# Interactive Proofs for Quantum Black-Box Computations

Jiang Zhang[*] Yu Yu[†] Dengguo Feng[‡] Shuqin Fan[§] Zhenfeng Zhang[¶]
Kang Yang[‖]

### Abstract

In this paper, we initiate the study of interactive proofs for the promise problem $\mathsf{QBBC}$ (i.e., quantum black-box computations), which consists of a quantum device $\mathcal{D}$ acting on $(n+m)$ qubits, a classical device $\mathcal{R}_F$ deciding the input-output relation of some unknown function $F : \{0,1\}^n \to \{0,1\}^m$, and two reals $0 < b < a \leq 1$. Let $p(\mathcal{D}, x) = \| \left|x, F(x)\right\rangle \left\langle x, F(x)\right| \mathcal{D}(\left|x\right\rangle \left|0^m\right\rangle)\|^2$ be the probability of obtaining $(x, F(x))$ as a result of a standard measurement of the $(n+m)$-qubit state returned by $\mathcal{D}$ on input $\left|x\right\rangle \left|0^m\right\rangle$. The task of the problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ is to distinguish between two cases for all $x \in \{0,1\}^n$:

- YES Instance: $p(\mathcal{D}, x) \geq a$;
- NO Instance: $p(\mathcal{D}, x) \leq b$.

First, we show that for any constant $15/16 < a \leq 1$, the problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ has an efficient two-round interactive proof $(\mathcal{P}^{\mathcal{D}}, \mathcal{V}^{\mathcal{R}_F})$ which basically allows a verifier $\mathcal{V}$, given a classical black-box device $\mathcal{R}_F$, to efficiently verify if the prover $\mathcal{P}$ has a quantum black-box device $\mathcal{D}$ (correctly) computing $F$. This proof system achieves completeness $\frac{1+a}{2}$ and soundness error $\frac{31}{32} + \frac{\epsilon}{2} + \mathsf{negl}(n)$ for any constant $\max(0, b - \frac{15}{16}) < \epsilon < a - \frac{15}{16}$, given that the verifier $\mathcal{V}$ has some (limited) quantum capabilities. In terms of query complexities, the prover $\mathcal{P}^{\mathcal{D}}$ will make at most two quantum queries to $\mathcal{D}$, while the verifier $\mathcal{V}^{\mathcal{R}_F}$ only makes a single classical query to $\mathcal{R}_F$. This result is based on an information versus disturbance lemma, which may be of independent interest.

Second, under the learning with errors (LWE) assumption in (Regev 2005), we show that the problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ can even have an efficient interactive proof $(\mathcal{P}^{\mathcal{D}}, \mathcal{V}^{\mathcal{R}_F})$ with a fully classical verifier $\mathcal{V}$ that does not have any quantum capability. This proof system achieves completeness $\frac{1+a}{2} - \mathsf{negl}(n)$ and soundness error $\frac{1+b}{2} + \mathsf{negl}(n)$, and thus applies to any $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ with constants $0 < b < a \leq 1$. Moreover, this proof system has the same query complexities as above. This result is based on the techniques introduced in (Brakerski et al. 2018) and (Mahadev 2018).

As an application, we show that the problem of distinguishing the random oracle model (ROM) and the quantum random oracle model (QROM) in cryptography can be naturally seen as a $\mathsf{QBBC}$ problem. By applying the above result, we immediately obtain a separation between ROM and QROM under the standard LWE assumption.

## 1 Introduction

In the coming decades, the quantum technology advancement is promising to reshape the computing landscape. Before the technology to build universal quantum computers becomes available, it is likely that some dedicated quantum devices (such as those being developed by Google, IBM, IonQ and Rigetti) will be available to users via cloud platforms. This raises a natural

[*] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China. Email:jiangzhang09@gmail.com.
[†] Shanghai Jiao Tong University, Shanghai 200240, China. Email: yuyuathk@gmail.com.
[‡] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China. Email:feng@tca.iscas.ac.cn.
[§] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China. Email:shuqinfan78@163.com.
[¶] Institute of Software, Chinese Academy of Sciences, China. Email:zfzhang@tca.iscas.ac.cn.
[‖] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China. Email:yangk@sklc.org.

question: whether it is possible to verify a quantum computation performed on an untrusted server [AV13]. This question has become increasingly important and has already received substantial attention from the community [GKK19]. Although it remains open to find the ultimate solution allowing a classical verifier to verify any BQP computation performed on a single untrusted server, plenty of works considering two weaker settings have appeared. The first setting considers a limited quantum verifier interacting with a single quantum prover, where the verifier can only perform limited quantum operations or store a few constant qubits. The verifier must use those limited quantum capabilities (and a quantum communication channel) to delegate computation to a quantum server, and to ensure that the server indeed performs the correct quantum computation [ABOE10, HM15, MF16, ABOEM17, FK17, GWK17, HKSE17, MTH17, Bro18, TM18, HT19, ZH19a, ZH19b, ZZ20]. The second setting considers a fully classical verifier interacting with multiple non-communicating quantum servers, where the servers are allowed to share many entangled states before the computation, but cannot communicate with each other during the computation. The verifier typically plays some non-local games (e.g., the CHSH game [CHSH69]) with the non-communicating servers to characterize the servers' behaviors in the computation [RUV13, GKW15, HPDF15, Ji16, McK16, NV16, FHM18, CGJV19, Gri19].

In a recent work, Mahadev [Mah18a] presented a protocol allowing a classical verifier to delegate the standard and Hadamard measurements to an efficient quantum device in a verifiable way, under the hardness of the learning with errors (LWE) problem [Reg05]. Specifically, as long as there is no quantum polynomial time (QPT) algorithm solving the LWE problem (which is a common belief in post-quantum cryptography), a classical verifier can always use Mahadev's protocol to force any efficient quantum server to behave as a trusted measurement device. By combining the results from complexity theory [KKR06, BL08, MF16] that a classical verifier armed with a trusted measurement device is able to verify any BQP computation, Mahadev's work opens the way to classically verify the computation performed by a single quantum server under a mild cryptographic assumption [ACGH19, CCKW19, CCY20, GV19, MV20, Vid20].

All the above works implicitly use a fact that any BQP computation can be decomposed into a set of universal quantum gates, or equivalently a sequence of basic unitary quantum operations, which allows to reduce the verification of the whole computation to the verification of some set of universal quantum gates or some basic unitary quantum operations. Furthermore, the verification techniques used in existing solutions are usually specific to some particular way of decomposing the computation, and cannot be trivially extended to other settings. Here comes the question: *Can we verify a quantum computation without knowing an efficient way to decompose it (into a particular set of gates)?* Consider, for example, a cloud provider who buys a quantum device dedicated to a special task $F$ from some hardware company, and wants to convince cloud users that it indeed holds a quantum device correctly computing the task $F$. It is likely that both the provider and the cloud users do not know an efficient way to decompose the computations performed inside the black-box device (into a known set of quantum gates that they can compute). This motivates us to study interactive proofs for the following problem.

**Problem 1.1** *The promise problem* $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ *consists of a quantum device* $\mathcal{D}$ *computing some self-adjoint unitary operation on* $(n+m)$ *qubits (i.e.,* $\mathcal{D}(\mathcal{D}(|z\rangle)) = |z\rangle$*), a classical device* $\mathcal{R}_F$ *deciding the input-output relation of some unknown function* $F : \{0,1\}^n \to \{0,1\}^m$*, and two reals* $0 < b < a \leq 1$*. Let* $p(\mathcal{D}, x) = \| \, |x, F(x)\rangle\langle x, F(x)| \, \mathcal{D}(|x\rangle \, |0^m\rangle)\|^2$ *be the probability of obtaining* $(x, F(x))$ *as a result of a standard measurement of the* $(n+m)$*-qubit state returned by* $\mathcal{D}$ *on input* $|x\rangle \, |0^m\rangle$*. The task is to distinguish between two cases for all* $x \in \{0,1\}^n$*:*

- *YES Instance:* $p(\mathcal{D}, x) \geq a$*;*
- *NO Instance:* $p(\mathcal{D}, x) \leq b$*.*

For non-triviality, we assume that $b$ is not smaller than the probability for any QPT algorithm to correctly guess $F(x)$ without knowing $x \in \{0,1\}^n$. Otherwise, there is a trivial quantum device $\mathcal{D}'$ implementing this guess algorithm, s.t., $p(\mathcal{D}', x) > b$ for all $x \in \{0,1\}^n$. Informally, the YES instance says that $\mathcal{D}$ computes $F$ with an error probability at most $1 - a$, while the NO instance says that $\mathcal{D}$ computes $F$ with an error probability at least $1 - b$. In the above definition, the classical device $\mathcal{R}_F$ is used to capture all the information known about the function $F$, which takes a pair $(x, y) \in \{0,1\}^n \times \{0,1\}^m$ as input, outputs 1 if $y = F(x)$ and 0 otherwise. The choice of $\mathcal{R}_F$ is mainly based on the following considerations: 1) it is impossible to verify a statement depending on $F$ without knowing any information about $F$; 2) $F$ may compute a task (e.g., integer factorizations) which has no efficiently classical algorithm but can be easily verified [ABOEM17]; and 3) given a device computing $F$, one can efficiently implement $\mathcal{R}_F$, but the reverse does not necessarily hold.

