} is not provable under CPReds.

Theorem 5. *Let $n_1, n_2, n_3 \geq \kappa$ be any positive integers. Let $(\ell(\cdot), \ell_0(\cdot))$ be any two integer functions such that $\ell(\kappa) = \ell_0(\kappa) + n_3 = n_1 + n_2 + n_3$. There is no CPRed from the CCA-security of Π_{OAEP} to the partial-domain one-wayness of the trapdoor permutation family Π_{TDP} with respect to $(\ell(\cdot), \ell_0(\cdot))$.*

We first give a sketch of the proof. Technically, we will use the two-oracle separation technique of Hsiao and Reyzin [20] by giving two oracles Λ and Ω such that 1) Λ is an ideal trapdoor permutation and Ω is a "breaking" oracle; 2) there is a PPT adversary $\mathcal{A}^{\Lambda,\Omega}$ which can break the CCA2-security of the instantiation $\Pi^{\Lambda}_{\mathrm{OAEP}}$ of Π_{OAEP} using Λ; and 3) no PPT CPRed $\mathcal{S}$, given access to Λ and $\mathcal{A}^{\Lambda,\Omega}$ (i.e., $\mathcal{S}^{\Lambda,\mathcal{A}^{\Lambda,\Omega}}$), can break the partial-domain one-wayness of Λ. The basic idea is that $\mathcal{S}$ can only access the "breaking" oracle Ω by interacting with $\mathcal{A}$ which will only use Ω after he is convinced that Ω is "useless" to $\mathcal{S}$ (which needs $\mathcal{S}$ to invert the permutation and correctly answer a random decryption query without "seeing" the RO queries). In other words, the ability of $\mathcal{A}$ to access the breaking oracle Ω cannot be utilized by $\mathcal{S}$ to break the partial-domain one-wayness of Λ.

Proof. We will use the two-oracle separation technique of Hsiao and Reyzin [20] to prove this theorem by giving two oracles Λ and Ω such that 1) Λ is a family of ideal trapdoor permutations and Ω is a "breaking" oracle; 2) there is a PPT adversary $\mathcal{A}^{\Lambda,\Omega}$ which can break the CCA2-security of $\Pi^{\Lambda}_{\mathrm{OAEP}}$ which is an instantiation of Π_{OAEP} using Λ; and 3) no PPT CPRed $\mathcal{S}$, given access to Λ and $\mathcal{A}^{\Lambda,\Omega}$ (i.e., $\mathcal{S}^{\Lambda,\mathcal{A}^{\Lambda,\Omega}}$), can break the partial-domain one-wayness of Λ.

Formally, let $\mathcal{E} = (\mathcal{E}_{\mathsf{Perm}}, \mathcal{E}_E, \mathcal{E}_{E^{-1}})$ be an oracle which is initialized with a random permutation $\mathsf{Perm} : \{0,1\}^\kappa \to \{0,1\}^\kappa$ and a keyed family of random permutations $E : \{0,1\}^\kappa \times \{0,1\}^{\ell(\kappa)} \to \{0,1\}^{\ell(\kappa)}$ (i.e., given a key $s \in \{0,1\}^\kappa$, $E(s,\cdot) : \{0,1\}^{\ell(\kappa)} \to \{0,1\}^{\ell(\kappa)}$ is a random permutation). The two interfaces $\mathcal{E}_{\mathsf{Perm}}$ and $\mathcal{E}_E$ allow to evaluate Perm and E, respectively, whereas the interface $\mathcal{E}_{E^{-1}}$ allows to evaluate the inversion E^{-1} of E, i.e., $E^{-1}(s, E(s,x)) = x$. Now, we define the first oracle $\Lambda = (\mathsf{Gen}^{\mathcal{E}}, \mathsf{Eval}^{\mathcal{E}}, \mathsf{Inv}^{\mathcal{E}})$ consisting of three interfaces:

– $\mathsf{Gen}^{\mathcal{E}}(1^\kappa)$: first randomly choose $td \xleftarrow{\$} \{0,1\}^\kappa$, then send td to $\mathcal{E}_{\mathsf{Perm}}$ and obtain a response $s = \mathsf{Perm}(td)$. Finally, return (s, td), i.e., $(s, td) \leftarrow \mathsf{Gen}^{\mathcal{E}}(1^\kappa)$;