Our $\mathsf{QBBC}$ problem can be seen as a variant of the BQP-complete problem Q-CIRCUIT. In [ABOE10, ABOEM17, Bro18], the Q-CIRCUIT problem was shown to have quantum prover interactive proofs (QPIP), which allows a verifier with limited quantum capabilities to verify any BQP computation performed by an untrusted server, assuming that the computation can be represented by a quantum circuit $U$ consisting of a known set of universal quantum gates $U = U_T \dots U_1$ (see the discussions in Sec. 1.5). We wonder if it is possible for a user, given only access to a classical black-box device $\mathcal{R}_F$, to verify whether an untrusted server has a quantum black-box device $\mathcal{D}$ computing $F$. More specifically, for the promise problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$, does there exist an interactive proof between two efficient oracle algorithms $(\mathcal{P}^{\mathcal{D}}, \mathcal{V}^{\mathcal{R}_F})$ such that for some $c - s > \mathsf{poly}(n)$, the verifier $\mathcal{V}^{\mathcal{R}_F}$ accepts a YES instance with probability at least $c$ after interacting with an honest prover $\mathcal{P}^{\mathcal{D}}$, and rejects a NO instance with probability at least $1 - s$ after interacting with any QPT malicious prover $\widetilde{\mathcal{P}}^{\mathcal{D}, \mathcal{O}_F}$, where $\mathcal{O}_F$ is a classical oracle computing $F$? Note that instead of providing $\mathcal{R}_F$ to the malicious prover $\widetilde{\mathcal{P}}$, we directly allow $\widetilde{\mathcal{P}}$ to access $\mathcal{O}_F$. This will only increase the capability of the malicious prover $\widetilde{\mathcal{P}}$, and can capture the potential case that $\widetilde{\mathcal{P}}$ may have a device which hardwires some correct pairs $(x, F(x))$.

## 1.1 Our Results

In this paper, we first show that for any constant $15/16 < a \le 1$, the problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ has a two-round interactive proof $(\mathcal{P}^{\mathcal{D}}, \mathcal{V}^{\mathcal{R}_F})$ where the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ are efficient oracle algorithms with some (limited) quantum capabilities: the prover $\mathcal{P}$ is essentially a trivial algorithm which simply replays the quantum messages between the verifier $\mathcal{V}$ and the device $\mathcal{D}$; the verifier $\mathcal{V}$ needs to prepare and send at most two quantum messages, and make appropriate measurements to the responses from the prover. For query complexities, $\mathcal{P}^{\mathcal{D}}$ makes at most two quantum queries to $\mathcal{D}$, and $\mathcal{V}^{\mathcal{R}_F}$ only makes a single classical query to $\mathcal{R}_F$. Moreover, this proof system achieves completeness $\frac{1+a}{2}$ and soundness error at most $\frac{31}{32} + \frac{\epsilon}{2} + \mathsf{negl}(n)$ for any constant $\max(0, b - \frac{15}{16}) < \epsilon < a - \frac{15}{16}$. Our first result is summarized in Theorem 1.1.

**Theorem 1.1 (Informal)** *For the promise problem* $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ *with constant* $15/16 < a \le 1$, *there exists a two-round interactive proof* $(\mathcal{P}^{\mathcal{D}}, \mathcal{V}^{\mathcal{R}_F})$ *such that*

- $\mathcal{P}^{\mathcal{D}}$ *is a QPT oracle algorithm, making at most two quantum queries to* $\mathcal{D}$;
- $\mathcal{V}^{\mathcal{R}_F}$ *is a QPT oracle algorithm, making a single classical query to* $\mathcal{R}_F$;
- **Completeness:** *if* $(\mathcal{D}, \mathcal{R}_F, a, b)$ *is a YES instance, then the probability that* $\mathcal{V}^{\mathcal{R}_F}$ *accepts after interacting with an honest prover* $\mathcal{P}^{\mathcal{D}}$ *is at least* $\frac{1+a}{2}$;
- **Soundness:** *if* $(\mathcal{D}, \mathcal{R}_F, a, b)$ *is a NO instance, then the probability that* $\mathcal{V}^{\mathcal{R}_F}$ *accepts after interacting with any QPT algorithm* $\widetilde{\mathcal{P}}^{\mathcal{D}, \mathcal{O}_F}$ *is at most* $\frac{31}{32} + \frac{\epsilon}{2} + \mathsf{negl}(n)$ *for any constant* $\max(0, b - \frac{15}{16}) < \epsilon < a - \frac{15}{16}$, *where* $\mathsf{negl}(n)$ *denotes an unspecified negligible function in* $n$.

Moreover, we show that the problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ even has an efficient interactive proof $(\mathcal{P}^{\mathcal{D}}, \mathcal{V}^{\mathcal{R}_F})$ such that the verifier is a probabilistic polynomial time (PPT) algorithm and does not have any quantum capability (i.e., $\mathcal{V}$ is a fully classical oracle algorithm), under the computational assumption that no QPT algorithm can solve the LWE problem [Reg05]. This proof system achieves the same query complexities as the one in Theorem 1.1. Furthermore, it has completeness $\frac{1+a}{2} - \mathsf{negl}(n)$ and soundness error $\frac{1+b}{2} + \mathsf{negl}(n)$, and thus applies to any $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ with constants $b < a$. Our second result is summarized in Theorem 1.2.

**Theorem 1.2 (Informal)** *If the LWE problem is hard for all QPT algorithms, then for the promise problem* $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$*, there exists an interactive proof* $(\mathcal{P}^{\mathcal{D}}, \mathcal{V}^{\mathcal{R}_F})$ *such that:*

- $\mathcal{P}^{\mathcal{D}}$ *is a QPT oracle algorithm, making at most two quantum queries to* $\mathcal{D}$*;*
- $\mathcal{V}^{\mathcal{R}_F}$ *is a PPT oracle algorithm, making a single classical query to* $\mathcal{R}_F$*;*
- **Completeness:** *if* $(\mathcal{D}, \mathcal{R}_F, a, b)$ *is a YES instance, then the probability that* $\mathcal{V}^{\mathcal{R}_F}$ *accepts after interacting with an honest prover* $\mathcal{P}^{\mathcal{D}}$ *is at least* $\frac{1+a}{2} - \mathsf{negl}(n)$*;*
- **Soundness:** *if* $(\mathcal{D}, \mathcal{R}_F, a, b)$ *is a NO instance, then the probability that* $\mathcal{V}^{\mathcal{R}_F}$ *accepts after interacting with any QPT algorithm* $\widetilde{\mathcal{P}}^{\mathcal{D}, \mathcal{O}_F}$ *is at most* $\frac{1+b}{2} + \mathsf{negl}(n)$*.*

As the proof systems in Theorems 1.1 and 1.2 have almost constant completeness-soundness gaps (i.e., $\epsilon' - \mathsf{negl}(n)$ for some constant $\epsilon' > 0$), they can be sequentially repeated a polynomial number of times to obtain almost perfect completeness and negligible soundness error.

## 1.2 Application: Separation between ROM and QROM

In the random oracle model (ROM), all parties, including the adversary, are given classically access to a black-box random function (i.e., a random oracle, RO). Since its introduction [BR93], the ROM has been successfully used to design and analyze many well-known cryptosystems such as the OAEP encryption [BR95] and the FDH signature [BR96]. By observing that the ROM may be problematic for quantum adversaries, the authors of [BDF$^+$11] introduced the quantum ROM (QROM) where honest parties (e.g., the cryptosystems) still access the RO in a classical way, but the adversary is explicitly allowed to make quantum queries to the RO. The QROM has also been widely used to analyze the security of many post-quantum cryptosystems, including the ones submitted to NIST Post-Quantum Cryptography Standardization [NIS16].

Note that the only difference between ROM and QROM is that the adversary in the QROM can make quantum queries to the RO while that in the ROM can only make classical queries. This can be naturally seen as that the adversary in the QROM has a quantum device $\mathcal{D}$ correctly computing an idealized random function $\mathcal{O}(\cdot) : \{0,1\}^n \to \{0,1\}^m$, while that in the ROM only has a trivial quantum device $\mathcal{D}$ simply guessing the output of $\mathcal{O}(\cdot)$ at each point (with a correct probability at most $\frac{1}{2^m}$). As the honest parties are given classically access to the RO $\mathcal{O}(\cdot)$ (and thus has a natural device $\mathcal{R}_{\mathcal{O}}$ deciding the input-output relation of $\mathcal{O}(\cdot)$), the problem of distinguishing ROM and QROM is a natural promise problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_{\mathcal{O}}, 1, \frac{1}{2^m})$.

By Theorem 1.2, we immediately have that there is an efficient classical distinguisher $\mathcal{V}'$ (obtained by repeatedly running the verifier $\mathcal{V}$ in Theorem 1.2 a polynomial number of times) for ROM and QROM, such that it almost always outputs 1 after interacting with a QROM adversary performing the strategy of an honest prover $\mathcal{P}$, and 0 after interacting with any adversary in the ROM. This distinguisher $\mathcal{V}'$ can be used as a building block to construct cryptosystems that are secure in the ROM but insecure in the QROM, as it allows to embed some malicious behaviors that can only be utilized by an adversary in the QROM: given a secure cryptosystem $\mathcal{C}$, one can construct another cryptosystem $\mathcal{C}'$ which first internally runs the distinguisher $\mathcal{V}'$ to detect if the adversary runs in the QROM, and then performs normally as $\mathcal{C}$ does if $\mathcal{V}'$ outputs 0, otherwise behaves maliciously (e.g., directly outputting the secret key to the adversary) if $\mathcal{V}'$ outputs 1 (see Sec. 5). A direct corollary of Theorem 1.2 is as follows.