– $\mathsf{Eval}^{\mathcal{E}}(s,x)$: first send $(s,x) \in \{0,1\}^\kappa \times \{0,1\}^{\ell(\kappa)}$ to $\mathcal{E}_E$ and obtain a response $y = E(s,x)$. Then, return $y \in \{0,1\}^{\ell(\kappa)}$, i.e., $y \leftarrow \mathsf{Eval}^{\mathcal{E}}(s,x)$;
– $\mathsf{Inv}^{\mathcal{E}}(td,y)$: first send $td \in \{0,1\}^\kappa$ to $\mathcal{E}_{\mathsf{Perm}}$ and obtain a response $s = \mathsf{Perm}(td)$. Then, send $(s,y) \in \{0,1\}^\kappa \times \{0,1\}^{\ell(\kappa)}$ to $\mathcal{E}_{E^{-1}}$ and obtain a response $x = E^{-1}(s,y)$. Finally, return $x \in \{0,1\}^{\ell(\kappa)}$, i.e., $x \leftarrow \mathsf{Inv}^{\mathcal{E}}(td,y)$.

We first show that Λ is a family of ideal trapdoor permutations, i.e., no PPT algorithm $\mathcal{S}^\Lambda$ given oracle access to Λ can break the partial-domain one-wayness of Λ with respect to functions $\ell(\kappa) \geq \ell_0(\kappa) \geq \kappa$. Note that given a challenge pair $s^* = \mathsf{Perm}(td^*)$ and $y^* = E(s^*, x^*)$ as inputs, $\mathcal{S}^\Lambda$ is asked to output the first $\ell_0(\kappa)$-bit of x^* by making at most a polynomial number of queries to Λ. First, since Perm is a random permutation and $td^* \in \{0,1\}^\kappa$ is uniformly chosen at random, the probability that $\mathcal{S}^\Lambda$ can find td^* is negligible, which means that the inversion oracle $\Lambda.\mathsf{Inv}^{\mathcal{E}}$ cannot help $\mathcal{S}^\Lambda$ to invert $E(s^*,\cdot)$. Second, since $E(s^*,\cdot)$ is a random permutation, conditioned on that $\mathcal{S}^\Lambda$ cannot find td^*, the probability that $\mathcal{S}^\Lambda$ can output the first $\ell_0(\kappa)$-bit of x^* is also negligible (because it can only determine the first $\ell_0(\kappa)$-bit of x^* by making a query (s^*, x^*) to $\Lambda.\mathsf{Eval}^{\mathcal{E}}(\cdot,\cdot)$). This shows that Λ is partial-domain one-way.

The second oracle $\Omega = (\mathcal{R}_1, \mathcal{R}_2, \mathsf{Inv}^{\mathcal{E}})$ also consists of three interfaces, where $\mathcal{R}_1(\cdot)$ evaluates a random function from $\{0,1\}^{2^\kappa}$ to $\{0,1\}^{n_1}$, $\mathcal{R}_2(\cdot)$ evaluates a random function from $\{0,1\}^{2^\kappa}$ to $\{0,1\}^{n_3}$, and $\mathsf{Inv}^{\mathcal{E}}(\cdot,\cdot)$ simply relays its query to the oracle $\mathcal{E}$ via the interface $\mathcal{E}_{E^{-1}}$ and returns whatever it obtains from the oracle. Now, we give an adversary $\mathcal{A}^{\Lambda,\Omega}$ breaking the CCA2-security of the Π^Λ_{PKE}, but no CPRed $\mathcal{S}^{\Lambda,\mathcal{A}^{\Lambda,\Omega}}$ can break the partial-domain one-wayness of Λ.

Description of $\mathcal{A}^{\Lambda,\Omega}$. Given a security parameter κ and a public key $pk = s$ of $\Pi^\Lambda_{\text{OAEP}}$ as inputs, the goal of $\mathcal{A}^{\Lambda,\Omega}$ is to break the CCA-security of $\Pi^\Lambda_{\text{OAEP}}$. The adversary $\mathcal{A}^{\Lambda,\Omega}$ which is also given access to an G-oracle and H-oracle used by the scheme $\Pi^\Lambda_{\text{OAEP}}$ works as follows:

1. Let $V = \kappa \| s$ (i.e., the initial view of $\mathcal{A}^{\Lambda,\Omega}$);
2. Send $r = \Omega.\mathcal{R}_2(V)$ to the G-oracle and obtain a response z_1;
3. Update $V := V \| z_1$, and compute $\mu = \Omega.\mathcal{R}_1(V) \in \{0,1\}^{n_1}$;
4. Send $x_0 = (\mu \| 0_{n_2}) \oplus z_1$ to the H-oracle and obtain a response z_2;
5. Send $y = \Lambda.\mathsf{Eval}^{\mathcal{E}}(s, x_0 \| x_1))$ to the decryption oracle and obtain a result μ', where $x_1 = r \oplus z_2$;
6. If $\mu \neq \mu'$, output $\perp$ and abort;
7. Update $V := V \| z_2 \| \mu'$, compute $\mu_0^* = \Omega.\mathcal{R}_1(V\|0), \mu_1^* = \Omega.\mathcal{R}_1(V\|1)$, and send (μ_0^*, μ_1^*) to obtain a challenge ciphertext y^*;
8. Compute $x^* = x_0^* \| x_1^* = \Omega.\mathsf{Inv}^{\mathcal{E}}(s, y^*)$, send $x_0^* \in \{0,1\}^{n_1+n_2}$ to the oracle for H and obtain a response z_2^*;
9. Send $r^* = x_1^* \oplus z_2^* \in \{0,1\}^{n_3}$ to the oracle for G and obtain a response z_1^*;
10. If $z_1^* \oplus x_0^* = \mu_0^* \| 0_{n_2}$, output 0, else output 1.

Note that no matter how the oracles G and H are instantiated, $\mathcal{A}^{\Lambda,\Omega}$ can always break the CCA2-security of $\Pi^\Lambda_{\text{OAEP}}$. Let (s^*, y^*) be the challenge pair of the partial-domain one-wayness of Λ. We now show that no CPRed $\mathcal{S}(s^*, y^*)$

given oracle access to Λ and $\mathcal{A}^{\Lambda,\Omega}$ can find the first $\ell_0(\kappa)$-bit of $x^* = E^{-1}(s^*, y^*)$ with non-negligible probability. The basic idea is that the adversary $\mathcal{A}^{\Lambda,\Omega}$ will not use the inversion oracle $E^{-1}(s^*, \cdot)$ until it has been convinced that $\mathcal{S}^{\Lambda}$ can invert the function $E(s^*, \cdot)$, which is infeasible for $\mathcal{S}^{\Lambda}$ given only oracle access to Λ.

By Definition 4, a CPRed $\mathcal{S}^{\Lambda} = (\mathcal{S}_1^{\Lambda}, \mathcal{S}_2^{\Lambda}, \mathcal{S}_3^{\Lambda}, \mathcal{S}_4^{\Lambda})$ is a single-instance reduction and will not rewind the adversary $\mathcal{A}^{\Lambda,\Omega}$. Moreover, given a security parameter κ and a challenge pair (s^*, y^*) as inputs, $\mathcal{S}^{\Lambda} = (\mathcal{S}_1^{\Lambda}, \mathcal{S}_2^{\Lambda}, \mathcal{S}_3^{\Lambda}, \mathcal{S}_4^{\Lambda})$ works as follows:[7] First, it runs $(pk, st_1) \leftarrow \mathcal{S}_1^{\Lambda}(1^{\kappa}, (s^*, y^*))$ to obtain a public key pk and a state st_1. Second, it runs $st_{2,0} \leftarrow \mathcal{S}_2^{\Lambda}(pk, st_1)$ to obtain a state $st_{2,0}$ by invoking the adversary $\mathcal{A}^{\Lambda,\Omega}$ and answering the decryption queries from $\mathcal{A}^{\Lambda,\Omega}$. Third, it runs $st_{2,1} \leftarrow \mathcal{S}_3^{\Lambda}(st_1)$ to simulate the ROs for G, H and obtain a state $st_{2,1}$ which is either an empty string ϵ or one of the oracle queries to G and H. Finally, it runs $\mathcal{S}_4^{\Lambda}(st_1, st_{2,b})$ to obtain a candidate solution x'.