**Corollary 1.1 (Informal)** *There exist cryptosystems which are secure in the ROM but are insecure in the QROM.*

## 1.3 Overview of Interactive Proof with a (Limited) Quantum Verifier

In this overview, we consider a simplified case where $a = 1$ (i.e., for a YES instance, $\mathcal{D}$ always correctly computes $F$), but the same idea can be easily extended to the case $15/16 < a \leq 1$. In this case, if the verifier $\mathcal{V}$ is allowed to directly access the device $\mathcal{D}$, then it can easily distinguish whether the device $\mathcal{D}$ is a YES instance or a NO instance. Specifically, $\mathcal{V}$ can simply send an arbitrary query to the device $\mathcal{D}$, measure the obtained state from $\mathcal{D}$ and accept if the outcome is a correct pair $(x, F(x))$ via a query to the device $\mathcal{R}_F$. Clearly, if $\mathcal{D}$ is a YES instance, then $\mathcal{V}$ will always obtain a correct pair, otherwise it will obtain a correct pair with probability at most $b$. The problem here is that $\mathcal{V}$ can only access the device $\mathcal{D}$ via the prover, and a malicious prover $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ may cheat the verifier with the classical oracle $\mathcal{O}_F$ (that always correctly computes $F$).

Our starting point is that $\mathcal{V}$ can encode exponentially many classical queries into a single quantum state $|\phi\rangle = \sum_{x\in\{0,1\}^n} |x\rangle$ in superposition, while an efficient malicious prover $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ can only make a polynomial number of classical queries to $\mathcal{O}_F$ (and thus can only obtain a polynomial number of valid pairs $(x, F(x))$ with certainty). In particular, if $\mathcal{V}$ sends the quantum challenge $|\phi\rangle$ to $\widetilde{\mathcal{P}}$, it is infeasible for an efficient $\widetilde{\mathcal{P}}$ to compute a correct response $|\psi\rangle = \sum_{x\in\{0,1\}^n} |x, F(x)\rangle = \sum_{x\in\{0,1\}^n} |x, F(x)\rangle$ (note that in our simplified case, an honest prover $\mathcal{P}^{\mathcal{D}}$ with a YES instance $\mathcal{D}$ can always return a correct response). The main problem is that $\mathcal{V}^{\mathcal{R}_F}$ cannot check if a response from $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ is correct or not, as by the quantum uncertainty principle it cannot extract all the classical pairs $\{(x, F(x))\}_{x\in\{0,1\}^n}$ from the quantum state $|\psi\rangle$.

To get around the above obstacle, we let $\mathcal{V}$ send a challenge state $|\phi\rangle = \sum_{x\in X} |x\rangle$ using a random subset $X \subseteq \{0,1\}^n$, which ensures that a measurement on any correct response $|\psi\rangle = \sum_{x\in X} |x, F(x)\rangle = \sum_{x\in X} |x, F(x)\rangle$ always results in a pair $(x \in X, F(x))$. Clearly, if we could show that $\mathcal{P}$ cannot obtain sufficient information of $X$ from $|\phi\rangle = \sum_{x\in X} |x\rangle$ to make a query $x \in X$ to $\mathcal{O}_F$, then the proof is completed. However, this is not achievable as $\mathcal{P}$ can easily obtain an element $x \in X$ by simply measuring the challenge state $|\phi\rangle = \sum_{x\in X} |x\rangle$. Fortunately, we can show that there is a way to choose $X$ such that it is infeasible for $\mathcal{P}$ to obtain sufficient information of $X$ without significantly disturbing the state $|\phi\rangle = \sum_{x\in X} |x\rangle$. Technically, we will establish an *information versus disturbance* lemma (see Lemma 3.1), which gives a quantitative connection between the information that an algorithm obtains about $X$ from a state $|\phi\rangle = \sum_{x\in X} |x\rangle$ and the disturbance that it causes to the state $|\phi\rangle$. A main corollary of this lemma we needed is stated as follows. Let $H^1$ be the single-qubit Hadamard operation $H$, and let $H^0$ be the identity operation $\mathsf{Id}$. For any $s = s_1 \dots s_n \in \{0,1\}^n$, define an $n$-qubit quantum operation $H^s \overset{\text{def}}{=} H^{s_1} \otimes H^{s_2} \cdots \otimes H^{s_n}$.

**Corollary 1.2 (Informal)** *Let $x, s$ be two random variables uniformly distributed over $\{0,1\}^n$. For any quantum algorithm $(e, |\zeta\rangle) \leftarrow \mathcal{P}(H^s |x\rangle)$, the mutual information $I(x, e)$ between $x$ and $e \in \{0,1\}^{\mathsf{poly}(n)}$ is upper bounded by $2n\sqrt{1-\delta}$, where $\delta$ is the probability of obtaining $x \in \{0,1\}^n$ as a result of a standard measurement of the $n$-qubit state $H^s |\zeta\rangle$.*

Let $X_{x,s} \subseteq \{0,1\}^n$ be the set of $\tilde{x} \in \{0,1\}^n$ that agrees with $x$ at the positions indexed by zeros in $s \in \{0,1\}^n$. Then, we have $H^s |x\rangle = \sum_{\tilde{x}\in X_{x,s}} |\tilde{x}\rangle$. Informally, the above corollary says that given the state $|\phi\rangle = \sum_{\tilde{x}\in X_{x,s}} |\tilde{x}\rangle$, it is impossible for any quantum algorithm $\mathcal{P}$ to obtain sufficient information of $x$ while at the same time outputting a state to recover $x$ in a particular way. At a high level, the verifier $\mathcal{V}$ for Theorem 1.1 will prepare and send a quantum state $H^s |x\rangle$ to the prover by randomly choosing two classical strings $x, s \in \{0,1\}^n$ as in Corollary 1.2, and accept a quantum state returned by the prover as a valid proof only if 1) a measurement on the received state results in a pair $(\tilde{x} \in X_{x,s}, F(\tilde{x}))$; and 2) one can always "uncompute" the received state to obtain $x$ (which needs to send the received quantum state back to the prover to undo the computation) in a particular way. The proof system is briefly described as follows:

1. The verifier $\mathcal{V}$ randomly chooses $x, s \in \{0,1\}^n$, computes and sends $|\phi\rangle = H^s |x\rangle = \sum_{\tilde{x}\in X_{x,s}} |\tilde{x}\rangle$ to the prover;

2. The prover $\mathcal{P}$ sends a query $|\phi\rangle |0^m\rangle$ to the device $\mathcal{D}$, and returns the received $(n+m)$-qubit state $|\psi\rangle$ from $\mathcal{D}$ to the verifier;

3. $\mathcal{V}$ randomly chooses a bit $\delta \in \{0,1\}$,

   3.1 In case $\delta = 0$, $\mathcal{V}$ checks if a measurement on $|\psi\rangle$ results in a valid pair $(\tilde{x} \in X_{x,s}, F(\tilde{x}))$ by using a query to $\mathcal{R}_F$. If yes, $\mathcal{V}$ accepts, otherwise rejects;

   3.2 In case $\delta = 1$, $\mathcal{V}$ sends back $|\psi\rangle$ to the prover;

4. $\mathcal{P}$ sends a query $|\psi\rangle$ to the device $\mathcal{D}$, and returns a state $|\phi'\rangle$ (which is expected to be $|\phi\rangle$) containing the first $n$ qubits of the received state from $\mathcal{D}$ to the verifier;

5. $\mathcal{V}$ applies $H^s$ on $|\phi'\rangle$, and checks if a measurement on the resulting state gives $x \in \{0,1\}^n$. If yes, $\mathcal{V}$ accepts, otherwise rejects.

Clearly, for a YES instance $\mathcal{D}$, an honest prover $\mathcal{P}^{\mathcal{D}}$ can always pass the two checks made by the verifier $\mathcal{V}$ in step 3.1 and step 5. As $\delta$ is randomly chosen by $\mathcal{V}$ after receiving $|\psi\rangle$, a successfully malicious prover $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ must output a state $|\psi\rangle$ simultaneously passing the two checks. In particular, $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ has to obtain sufficient information about $x$ to determine a correct pair $(\tilde{x} \in X_{x,s}, F(\tilde{x}))$ (via a query to $\mathcal{O}_F$) to pass the check in step 3.1 while at the same time outputting a state $|\phi'\rangle$ to recover $x$ in a particular way to pass the check in step 5 (note that the strategy of $\mathcal{V}$ essentially requires that $\widetilde{\mathcal{P}}$ can directly output $|\phi'\rangle$), which is impossible by Corollary 1.2. The analysis for $15/16 < a \leq 1$ is almost similar, please refer to Sec. 3 for details.

### 1.4 Overview of Interactive Proof with a Fully Classical Verifier

The idea is essentially the same as above: the verifier $\mathcal{V}$ prepares a quantum state, which contains a verifiable hidden set $X$, as a challenge to the prover such that it is infeasible for an efficiently malicious prover $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ to output a valid pair $(\tilde{x} \in X, F(\tilde{x}))$ with probability more than $b$. The problem is that the verifier now has no quantum capabilities, and cannot prepare quantum states to $\widetilde{\mathcal{P}}$ or measure the quantum states returned by $\widetilde{\mathcal{P}}$ to perform any check.

Our starting point is to delegate "the computation of the quantum verifier $\mathcal{V}$" to the prover. At first glance, one might think it is trivial, because Mahadev [Mah18a] showed that classical verification of any BQP computation can be achieved under the LWE assumption. However, this intuition does not work because the result in [Mah18a] uses an assumption that the computation can be implemented by a quantum circuit consisting of a polynomial number of universal quantum gates, while in our case 1) both the prover $\mathcal{P}^{\mathcal{D}}$ and the verifier $\mathcal{V}^{\mathcal{R}_F}$ are oracle algorithms, which cannot be represented by quantum circuits with simple universal quantum gates; and 2) the verifier $\mathcal{V}^{\mathcal{R}_F}$ needs to have some secret information (i.e., the hidden set $X$), which cannot be encoded into a public quantum circuit given to the (malicious) prover.

Instead, we focus on the two main quantum capabilities (i.e., quantum states generation and measurement) that we want the verifier to have, and directly present two protocols to do the delegations. Both protocols are based on a primitive called extended noisy trapdoor claw-free functions (eNTCF) in [BCM+18, Mah18a], which can be securely instantiated under the LWE assumption. For simplicity, we use a perfect primitive, namely, extended trapdoor claw-free functions (eTCF), to elaborate the high-level ideas without brothering about the noisy feature of eNTCF, but only keep in mind that changing eTCF back to eNTCF will only introduce a negligible part to both the correctness and the security of the resulting protocols.

We begin by briefly describing two families of functions $\mathcal{F}$ and $\mathcal{G}$ from some finite set $\mathcal{X}$ to $\mathcal{Y}$. In this overview, we assume $\mathcal{X} = \{0,1\}^w$ for convenience. Informally, we say that $\mathcal{F} = \{f_{k,b} : \{0,1\}^w \to \mathcal{Y}\}_{k \in \mathcal{K}_{\mathcal{F}}, b \in \{0,1\}}$ is a trapdoor claw-free functions (TCF) family if

1. One can efficiently sample a key $k \in \mathcal{K}_F$ and a trapdoor $t_k$ corresponding to a pair of functions $f_{k,0}, f_{k,1} \in \mathcal{F}$;

2. Both $f_{k,0}$ and $f_{k,1}$ are injective and have *equal* images (i.e., for any image $y$ of $f_{k,0}$ or $f_{k,1}$, there only exists a unique claw $(x_0, x_1) \in \{0,1\}^w \times \{0,1\}^w$, s.t., $y = f_{k,0}(x_0) = f_{k,1}(x_1)$);
3. Given only $k \in \mathcal{K}_F$, it is quantum computationally hard to find a claw $(x_0, x_1)$ or a tuple $(x_b, d, d \cdot (x_0 \oplus x_1))$ for some $d \subseteq \{0,1\}^n$, s.t., $f_{k,0}(x_0) = f_{k,1}(x_1)$;
4. Given the trapdoor $t_k$, one can efficiently recover a claw $(x_0, x_1)$ from any image $y$ of $f_{k,0}$ or $f_{k,1}$, s.t., $y = f_{k,0}(x_0) = f_{k,1}(x_1)$.

We say that $\mathcal{G} = \{g_{k,b} : \{0,1\}^w \to \mathcal{Y}\}_{k \in \mathcal{K}_\mathcal{G}, b \in \{0,1\}}$ is a trapdoor injective function family if

1. One can efficiently sample a key $k \in \mathcal{K}_G$ and a trapdoor $t_k$ corresponding to a pair of functions $g_{k,0}, g_{k,1} \in \mathcal{G}$;
2. Both $g_{k,0}$ and $g_{k,1}$ are injective and have *disjoint* images (i.e., for any image $y$ of $g_{k,0}$ or $g_{k,1}$, there only exists a unique pair $(b, x_b) \in \{0,1\} \times \{0,1\}^w$, s.t., $y = g_{k,b}(x_b)$);
3. Given the trapdoor $t_k$, one can efficiently recover the unique pair $(b, x_b)$ from any image $y$ of $g_{k,0}$ or $g_{k,1}$, s.t., $y = g_{k,b}(x_b)$.

We say that $\mathcal{F}$ is an extended TCF (eTCF) if there exists a trapdoor injective function family $\mathcal{G}$ such that the keys of $\mathcal{F}$ and $\mathcal{G}$ are quantum computationally indistinguishable, and both $\mathcal{F}$ and $\mathcal{G}$ have the same efficient evaluation algorithm.

**Generation of Quantum State with Verifiable Hidden Set.** Our first protocol allows a classical verifier $\mathcal{V}$ and a quantum prover $\mathcal{P}$ to cooperatively generate an $n$-qubit state $|\phi\rangle = \sum_{x \in X} |x\rangle$ with a verifiable hidden set $X$. The protocol has a generation phase and a verification phase (see Sec. 4.1). At the end of the generation phase, the prover $\mathcal{P}$ holds a state $|\phi\rangle$, while the verifier $\mathcal{V}$ obtains the set $X$ and some trapdoor information *td*. In the verification phase, $\mathcal{V}$ can use *td* to ensure that the prover indeed produces the state $|\phi\rangle = \sum_{x \in X} |x\rangle$ without knowing $X$ (in particular, the prover cannot output an element of $X$ before the verification phase).

Technically, this protocol is built upon the protocol in [BCM+18], which allows a classical verifier to delegate the generation of randomness to an untrusted quantum prover (that is polynomial-time bounded). The high-level idea works as follows. The verifier first samples a key $k \in \mathcal{K}_\mathcal{F}$ and a trapdoor $t_k$ corresponding to a pair of trapdoor claw-free functions $f_{k,0}, f_{k,1} \in \mathcal{F}$, and sends the key $k$ to the prover. The prover is asked to first prepare a quantum state:

$$\frac{1}{\sqrt{2^{(w+1)}}} \sum_{b \in \{0,1\}, x \in \{0,1\}^w} |b, x\rangle |0\rangle_\mathcal{Y},$$

Then, the prover evaluates the function $f_{k,b}$ on the state using $k \in \mathcal{K}_\mathcal{F}$:

$$\frac{1}{\sqrt{2^{(w+1)}}} \sum_{b \in \{0,1\}, x \in \{0,1\}^w, y = f_{k,b}(x)} |b, x\rangle |y\rangle_\mathcal{Y},$$

measures the $\mathcal{Y}$-register and sends the outcome $\hat{y}$ to the verifier. After this, the prover will obtain a state

$$\frac{1}{\sqrt{2}} \sum_{b \in \{0,1\}, x_{b,\hat{y}} \in \{0,1\}^w} |b, x_{b,\hat{y}}\rangle |\hat{y}\rangle_\mathcal{Y},$$

where $(x_{0,\hat{y}}, x_{1,\hat{y}})$ satisfies $\hat{y} = f_{k,0}(x_{0,\hat{y}}) = f_{k,1}(x_{1,\hat{y}})$ by the property of $\mathcal{F}$. Clearly, if the prover performs honestly, measuring the first qubit of the above state will give a random bit $b \in \{0,1\}$.

In order to certify the behaviors of the prover, the verifier repeatedly executes the above process a polynomial number $n$ of times by using a fresh key $k_i$ at each time, and randomly determine each repetition as a test one or a normal one with a certain probability [BCM+18].[1] The two types of repetitions only differ after the verifier receives $\hat{y}$ from the quantum prover. Specifically, in a normal repetition, the prover is asked to measure the above state and return the outcome $(b, x_{b,\hat{y}})$ to the verifier as desired. While in a test one, the prover is asked to output either $(b, x_{b,\hat{y}})$ or $(d, d \cdot (x_{0,\hat{y}} \oplus x_{1,\hat{y}}))$ with equal probability, where the latter is computed by first applying a Hadamard transform to the first $w+1$ qubits of the above state to obtain

$$\frac{1}{\sqrt{2^{w+2}}} \sum_{b,u\in\{0,1\},d\in\{0,1\}^w} (-1)^{d\cdot x_{b,\hat{y}}\oplus ub} |u, d\rangle |\hat{y}\rangle_{\mathcal{Y}} = \frac{1}{\sqrt{2^w}} \sum_{d\in\{0,1\}^w} (-1)^{d\cdot x_{0,\hat{y}}} |d \cdot (x_{0,\hat{y}} \oplus x_{1,\hat{y}}), d\rangle |\hat{y}\rangle_{\mathcal{Y}}$$

and then measuring the resulting state. Let $S \subseteq [n] = \{1, \ldots, n\}$ specify the positions of the normal repetitions (in the whole $n$ repetitions). For each $i \in S$, let $b_i \in \{0, 1\}$ be the first bit of "$(b, x_{b,\hat{y}})$" obtained in the normal repetition $i$. The authors of [BCM+18] showed that if the prover correctly answers the questions in most test repetitions, then the bits $\{b_i\}_{i\in S}$ have sufficient entropy even conditioned on all other transcripts of the interactions.

We modify the above protocol by replacing $(k_i \in \mathcal{K}_{\mathcal{F}}, t_{k_i})$ used in a normal repetition $i \in S$ with $(k'_i \in \mathcal{K}_{\mathcal{G}}, t_{k'_i})$ corresponding to a pair of trapdoor injective functions $g_{k'_i,0}, g_{k'_i,1} \in \mathcal{G}$. As $f_{k_i,b}$ and $g_{k'_i,b}$ have the same evaluation algorithm, the prover can compute a state:

$$\frac{1}{\sqrt{2^{(w+1)}}} \sum_{b\in\{0,1\},x\in\{0,1\}^w,y=g_{k'_i,b}(x)} |b, x\rangle |y\rangle_{\mathcal{Y}} .$$

The prover then measures the $\mathcal{Y}$-register and sends the outcome $\hat{y}_i$ to the verifier. By the injective property of $\mathcal{G}$, the state held by the prover will collapse to $|b_i, x_{b_i,\hat{y}_i}\rangle |\hat{y}_i\rangle$, where $\hat{y}_i = g_{k'_i,b_i}(x_{b_i,\hat{y}_i})$. Thus, given the trapdoor $t_{k'_i}$ corresponding to $k'_i \in \mathcal{K}_{\mathcal{G}}$, the verifier can invert $b_i \in \{0, 1\}$ from $\hat{y}_i$. For a test repetition $i \notin S$, the verifier samples $(k_i \in \mathcal{K}_{\mathcal{F}}, t_{k_i})$ normally so that it can certify the behaviors of the prover the same way as that in [BCM+18]. By the assumption that $k_i$ and $k'_i$ are quantum computationally indistinguishable, the prover cannot distinguish this modified protocol from the original one, and thus it cannot determine $b_i \in \{0, 1\}$ at the time of generating $\hat{y}_i$ by the security of the original protocol. Because the prover does not know the repetition type at the time of sending $\hat{y}_i$ to the verifier (i.e., his behavior of producing $\hat{y}_i$ should be independent from the repetition type) and $k_i$ is freshly sampled for each repetition, we can reorganize this modified protocol into a generation phase and a verification phase.