Note that if $\mathcal{A}^{\Lambda,\Omega}$ does not use s^* as the first input in a query to $\Omega.\mathsf{Inv}^{\mathcal{E}}(\cdot, \cdot)$ (and thus does not use s^* as the first input to the inversion oracle $E^{-1}(\cdot, \cdot)$), then $\mathcal{S}^{\Lambda}$ cannot obtain any advantage from $\mathcal{A}^{\Lambda,\Omega}$ to invert $y^* = E(s^*, x^*)$. This is because for any $s \neq s^*$, $E(s^*, \cdot)$ is a random permutation which is independent from $E(s, \cdot)$. Since $\mathcal{A}^{\Lambda,\Omega}$ will only make a single query using $pk = s$ as the first input to $\Omega.\mathsf{Inv}^{\mathcal{E}}(\cdot, \cdot)$ after $\mathcal{S}_2^{\Lambda}$ correctly answers a random decryption query, it suffices to show that $\mathcal{S}_2^{\Lambda}$ cannot correctly answer the decryption query from $\mathcal{A}^{\Lambda,\Omega}$ with non-negligible probability when $s = s^*$.

Recall that given a security parameter κ and a public key $pk = s^*$ as inputs, $\mathcal{A}^{\Lambda,\Omega}$ works as follows: 1) set $V = \kappa \| s^*$ and compute $r = \Omega.\mathcal{R}_2(V)$; 2) send r as a G-oracle query to $\mathcal{S}_3^{\Lambda}(st_1)$ and obtain a response z_1; 3) update $V := V \| z_1$, and compute $\mu = \Omega.\mathcal{R}_1(V) \in \{0,1\}^{n_1}$; 4) send $x_0 = (\mu \| 0_{n_2}) \oplus z_1$ as an H-oracle query to $\mathcal{S}_3^{\Lambda}(st_1)$ and obtain a response z_2; 5) send $y = \Lambda.\mathsf{Eval}^{\mathcal{E}}(s, x_0 \| x_1))$ as a decryption query to $\mathcal{S}_2^{\Lambda}(pk, st_1)$ and obtain a result μ', where $x_1 = r \oplus z_2$. We now show that the probability that $\mu' = \mu$ is negligible. First, by the fact that $\Omega.\mathcal{R}_1$ and $\Omega.\mathcal{R}_2$ evaluates random functions, we have that r and μ are both uniformly random, and in particular is independent from the inputs (st_1, y) of $\mathcal{S}_2^{\Lambda}$. Second, since $\mathcal{S}_2^{\Lambda}$ is unaware of the RO queries made to G and H when answering the decryption query $y = \Lambda.\mathsf{Eval}^{\mathcal{E}}(s, x_0 \| x_1)$, and

$$x_0 \| x_1 = ((\mu \| 0_{n_2}) \oplus z_1) \| (r \oplus z_2) = ((\mu \| 0_{n_2}) \oplus G(r)) \| (r \oplus H(x_0))$$

has min-entropy at least $2^{n_1} \geq 2^{\kappa}$ in the view of $\mathcal{S}_2^{\Lambda}$ (since μ is uniformly distributed over $\{0,1\}^{n_1}$), the probability that $\mathcal{S}_2^{\Lambda}$ can output $\mu' = \mu$ by making

[7] Here, we allow the sub-algorithm $\mathcal{S}_3$ to access to the oracle Λ, which does not conflict with the restriction that "$\mathcal{S}_3$ does not have access to $\mathrm{P}.\mathsf{Oral}(s_x, \cdot)$" in Definition 3, since Λ is an oracle which implement the trapdoor permutations and is assumed to be publicly known to all parties, while $\mathrm{P}.\mathsf{Oral}(s_x, \cdot)$ is assumed to be a private oracle which can only be accessed by the party who owns the secret value s_x. In fact, the problem we considered here is to invert the trapdoor permutations (i.e., the partial-domain one-wayness of Λ), which is a non-interactive problem (see Sec. 5), i.e., the corresponding oracle "$\mathrm{P}.\mathsf{Oral}(s_x, \cdot)$" is actually a dummy one $\mathrm{P}.\mathsf{Oral}(s_x, \cdot) = \perp$.