In the generation phase, the verifier first randomly chooses a set $S \subset [n]$ by independently picking each element $i \in S$ with a certain probability. For each $j \in [n]$, it samples a key $k_j \in \mathcal{K}_{\mathcal{G}}$ and a trapdoor $t_{k_j}$ corresponding to a pair of trapdoor injective functions $g_{k_j,0}, g_{k_j,1} \in \mathcal{G}$ if $j \in S$, and samples a key $k_j \in \mathcal{K}_{\mathcal{F}}$ and a trapdoor $t_{k_j}$ corresponding to a pair of trapdoor claw-free functions $f_{k_j,0}, f_{k_j,1} \in \mathcal{F}$ otherwise. Then, it sends $\{k_j\}_{j\in[n]}$ to the prover, and asks the prover to return a set $\{\hat{y}_j\}_{j\in[n]}$ by computing each $\hat{y}_j$ using $k_j$ as above. After this, the prover will obtain an $n$-qubit (non-normalized) state $|\phi\rangle$ (consisting of the first qubit of the state obtained after producing $\hat{y}_j$ for all $j \in [n]$); the verifier can compute $b_j \in \{0, 1\}$ by using the trapdoor $t_{k_j}$ and $\hat{y}_j$ for all $j \in S$. In the verification phase, for each $j \in [n]$, the verifier performs the same as the original protocol in [BCM+18] to certify the behaviors of the prover.

Let $X \subset \{0, 1\}^n$ be the set of bit strings $x \in \{0, 1\}^n$ whose $j$-th bit agrees with $b_j$ for all $j \in S$. By using almost the same security analysis as that in [BCM+18] (as in the view of the prover, our protocol is quantum computationally indistinguishable from a variant of [BCM+18]

[1] Strictly speaking, the randomness expansion protocol [BCM+18] will reuse some key $k_i$ to save the number of random bits consumed by the verifier, which requires the verifier to determine the type of each repetition in a sequential way. We use a fresh key $k_i$ at each repetition because we do not have the concern of randomness expansion (i.e., the length of the generated random bits should be longer than that of the input ones) and we want to compute the $n$ repetitions in a partially parallel manner as we will see later. Note that the security analysis in [BCM+18] naturally extends to the case of using a fresh key $k_i$ at each repetition.

using a fresh key $k_j$ at each repetition), the probability that the prover outputs $x \in X$ before the verification phase is negligible. In summary, we have the following theorem.

**Theorem 1.3 (informal)** *Under the LWE assumption, there is a protocol $\Pi_G$ consisting of a generation phase and a verification phase between a classical verifier $\mathcal{V}$ and a quantum prover $\mathcal{P}$. After the generation phase, $\mathcal{P}$ will obtain an $n$-qubit state $|\phi\rangle$; $\mathcal{V}$ will obtain a set $X$ and some trapdoor information td. After the verification phase, $\mathcal{V}$ will output a bit indicating whether accepts or rejects.*

- **Correctness:** *In an honest execution, the state $|\phi\rangle$ held by the prover is within negligible trace distance from the state $\sum_{x \in X} |x\rangle$, and the verifier will almost always accept.*
- **Security:** *Conditioned on the event that $\mathcal{V}$ accepts in the verification phase, the probability that any QPT prover outputs an element $x \in X$ before the verification phase is negligible.*

**Oblivious Measurement on Quantum State.** Our second protocol allows a classical verifier $\mathcal{V}$ to obliviously measure a state $|\psi\rangle$ held by the quantum prover $\mathcal{P}$ (see Sec. 4.2). At the end of the protocol, $\mathcal{P}$ will obtain a state $|\psi'\rangle$; depending on a choice bit $\delta \in \{0,1\}$ the verifier $\mathcal{V}$ will obtain either a measurement outcome of $|\psi\rangle$ or some information $aux \in \{0,1\}^*$ which can be used to recover $|\psi\rangle$ from $|\psi'\rangle$. The term "oblivious" comes from the feature that the prover does not know which case the verifier chooses (i.e., $\delta = 0$ or $\delta = 1$).

For simplicity, it suffices to describe the idea of obliviously measuring a single qubit. Assume that the prover $\mathcal{P}$ holds a state $|\psi\rangle = \sum_{b \in \{0,1\}} \alpha_b |b\rangle$, and the verifier $\mathcal{V}$ holds a bit $\delta \in \{0,1\}$. In order to obliviously measure the state $|\psi\rangle$, $\mathcal{V}$ first samples a key $k \in \mathcal{K}_{\mathcal{G}}$ and a trapdoor $t_k$ corresponding to a pair of trapdoor injective functions $g_{k,0}, g_{k,1} \in \mathcal{G}$ if $\delta = 0$, and samples a key $k \in \mathcal{K}_{\mathcal{F}}$ and a trapdoor $t_k$ corresponding to a pair of trapdoor claw-free functions $f_{k,0}, f_{k,1} \in \mathcal{F}$ otherwise. Then, it sends $k$ to the prover $\mathcal{P}$, and asks $\mathcal{P}$ to evaluate $f_{k,b}$ or $g_{k,b}$ (note that this can be done without telling $\mathcal{P}$ which one to evaluate) with inputs the first two registers of the following state

$$|\psi\rangle = \frac{1}{\sqrt{2^w}} \sum_{b \in \{0,1\}, x \in \{0,1\}^w} \alpha_b |b\rangle |x\rangle |0\rangle_{\mathcal{Y}} .$$

This will result in a state

$$\frac{1}{\sqrt{2^w}} \sum_{b \in \{0,1\}, x \in \{0,1\}^w, y = \tilde{g}_{k,b}(x)} \alpha_b |b\rangle |x\rangle |y\rangle_{\mathcal{Y}} ,$$

where $\tilde{g}_{k,b} = g_{k,b}$ if $\delta = 0$, otherwise $\tilde{g}_{k,b} = f_{k,b}$. Then, $\mathcal{P}$ measures the $\mathcal{Y}$-register and sends the outcome $\hat{y}$ to the verifier. If $\delta = 0$, by the injective property of $\mathcal{G}$, this will result in a state $\left|\hat{b}\right\rangle \left|x_{\hat{b},\hat{y}}\right\rangle |\hat{y}\rangle_{\mathcal{Y}}$ with probability $(\alpha_{\hat{b}})^2$ where $\hat{y} = g_{k,\hat{b}}(x_{\hat{b},\hat{y}})$. Thus, the verifier can use $t_k$ to invert $\hat{y}$ to obtain $\hat{b} \in \{0,1\}$, just as directly measuring the state $|\phi\rangle = \sum_{b \in \{0,1\}} \alpha_b |b\rangle$. Else if $\delta = 1$, by the property of $\mathcal{F}$ the prover will obtain a state

$$\sum_{b \in \{0,1\}, x_{b,\hat{y}} \in \{0,1\}^w} \alpha_b |b\rangle |x_{b,\hat{y}}\rangle |\hat{y}\rangle_{\mathcal{Y}} ,$$

where $\hat{y} = f_{k,0}(x_{0,\hat{y}}) = f_{k,1}(x_{1,\hat{y}})$. Clearly, if the verifier $\mathcal{V}$ gives the trapdoor $t_k$ to the prover $\mathcal{P}$, then $\mathcal{P}$ can uncompute the register containing $x_{b,\hat{y}}$ by recovering $x_{b,\hat{y}}$ using $t_k$ and the registers containing $b$ and $\hat{y}$ as inputs. Thus, $\mathcal{V}$ holds the information $aux = t_k$ which can be used by $\mathcal{P}$ to efficiently compute a state

$$\sum_{b \in \{0,1\}} \alpha_b |b\rangle |0\rangle |\hat{y}\rangle_{\mathcal{Y}} .$$

Tracing out the last two registers, this yields the input state $|\phi\rangle = \sum_{b \in \{0,1\}} \alpha_b |b\rangle$. By the assumption that the keys of $\mathcal{F}$ and $\mathcal{G}$ are quantum computationally indistinguishable, we have no (malicious) prover can determine the value of $\delta$ with non-negligible advantage.