a polynomial number of queries to Λ is negligible (because $\mathcal{S}_2^{\Lambda}$ can only determine μ by making a query $(s^*, x_0 \| x_1)$ to $\Lambda.\mathsf{Eval}^{\mathcal{E}}(\cdot, \cdot)$). In other words, $\mathcal{A}^{\Lambda,\Omega}$ will abort at step 6 with probability negligibly close to 1. Thus, $\mathcal{S}^{\Lambda}$ can only obtain negligible advantage from $\mathcal{A}^{\Lambda,\Omega}$ in inverting $y^* = E(s^*, x^*)$, which completes the proof of Theorem 5. $\square$

5.2 CPRed $\nRightarrow$ RPRed

In this subsection, we show that the FDH signature in [4] which was shown not to be provable under RPReds is provably secure under CPReds. Formally, let κ be the security parameter, and let ℓ_μ be positive integer. Let $\ell(\cdot)$ be an integer function. Let $\Pi_{\mathrm{TDP}} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ be a trapdoor one-way permutation family with respect to $\ell(\cdot)$. Let $H : \{0,1\}^{\ell_\mu} \to \{0,1\}^{\ell(\kappa)}$ be a hash function. The signature scheme $\Pi_{\mathrm{FDH}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ given in [4] works as follows:

- $\mathsf{KeyGen}(1^\kappa)$: take a security parameter κ as input, compute and return the verification and signing key pair $(vk, sk) = (s, td) \leftarrow \Pi_{\mathrm{TDP}}.\mathsf{Gen}(1^\kappa)$;
- $\mathsf{Sign}(sk, \mu)$: take the signing key $sk = td$ and a message $\mu \in \{0,1\}^{\ell_\mu}$ as inputs, compute $y = H(\mu)$ and return the signature $\sigma = x = \Pi_{\mathrm{TDP}}.\mathsf{Inv}(td, y)$;
- $\mathsf{Verify}(vk, \mu, \sigma)$: take the verification key $vk = s$, a message $\mu \in \{0,1\}^{\ell_\mu}$ and a signature $\sigma = x \in \{0,1\}^{\ell(\kappa)}$ as inputs, first compute $y = H(\mu) \in \{0,1\}^{\ell(\kappa)}$. Then, returns 1 if $y = \Pi_{\mathrm{TDP}}.\mathsf{Eval}(s, x)$, else return 0.

We have the following lemma which is implicit in [13].

Lemma 10. *There is no RPRed from the EUF-CMA security of the signature scheme Π_{FDH} to the one-wayness of the trapdoor permutation family Π_{TDP}.*

This lemma directly follows from two facts in [13]: 1) a RPRed implies a reduction in the weakly programming ROM (WPROM); and 2) Π_{FDH} is not provable even against key-only attacks in the WPROM. We omit the details.

Theorem 6. *If the trapdoor permutation family Π_{TDP} is one-way, the signature scheme Π_{FDH} is provably EUF-CMA secure under CPReds.*

We first give a sketch of the proof. Note that given an index s and a challenge point y^*, the CPRed $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$ is asked to output a preimage x^* of y^* such that $y^* = \Pi_{\mathrm{TDP}}.\mathsf{Eval}(s, x^*)$. Technically, given a query μ to the RO $\mathcal{O}(\cdot)$, the reduction $\mathcal{S}$ sets $\mathcal{O}(\mu) = y^*$ with certain probability $1/p$ for some $p \geq 1$, and hopes that the adversary will use μ in creating a forgery $(\mu^* = \mu, \sigma^*)$, otherwise it sets $\mathcal{O}(\mu) = \Pi_{\mathrm{TDP}}.\mathsf{Eval}(s, x)$ with probability $1 - 1/p$ by randomly choosing $x \overset{\$}{\leftarrow} \{0,1\}^{\ell(\kappa)}$, and hopes that the adversary will use μ in a signing query. Clearly, this strategy does not need to know the actual value of μ in advance, and can be done by using k-wise independent functions (i.e., replacing the randomly chosen $x \overset{\$}{\leftarrow} \{0,1\}^{\ell(\kappa)}$ with $x = f(\mu)$ for an appropriate k-wise independent function f).

Proof. We now give a CPRed $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$ from the EUF-CMA security of $\mathit{\Pi}_{\mathrm{FDH}}$ to the one-wayness of $\mathit{\Pi}_{\mathrm{TDP}}$. Note that given an index s^* and a uniformly random $y^* \in \{0,1\}^{\ell(\kappa)}$ as inputs, the goal of $\mathcal{S}$ is to output a preimage x^* of y^* such that $y^* = \mathit{\Pi}_{\mathrm{TDP}}.\mathsf{Eval}(s^*, x^*)$. Let $n_1, n_2 \geq \kappa$ be positive integers. Let $\mathcal{A}$ be an adversary which breaks the existential unforgeability under chosen message attacks (EUF-CMA) of $\mathit{\Pi}_{\mathrm{FDH}}$ with probability at least ϑ_A by making at most q_r RO queries and q_s signing queries. Now, we are ready to describe the reduction $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$:

– $\mathcal{S}_1$ takes a security parameter κ, an index s^* and a uniformly random $y^* \in \{0,1\}^{\ell(\kappa)}$ as inputs, randomly chooses two k-wise independent hash functions $f_1 : \{0,1\}^{\ell_\mu} \to \{1,\ldots,p\}$ and $f_2 : \{0,1\}^{\ell_\mu} \to \{0,1\}^{\ell(\kappa)}$, where $k, p \in \mathbb{Z}$ will be determined later. Finally, outputs s^* and $st_1 = (f_1, f_2, s^*, y^*)$, i.e., $(s^*, st_1) \leftarrow \mathcal{S}_1(1^\kappa, s^*, y^*)$;
– $\mathcal{S}_2$ takes an index s^* and a state $st_1 = (f_1, f_2, s^*, y^*)$ as inputs, invokes $\mathcal{A}$ with input s^*. Whenever receiving a signing query $\mu \in \{0,1\}^{\ell_\mu}$ from $\mathcal{A}$, the algorithm $\mathcal{S}_2$ returns $\perp$ if $f_1(\mu) = 1$, else returns a signature $\sigma = f_2(\mu) \in \{0,1\}^{\ell(\kappa)}$. Finally, if $\mathcal{A}$ outputs a forgery (μ^*, σ^*) and abort, $\mathcal{S}_2$ outputs $st_{2,0} = (\mu^*, \sigma^*)$;
– $\mathcal{S}_3$ takes a state $st_1 = (f_1, f_2, s^*, y^*)$ as input, sets $st_{2,1} = \epsilon$ and generates a response $y = F(st_1, \mu)$ to a RO query $\mu \in \{0,1\}^{\ell_\mu}$ from $\mathcal{A}$, where

$$F(st_1, \mu) = \begin{cases} y^*, & \text{if } f_1(\mu) = 1 \\ \mathit{\Pi}_{\mathrm{TDP}}.\mathsf{Eval}(s^*, f_2(\mu)), & \text{otherwise} \end{cases};$$

– $\mathcal{S}_4$ takes states $st_1 = (f_1, f_2, s^*, y^*)$ and $st_{2,0} = (\mu^*, \sigma^*)$ as inputs, outputs $x^* = \sigma^*$ if $f_1(\mu^*) = 1 \wedge \mathit{\Pi}_{\mathrm{TDP}}.\mathsf{Eval}(s^*, \sigma^*) = y^*$, else outputs $\perp$.