**Theorem 1.4 (informal)** *Under the LWE assumption, there is a protocol $\Pi_M$ between a classical verifier $\mathcal{V}$ with input a uniformly random bit $\delta \in \{0,1\}$ and a quantum prover $\mathcal{P}$ with input a quantum state $|\psi\rangle$.*

- **Correctness:** *In an honest execution, the prover $\mathcal{P}$ will obtain a state $|\psi'\rangle$; the verifier will obtain either a measurement outcome on $|\psi\rangle$ if $\delta = 0$, or some information $aux \in \{0,1\}^*$ which can be used by $\mathcal{P}$ to recover a state within negligible trace distance from $|\psi\rangle$;*
- **Security:** *The probability that any QPT prover outputs $\delta \in \{0,1\}$ is at most $\frac{1}{2} + \mathsf{negl}(n)$.*

**The Interactive Proof System.** By gluing protocols $\Pi_G$ and $\Pi_M$ together, we obtain an interactive proof with a classical verifier, which has the same structure as the quantum one:

1. The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ execute the generation phase of $\Pi_G$. After this, $\mathcal{P}$ will obtain an $n$-qubit state $|\phi\rangle$, and $\mathcal{V}$ will obtain classical information $(X, td)$;
2. $\mathcal{P}$ sends a query $|\phi\rangle |0^m\rangle$ to the device $\mathcal{D}$, and obtain a $(n + m)$-qubit state $|\psi\rangle$;
3. $\mathcal{V}$ randomly chooses a bit $\delta \in \{0,1\}$, and uses $\delta$ to execute the oblivious measurement protocol $\Pi_M$ with the prover $\mathcal{P}$ on input state $|\psi\rangle$:
   - In case $\delta = 0$, the verifier $\mathcal{V}$ accepts if it obtains a valid pair $(\hat{x} \in X, F(\hat{x}))$ at the end of running $\Pi_M$, and rejects otherwise; (note that this requires a query to $\mathcal{R}_F$)
   - In case $\delta = 1$, at the end of running $\Pi_M$, the prover obtains a state $|\psi'\rangle$; the verifier $\mathcal{V}$ obtains some information $aux \in \{0,1\}^*$. $\mathcal{V}$ sends $aux$ to the prover;
4. $\mathcal{P}$ recovers a state $|\psi''\rangle$ from $|\psi'\rangle$ and $aux$ (using the algorithm in Theorem 1.4); sends a query $|\psi''\rangle$ to $\mathcal{D}$ to obtain an $(n + m)$-qubit state $|\phi'\rangle$ (which is expected to be $|\phi\rangle |0^m\rangle$);
5. $\mathcal{P}$ and $\mathcal{V}$ execute the verification phase of $\Pi_G$ on input the first $n$ qubits of $|\phi'\rangle$. $\mathcal{V}$ accepts if and only if it accepts in the verification phase of $\Pi_G$.

The analysis for the simplified case $a = 1$ is the same as for the quantum one. Specifically, by the correctness of the underlying protocols, an honest prover $\mathcal{P}^{\mathcal{D}}$ can almost always pass the checks in step 3.1 and step 5. By the security of $\Pi_M$, $\delta$ is hidden from the prover during the execution of $\Pi_M$, this means that a malicious prover $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ has to determine $\hat{x} \in X$ to pass the check in step 3.1 while at the same time convincing the verifier in step 5 that it cannot output $\hat{x} \in X$ before the verification phase of $\Pi_G$, which is infeasible by the security of $\Pi_G$. This analysis can be easily extended to the general case $a \leq 1$, please refer to Sec. 4 for details.

## 1.5 Related work and Discussions

**The Q-CIRCUIT Problem.** The promise problem Q-CIRCUIT consists of two reals $a - b > 1/\mathsf{poly}(n)$, and a quantum circuit made of a sequence of gates $U = U_T \dots U_1$, acting on $n$ input bits. The task is to distinguish between the two cases for all $x \in \{0,1\}^n$:

- YES Instance: $\|(|1\rangle\langle 1| \otimes \mathsf{Id}_{n-1})U(|x\rangle)\|^2 \geq a$;
- NO Instance: $\|(|1\rangle\langle 1| \otimes \mathsf{Id}_{n-1})U(|x\rangle)\|^2 \leq b$.

The Q-CIRCUIT problem is a BQP-complete problem, and has been used to prove that any BQP computation has quantum prover interactive proofs (QPIP) [ABOE10, ABOEM17, Bro18], where the verifier has limited quantum capabilities (and cannot evaluate the circuit $U$); the prover has full quantum abilities and wants to convince the verifier that $U$ is a YES instance. One can think that our $\mathsf{QBBC}$ problem as a special case of the Q-CIRCUIT problem where the quantum circuit $U'$ consists of a (black-box) circuit $\mathcal{D}$ followed by a circuit $U_{\mathcal{R}_F}$ computing

$\mathcal{R}_F$. Since either the prover or the verifier in QPIP for the Q-CIRCUIT problem needs to know how to decompose the quantum circuit $U'$ into a set of universal gates $\{U_i\}_{i=1,\ldots,T}$, one cannot adapt existing results for the Q-CIRCUIT problem to our $\mathsf{QBBC}$ problem where the prover is given a quantum black-box device $\mathcal{D}$ and the verifier is given a classical black-box device $\mathcal{R}_F$.

**Blind Quantum Computation.** In blind quantum computation [BFK09, Chi05, GMMR13, MPDF13, FK17, Fit17], a client can delegate any efficient quantum computation to quantum servers while at the same time keeping the inputs and the details of the computation (e.g., the quantum circuits) hidden from the servers. Blind quantum computation plays an important role in the development of verifiable quantum computation [GKK19], because it generally allows the client to execute some test quantum computation at the server side to check if the servers perform honestly (and the blind property ensures that the servers cannot distinguish whether a computation is a real one or a test one). As our proof systems for the $\mathsf{QBBC}$ problem essentially rely on the fact that the prover cannot obtain the information of a random set in a quantum state, one might think if it is possible to extend existing results for blind quantum computation to our case. Unfortunately, this does not work as all the existing works for blind quantum computation, to the best of our knowledge, are based on the fact that the client can decompose the computation into a set of universal gates or a sequence of simple unitary quantum operations.

**Quantum Fully Homomorphic Encryption (FHE).** Quantum FHE is an extension of classical FHE [Gen09], which allows to apply arbitrary efficient quantum computation to (classical or quantum) encrypted data [BJ15, DSS16, Mah18b, Bra18], and can be used to delegate the computation to untrusted servers while still keeping the data private. Due to the same reason as above, the attempt to first encrypt a quantum query $|x\rangle$ using QFHE, and then ask the prover to homomorphically compute $\mathcal{D}(|x\rangle\,|0^m\rangle)$ does not work, as existing QFHEs can only support quantum computation consisting of some known set of universal quantum gates.

**Remote State Preparation.** Several works [CCKW18, CCKW19, GV19] have shown how to classically delegate the preparation of quantum states to a remote quantum server. Unlike our protocol for generating quantum states with verifiable hidden sets (i.e., multiple-qubit states with relatively low quality), these protocols focus on generating single-qubit states with high quality, which are more complex than ours and cannot be directly used for our goal. However, all the protocols share some similarities due to the use of same techniques in [BCM$^+$18, Mah18a].

**Separations between ROM and QROM.** The first separation between ROM and QROM was given in [BDF$^+$11], which presented an identification protocol that is secure in the ROM but is insecure in the QROM. The protocol is directly built upon the gap in finding a collision of an $m$-bit output hash function between using the birthday attack with $O(2^{m/2})$ classical queries and using the Grover algorithm with $O(2^{m/3})$ quantum queries [Gro96, BHT98]. Since the query gap is polynomial (as $2^{m/2}$ can be naturally written as a polynomial of $2^{m/3}$), the argument in [BDF$^+$11] requires non-standard timing assumptions (e.g., "unit time" and "zero time" assumptions) to ensure that the running time of the protocol is longer than $O(2^{m/3})$ "unit time" for a QROM adversary to run the Grover algorithm [Gro96], but is shorter than $O(2^{m/2})$ "unit time" for a ROM adversary to carry out the birthday attack. This leaves a nine-year open question of finding a separation between ROM and QROM under standard assumptions.

In a concurrent work [YZ20b] (appearing after our initial result was posted online [ZYF$^+$19]), Yamakawa and Zhandry presented a separation between ROM and QROM (also see an update [YZ20a]). Their result is also based on the LWE assumption, but the underlying techniques are completely different from ours. In particular, their result seems to be restricted to the case of RO and cannot be adapted to the general $\mathsf{QBBC}$ problem, because their analysis relies on the fact that a security reduction can extract the RO queries of the adversary in the ROM.

**Outline.** In Section 2, we give some preliminaries, including the formal definitions of extended noisy trapdoor claw-free functions. In Section 3, we first prove an information versus disturbance lemma, which might be of independent of interest. Then, an interactive proof for the $\mathsf{QBBC}$ problem, where the verifier is able to perform some quantum operations, is presented. In Section 4, we first give concrete descriptions of the protocols for quantum states generation and oblivious measurement, which are followed by an interactive proof for the $\mathsf{QBBC}$ problem with a fully classical verifier. In Section 5, we show a concrete application of our result in separating the random oracle model (ROM) from the quantum ROM (QROM).

**Acknowledgments.** We thank Thomas Vidick for his patient and helpful explanations on the details of the protocol in [BCM+18]. Jiang Zhang and Kang Yang are supported by the National Natural Science Foundation of China (Grant Nos. 62022018, 61932019), the National Key Research and Development Program of China (Grant Nos. 2017YFB0802005, 2018YFB0804105). Yu Yu is supported by the National Natural Science Foundation of China (Grant No. 61872236). Shuqin Fan and Zhenfeng Zhang are supported by the National Key Research and Development Program of China (Grant No. 2017YFB0802005).

# 2 Preliminaries

## 2.1 Notation

Let $\mathbb{C}$ be the set of complex numbers. Denote log as the logarithm with base 2. A function $f(n)$ is negligible in $n$ if for every positive constant $c$, we have $f(n) < n^{-c}$ for sufficiently large $n$. By $\mathsf{negl}(n)$ we denote an unspecified negligible function. Denote $\mathsf{poly}(n)$ as an unspecified polynomial function in $n$. The notation $\overset{\$}{\leftarrow}$ denotes randomly choosing elements from a distribution (or the uniform distribution over a finite set). Denote $\emptyset$ as an empty set. For a finite set $\mathcal{X}$, let $D_{\mathcal{X}} = \{f : \mathcal{X} \to [0,1] | \sum_{x\in\mathcal{X}} f(x) = 1\}$ be the set of all densities on $\mathcal{X}$. For any $f \in D_{\mathcal{X}}$, denote $\mathsf{SUPP}(f)$ as the support of $f$: $\mathsf{SUPP}(f) = \{x \in \mathcal{X} | f(x) > 0\}$. For two densities $f_1, f_2 \in D_{\mathcal{X}}$, the Hellinger distance between $f_1$ and $f_2$ is

$$H^2(f_1, f_2) = 1 - \sum_{x\in\mathcal{X}} \sqrt{f_1(x) f_2(x)}.$$

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 vectors $|w\rangle = (w_0, \ldots, w_{N-1})^T, |v\rangle = (v_0, \ldots, v_{N-1})^T \in \mathbb{C}^N$, the inner product between $|w\rangle$ and $|v\rangle$ is defined as $\langle w|v\rangle = \sum_{i=0}^{N-1} w_i^* v_i \in \mathbb{C}$, where $w_i^*$ denotes the conjugate of $w_i$. Denote $|v\rangle\langle w| \in \mathbb{C}^{N\times N}$ as the outer product of $|v\rangle, |w\rangle \in \mathbb{C}^N$. The trace of a square matrix $\rho$, denoted $\mathrm{tr}(\rho)$, is defined to be the sum of elements on the main diagonal of $\rho$. The trace norm of a matrix $\rho$, denoted $\|\rho\|_{tr} = \frac{1}{2}\|\rho\|_1 = \frac{1}{2}\mathrm{tr}(\sqrt{\rho\rho^*})$, is the sum of the singular values of $\rho$.