Let $\mathcal{M} = \{\mu_1, \ldots, \mu_{q_r+q_s}\}$ be all messages that $\mathcal{A}$ used in both the RO queries and the signing queries. Let E_0 be the event that $\mathcal{S}_2$ returns a failure symbol $\perp$ to a signing query or there exists a $\mu \in \mathcal{M}\backslash\{\mu^*\}$ satisfying $f_1(\mu) = 1$ (i.e., the adversary finds a collision that $F(st_1, \mu) = F(st_1, \mu^*)$ which will happen with negligible probability for a real RO). Clearly, the event E_0 will not happen if $f_1(\mu^*) = 1$ and $f_2(\mu) \neq 1$ for all $\mu \in \mathcal{M}\backslash\{\mu^*\}$. Moreover, conditioned on that the event E_0 does not happen, $\mathcal{A}$'s view is almost identical to that in the real game. For simplification, we first assume that both f_1 and f_2 are real random functions. In this case, we have that that $\Pr[\neg E_0] \geq \frac{1}{p} \cdot (1 - \frac{1}{p})^{q_r+q_s-1}$ holds. Thus, $\mathcal{A}$ will output a signature σ^* on μ^* (i.e., $\mathit{\Pi}_{\mathrm{TDP}}.\mathsf{Eval}(s, \sigma^*) = y^*$) with probability at least $\vartheta' = \Pr[\neg E] \cdot \vartheta_A = \frac{1}{p} \cdot (1 - \frac{1}{p})^{q_r+q_s-1} \cdot \vartheta_A$. Since the value $\frac{1}{p} \cdot (1 - \frac{1}{p})^{q_r+q_s-1}$ is maximal when $p = q_r + q_s \geq 1$, we have that $\vartheta' \geq \frac{1}{4p}\vartheta_A$ always holds. In other words, $\mathcal{S}$ can break the one-wayness of $\mathit{\Pi}_{\mathrm{TDP}}$ with probability at least ϑ'. By instantiating f_1 and f_2 with $(q_r + q_s)$-wise independent functions, we have that $\mathcal{S}$ can break the one-wayness of $\mathit{\Pi}_{\mathrm{TDP}}$ with probability at least ϑ', since $(q_r + q_s)$-wise independent functions are equivalent to random functions up to $(q_r + q_s)$-queries, and f_1, f_2 will be evaluated at most $q_r + q_s$ times by $\mathcal{S}$. Thus, by setting $k = q_r + q_s, p = q_r + q_s$, we have that $\mathcal{S}$ is a $(0, F, \emptyset)$-CPRed. This completes the proof of Theorem 6. □

References

1. Ananth, P., Bhaskar, R.: Non observability in the random oracle model. In: Susilo, W., Reyhanitabar, R. (eds.) Provable Security 2013. pp. 86–103. Springer, Heidelberg (2013)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS 1993. pp. 62–73. ACM (1993)
3. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) Advances in Cryptology – EUROCRYPT'94. pp. 92–111. Springer, Heidelberg (1995)
4. Bellare, M., Rogaway, P.: The exact security of digital signatures-how to sign with RSA and Rabin. In: Maurer, U. (ed.) Advances in Cryptology — EUROCRYPT '96. pp. 399–416. Springer, Heidelberg (1996)
5. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. Theoretical Computer Science 560, 7 – 11 (2014)
6. Biham, E., Boyer, M., Boykin, P.O., Mor, T., Roychowdhury, V.: A proof of the security of quantum key distribution. Journal of Cryptology 19(4), 381–439 (Oct 2006)
7. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D., Wang, X. (eds.) ASIACRYPT 2011, pp. 41–69. Springer, Heidelberg (2011)
8. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001, pp. 213–229. Springer (2001)
9. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013. pp. 361–379. Springer, Heidelberg (2013)
10. Brassard, G., HØyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN'98: Theoretical Informatics. pp. 163–169. Springer, Heidelberg (1998)
11. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM 51(4), 557–594 (Jul 2004)
12. Dagdelen, Ö., Fischlin, M., Gagliardoni, T.: The Fiat–Shamir transformation in a quantum world. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology - ASIACRYPT 2013. pp. 62–81. Springer, Heidelberg (2013)
13. Fischlin, M., Lehmann, A., Ristenpart, T., Shrimpton, T., Stam, M., Tessaro, S.: Random oracles with(out) programmability. In: Abe, M. (ed.) Advances in Cryptology – ASIACRYPT 2010. pp. 303–320. Springer, Heidelberg (2010)
14. Fuchs, C.A., van de Graaf, J.: Cryptographic distinguishability measures for quantum-mechanical states. IEEE Transactions on Information Theory 45(4), 1216–1227 (May 1999)
15. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013)
16. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: RSA-OAEP is secure under the RSA assumption. Journal of Cryptology 17(2), 81–104 (Mar 2004)
17. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. pp. 197–206. ACM (2008)
18. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: STOC 1996. pp. 212–219. ACM (1996)
19. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography. pp. 341–371. Springer International Publishing, Cham (2017)

20. Hsiao, C.Y., Reyzin, L.: Finding collisions on a public road, or do secure hash functions need secret coins? In: Franklin, M. (ed.) Advances in Cryptology – CRYPTO 2004. pp. 92–105. Springer, Heidelberg (2004)
21. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisted. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018. pp. 96–125. Springer,Cham (2018)
22. Karloff, H., Mansour, Y.: On construction of k-wise independent random variables. Combinatorica 17(1), 91–107 (Mar 1997)
23. Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018. pp. 552–586. Springer International Publishing, Cham (2018)
24. Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) Theory of Cryptography. pp. 21–39. Springer, Heidelberg (2004)
25. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: Yung, M. (ed.) Advances in Cryptology – CRYPTO 2002. pp. 111–126. Springer, Heidelberg (2002)
26. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2000)
27. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018. pp. 520–551. Springer, Cham (2018)
28. Shor, P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5), 1484–1509 (1997)
29. Shoup, V.: A proposal for an ISO standard for public key encryption. Cryptology ePrint Archive, Report 2001/112 (2001)
30. Targhi, E.E., Unruh, D.: Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In: Hirt, M., Smith, A. (eds.) Theory of Cryptography, pp. 192–216. Springer, Heidelberg (2016)
31. Unruh, D.: Quantum position verification in the random oracle model. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology – CRYPTO 2014. pp. 1–18. Springer, Heidelberg (2014)
32. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015. pp. 755–784. Springer, Heidelberg (2015)
33. Unruh, D.: Revocable quantum timed-release encryption. J. ACM 62(6), 49:1–49:76 (Dec 2015)
34. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012, pp. 758–775. Springer, Heidelberg (2012)

A Randomly-Programming Reduction

The notion of randomly-programming reduction (RPRed) is formalized via the notion of randomly programmable random oracle (RPRO). Formally, a RPRO $\mathcal{O} = (R_{eval}, R_{rand}, R_{prog})$ is an idealized object consisting of a public interface R_{eval}, and two private interfaces R_{rand} and R_{prog}, where R_{eval} behaves as a conventional RO mapping $Dom \to Rng$; R_{rand} maps a string in $\{0,1\}^*$ to a random value in Rng; and $R_{prog}(X, Y)$ takes an $X \in Dom$ and a string $Y \in \{0,1\}^*$ as inputs, sets $R_{eval}(X) = R_{rand}(Y)$. As NPReds, a RPRed (i.e., a black-box reduction in the model equipped with a RPRO) is allowed to learn all the RO-queries made by the adversary $\mathcal{A}$ via the public interface R_{eval}, but unlike NPReds, a RPRed is also allowed to program the answer of a given RO-query X made by the adversary $\mathcal{A}$ via the two private interfaces R_{rand} and R_{prog}. Specifically, after receiving a RO query X from $\mathcal{A}$, the reduction can make a number of queries to R_{rand} and R_{prog} before returning the answer $R_{eval}(X)$ (which can be programmed by using the interface R_{prog}) to $\mathcal{A}$.

B Information Theory

In this section, we recall some definitions related to the Shannon entropy of random variables. Formally, let X, Y be two random variables with support D_X, D_Y, respectively. The entropy of X is defined as

$$H(X) = -\sum_{x \in D_X} \Pr[X = x] \log_2(\Pr[X = x]).$$

The entropy of X conditioned on $Y = y$ is defined as

$$H(X|y) = -\sum_{x \in D_X} \Pr[X = x|y] \log_2(\Pr[X = x|y]).$$

The entropy of X conditioned on random variable Y is defined as

$$H(X|Y) = \sum_{y \in D_Y} \Pr[Y = y] H(X|y).$$

The mutual information between X and Y is defined as

$$I(X, Y) = H(X) - H(X|Y).$$

Intuitively, the mutual information indicates the decrease in the entropy of X due to learning of Y, which is symmetric to X and Y.