A quantum system $\mathcal{Q}$ with $N$ configurations $\{0, \ldots, N-1\}$ 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 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 $\{0, \ldots, N-1\}$. The density matrix $\rho$ of a pure state $|\phi\rangle$ is given by $\rho = |\phi\rangle\langle\phi|$. The trace distance between two density matrices $\rho, \sigma$ is defined as $\|\rho - \sigma\|_{tr} = \frac{1}{2}\mathrm{tr}\left(\sqrt{(\rho-\sigma)^2}\right)$. The following lemma relates the Hellinger distance and the trace distance.

**Lemma 2.1** *Let $\mathcal{X}$ be a finite set and $f_1, f_2 \in D_{\mathcal{X}}$. Let*

$$|\psi_1\rangle = \sum_{x\in\mathcal{X}} \sqrt{f_1(x)}\,|x\rangle \ \text{ and } \ |\psi_2\rangle = \sum_{x\in\mathcal{X}} \sqrt{f_2(x)}\,|x\rangle\,.$$

*Then,*

$$\|\,|\psi_1\rangle\langle\psi_1| - |\psi_2\rangle\langle\psi_2|\,\|_{tr} = \sqrt{1 - (1 - H^2(f_1, f_2))^2}.$$

## 2.2 Information Theory

We recall some definitions related to the Shannon entropy of random variables. Formally, let $X, Y$ be two random variables with support $\mathcal{X}, \mathcal{Y}$, respectively. The entropy of $X$ is defined as

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

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

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

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

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

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 $Y$.

## 2.3 Interactive Proofs for the $\mathsf{QBBC}$ Problem

In this subsection, we first restate the promise problem related to quantum black-box computations (QBBC) in the following.

**Problem 1.1** *The promise problem* $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ *consists of a quantum device* $\mathcal{D}$ *computing some self-adjoint unitary operation on* $(n+m)$ *qubits (i.e.,* $\mathcal{D}(\mathcal{D}(|z\rangle)) = |z\rangle$*), a classical device* $\mathcal{R}_F$ *deciding the input-output relation of some unknown function* $F : \{0,1\}^n \to \{0,1\}^m$*, and two reals* $0 < b < a \leq 1$*. Let* $p(\mathcal{D}, x) = \| \, |x, F(x)\rangle\langle x, F(x)| \, \mathcal{D}(|x\rangle \, |0^m\rangle) \|^2$ *be the probability of obtaining* $(x, F(x))$ *as a result of a standard measurement of the* $(n+m)$*-qubit state returned by* $\mathcal{D}$ *on input* $|x\rangle \, |0^m\rangle$*. The task is to distinguish between two cases for all* $x \in \{0,1\}^n$*:*

- *YES Instance:* $p(\mathcal{D}, x) \geq a$*;*
- *NO Instance:* $p(\mathcal{D}, x) \leq b$*.*

For non-triviality, $b$ is assumed to be not smaller than the probability for any QPT algorithm to correctly guess $F(x)$ without knowing $x \in \{0,1\}^n$. Otherwise, there is a trivial quantum device $\mathcal{D}'$ implementing this guess algorithm, s.t., $p(\mathcal{D}', x) > b$ for all $x \in \{0,1\}^n$. Informally, the problem is to distinguish if $\mathcal{D}$ is a quantum device computing some $F$ with an error probability at most $1 - a$ or at least $1 - b$ for some known $b < a$. For our purpose, we want to have ab interactive proof for the $\mathsf{QBBC}$ problem in the following sense.

**Definition 2.1 (Interactive Proofs for the $\mathsf{QBBC}$ Problem)** *Let* $n$ *be a parameter. The problem* $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ *is said to have an efficient interactive proof with completeness* $c$ *and soundness error* $s$ *(where* $c - s > 1/\mathsf{poly}(n)$*) if there exists a pair of oracle algorithms* $(\mathcal{P}^{\mathcal{D}}, \mathcal{V}^{\mathcal{R}_F})$ *with the following properties:*

- *The prover* $\mathcal{P}$ *makes at most a fixed polynomial number of quantum queries to the device* $\mathcal{D}$*; moreover,* $\mathcal{P}$ *can be efficiently implemented given an efficient device* $\mathcal{D}$*;*

- *The verifier $\mathcal{V}$ makes at most a fixed polynomial number of classical queries to the device $\mathcal{R}_F$; moreover, $\mathcal{V}$ can be efficiently implemented given an efficient device $\mathcal{R}_F$;*

- *$\mathcal{P}$ and $\mathcal{V}$ always terminate after interacting at most a fixed polynomial number of rounds; moreover, they will transmit (quantum) messages with length bounded by a fixed polynomial in $n$ at each round;*

- **Completeness:** *if $(\mathcal{D}, \mathcal{R}_F, a, b)$ is a YES instance, then the probability that $\mathcal{V}^{\mathcal{R}_F}$ accepts after interacting with $\mathcal{P}^{\mathcal{D}}$ is at least c;*

- **Soundness:** *if $(\mathcal{D}, \mathcal{R}_F, a, b)$ is a NO instance, the probability that $\mathcal{V}^{\mathcal{R}_F}$ accepts after interacting with any QPT algorithm $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ is at most s, where $\mathcal{O}_F$ is a classical oracle computing F (i.e., given any $x \in \{0,1\}^n$ as input, $\mathcal{O}_F(x)$ always returns $F(x) \in \{0,1\}^m$).*

Instead of providing the classical oracle $\mathcal{R}_F$ to the malicious prover $\widetilde{\mathcal{P}}$, we allow $\widetilde{\mathcal{P}}$ to access the more powerful classical oracle $\mathcal{O}_F$ that directly computes $F$. This will only increase the capability of the malicious prover $\widetilde{\mathcal{P}}$, and can capture the potential case that $\widetilde{\mathcal{P}}$ may have a device which hardwires some correct pairs $(x, F(x))$.

## 2.4 Extended Noisy Trapdoor Claw-Free Functions

In this subsection, we recall formal definitions of (extended) noisy trapdoor claw-free functions (NTCF) in [BCM+18, Mah18a]. Intuitively, an NTCF family mainly differs from the perfect TCF family we mentioned in the overview in that 1) the range of the functions is not a set $\mathcal{Y}$ but the set $D_{\mathcal{Y}}$ of probability densities over $\mathcal{Y}$; and 2) the functions cannot be perfectly evaluated.

**Definition 2.2 (NTCF Family)** *Let $\lambda$ be a security parameter. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathcal{K}_{\mathcal{F}}$ be a finite set of keys. A family of functions*

$$\mathcal{F} = \{f_{k,b} : \mathcal{X} \to D_{\mathcal{Y}}\}_{k \in \mathcal{K}_{\mathcal{F}}, b \in \{0,1\}}$$

*is called a noisy trapdoor claw-free (NTCF) family if the following conditions hold:*

1. **Efficient Function Generation.** *There exists an efficient probabilistic algorithm $\mathsf{GEN}_{\mathcal{F}}$ which generates a key $k \in \mathcal{K}_{\mathcal{F}}$ together with a trapdoor $t_k$:*

$$(k, t_k) \leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda).$$

2. **Trapdoor Injective Pair.** *For all keys $k \in \mathcal{K}_{\mathcal{F}}$ the following conditions hold:*

   (a) *Trapdoor: For all $b \in \{0,1\}$ and $x \neq x' \in \mathcal{X}$, $\mathsf{SUPP}(f_{k,b}(x)) \cap \mathsf{SUPP}(f_{k,b}(x')) = \emptyset$. Moreover, there exists an efficient deterministic algorithm $\mathsf{INV}_{\mathcal{F}}$ such that for all $b \in \{0,1\}, x \in \mathcal{X}$ and $y \in \mathsf{SUPP}(f_{k,b}(x)), \mathsf{INV}_{\mathcal{F}}(t_k, b, y) = x$.*

   (b) *Injective pair: There exists a perfect matching $\mathcal{R}_k \subseteq \mathcal{X} \times \mathcal{X}$ such that $f_{k,b}(x_0) = f_{k,b}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_k$.*

3. **Efficient Range Superposition.** *For all keys $k \in \mathcal{K}_{\mathcal{F}}$ and $b \in \{0,1\}$ there exists a function $f'_{k,b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}$ such that*

   (a) *For all $(x_0, x_1) \in \mathcal{R}_k$ and $y \in \mathsf{SUPP}(f'_{k,b}(x_b)), \mathsf{INV}_{\mathcal{F}}(t_k, b, y) = x$ and $\mathsf{INV}_{\mathcal{F}}(t_k, b \oplus 1, y) = x_{b \oplus 1}$.*

   (b) *There exists an efficient deterministic procedure $\mathsf{CHK}_{\mathcal{F}}$ that, on input $k, b \in \{0,1\}, x \in \mathcal{X}$ and $y \in \mathcal{Y}$, returns 1 if $y \in \mathsf{SUPP}(f'_{k,b}(x))$ and 0 otherwise.*

(c) *For every $k$ and $b \in \{0,1\}$,*

$$E_{x \leftarrow \mathcal{X}}[H^2(f_{k,b}(x), f'_{k,b}(x))] \leq \mu(\lambda),$$

*for some negligible function $\mu(\cdot)$. Here $H^2$ is the Hellinger distance. Moreover, there exists an efficient procedure $\mathsf{SAMP}_{\mathcal{F}}$ that on input $k$ and $b \in \{0,1\}$ prepares the state*

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(f'_{k,b}(x))(y)}\, |x\rangle\, |y\rangle\,.$$

4. **Adaptive Hardcore Bit.** *For all keys $k \in \mathcal{K}_{\mathcal{F}}$ the following conditions hold, for some integer $w$ that is a polynomially bounded function of $\lambda$.*

   (a) *For all $b \in \{0,1\}$ and $x \in \mathcal{X}$, there exists a set $G_{k,b,x} \subseteq \{0,1\}^w$ such that $\Pr_{d \xleftarrow{\$} \{0,1\}^w}[d \notin G_{k,b,x}]$ is negligible, and moreover there exists an efficient algorithm that checks for membership in $G_{k,b,x}$ given $k, b, x$ and the trapdoor $t_k$.*

   (b) *There is an efficiently computable injection $J : \mathcal{X} \to \{0,1\}^w$, such that $J$ can be inverted efficiently on its range, and such that the following holds. If*

$$H_k = \{(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1))) \mid b \in \{0,1\}, (x_0, x_1) \in \mathcal{R}_k, d \in G_{k,0,x_0} \cap G_{k,1,x_1}\},$$
$$\bar{H}_k = \{(b, x_b, d, c) \mid (b, x, d, c \oplus 1) \in H_k\},$$

   *then for any QPT procedure $\mathcal{A}$ there exists a negligible function $\mu(\cdot)$ such that*

$$\left| \Pr_{(k,t_k) \leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in H_k] - \Pr_{(k,t_k) \leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in \bar{H}_k] \right| \leq \mu(\lambda).$$

In Definition 2.3 we give the definition of trapdoor injective function family.

**Definition 2.3 (Trapdoor Injective Function Family)** *Let $\lambda$ be a security parameter. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathcal{K}_{\mathcal{G}}$ be a finite set of keys. A family of functions*

$$\mathcal{G} = \{g_{k,b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}\}_{k \in \mathcal{K}_{\mathcal{G}}, b \in \{0,1\}}$$

*is called a trapdoor injective family if the following conditions hold:*

1. **Efficient Function Generation.** *There exists an efficient probabilistic algorithm $\mathsf{GEN}_{\mathcal{G}}$ which generates a key $k \in \mathcal{K}_{\mathcal{G}}$ together with a trapdoor $t_k$:*

$$(k, t_k) \leftarrow \mathsf{GEN}_{\mathcal{G}}(1^\lambda).$$

2. **Disjoint Trapdoor Injective Pair.** *For all keys $k \in \mathcal{K}_{\mathcal{G}}$, for all $b, b' \in \{0,1\}$ and $x, x' \in \mathcal{X}$, if $(b,x) \neq (b',x')$, $\mathsf{SUPP}(g_{k,b}(x)) \cap \mathsf{SUPP}(g_{k,b'}(x')) = \emptyset$. Moreover, there exists an efficient deterministic algorithm $\mathsf{INV}_{\mathcal{G}}$ such that for all $b \in \{0,1\}, x \in \mathcal{X}$ and $y \in \mathsf{SUPP}(g_{k,b}(x)), \mathsf{INV}_{\mathcal{G}}(t_k, y) = (b, x)$.*

3. **Efficient Range Superposition.** *For all keys $k \in \mathcal{K}_{\mathcal{F}}$ and $b \in \{0,1\}$*

   (a) *There exists an efficient deterministic procedure $\mathsf{CHK}_{\mathcal{G}}$ that, on input $k, b \in \{0,1\}, x \in \mathcal{X}$ and $y \in \mathcal{Y}$, returns $1$ if $y \in \mathsf{SUPP}(g_{k,b}(x))$ and $0$ otherwise.*

   (b) *There exists an efficient procedure $\mathsf{SAMP}_{\mathcal{G}}$ that on input $k$ and $b \in \{0,1\}$ prepares the state*

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(g_{k,b}(x))(y)}\, |x\rangle\, |y\rangle\,.$$

**Definition 2.4 (Injective Invariance)** *A noisy trapdoor claw-free family $\mathcal{F}$ is injective invariant if there exists a trapdoor injective family $\mathcal{G}$ such that:*

1. *The algorithms $\mathsf{CHK}_{\mathcal{F}}$ and $\mathsf{SAMP}_{\mathcal{F}}$ are the same as the algorithms $\mathsf{CHK}_{\mathcal{G}}$ and $\mathsf{SAMP}_{\mathcal{G}}$;*
2. *For all QPT procedures $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that*
$$\left| \Pr_{(k,t_k)\leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k)=0] - \Pr_{(k,t_k)\leftarrow \mathsf{GEN}_{\mathcal{G}}(1^\lambda)}[\mathcal{A}(k)=0] \right| \le \mu(\lambda).$$

Now, we are ready to define extended NTCF family.

**Definition 2.5 (Extended Noisy Trapdoor Claw-Free Family)** *A noisy trapdoor claw-free family $\mathcal{F}$ is an extended noisy trapdoor claw-free family if:*

1. *It is injective invariant.*
2. *For all $k \in \mathcal{K}_{\mathcal{F}}$ and $d \in \{0,1\}^w$, let*
$$H'_{k,d} = \{d \cdot (J(x_0) \oplus J(x_1)) \mid (x_0, x_1) \in \mathcal{R}_k\}.$$
*For all QPT procedure $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ and a string $d \in \{0,1\}^w$ such that*
$$\left| \Pr_{(k,t_k)\leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in H'_{k,d}] - \frac{1}{2} \right| \le \mu(\lambda).$$

As shown in [BCM+18, Mah18a], one can construct extended NTCF family under the LWE assumption. We refer the reader to [BCM+18, Mah18a] for the details.

# 3 Interactive Proof with a (Limited) Quantum Verifier

Before giving the proof system for the $\mathsf{QBBC}$ problem, we first present an information versus disturbance lemma in Sec. 3.1, which may be of independent interest.

## 3.1 An Information versus Disturbance Lemma

Let $H^1$ be the one-dimensional Hadamard transformation $H$ (i.e., $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$), and let $H^0$ be the identity transformation $\mathsf{Id}$ (i.e., $H^0|b\rangle = |b\rangle$ for any $b \in \{0,1\}$). Then, for any $s = s_1 \dots s_n \in \{0,1\}^n$, we can define a quantum operation $H^s \stackrel{\text{def}}{=} H^{s_1} \otimes H^{s_2} \dots \otimes H^{s_n}$ on $n$ qubits: for any bit string $x = x_1 \dots x_n \in \{0,1\}^n$ and $s = s_1 \dots s_n \in \{0,1\}^n$, denote $|x\rangle_s \stackrel{\text{def}}{=} H^s |x\rangle$ as the quantum encoding $|x\rangle_s = |x_1\rangle_{s_1} \cdots |x_n\rangle_{s_n}$ of $x$, where $|x_i\rangle_{s_i} \stackrel{\text{def}}{=} |x_i\rangle$ if $s_i = 0$, and $|x_i\rangle_{s_i} \stackrel{\text{def}}{=} H|x_i\rangle$ otherwise.

Informally, this lemma says that for randomly chosen $x, s \in \{0,1\}^n$, an algorithm given a quantum state $|x\rangle_s$ cannot obtain sufficient information of $x$ without destroying the state $|x\rangle_s$.

**Lemma 3.1 (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 $|x\rangle_s$ as input, outputs a bit string $e \in \{0,1\}^{\mathsf{poly}(n)}$ and an $n$-qubit quantum state $|\zeta\rangle$, i.e., $(e, |\zeta\rangle) \leftarrow \mathcal{P}(|x\rangle_s)$. Let $x' \in \{0,1\}^n$ be a bit string obtained by first applying $H^s$ on the state $|\zeta\rangle$, and then measuring the state $H^s|\zeta\rangle$ 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*
$$I(x,e) \le n(\alpha + \frac{1}{\alpha} \sum_{\mathsf{hw}(z) \ge 1} \Pr[z])$$
*holds for any $\alpha > 0$, where $\mathsf{hw}(z)$ denotes the hamming-weight of $z \in \{0,1\}^n$ which captures the difference between the two $n$-bit strings $x, x' \in \{0,1\}^n$.*

Note that if $|\zeta\rangle = |x\rangle_s$, 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 of $x$ from the state $|x\rangle_s$ will necessarily disturb the state. The proof of Lemma 3.1 will use some techniques introduced in the security proof [BBB$^+$06] of the BB84 quantum key distribution.

**Proof.** Without loss of generality, we can assume that $\mathcal{P}$ works as follows: given an input state $|x\rangle_s$ 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 [BBB$^+$06], 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, the authors of [BBB$^+$06] 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 [BBB$^+$06, 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 $x = (x_1, \ldots, x_n), t = (t_1, \ldots, t_n) \in \{0,1\}^n$ 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

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

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

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

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

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

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 $|\phi_t\rangle_s$ is a pure state (which implies $\langle\phi_t|\phi_t\rangle_s=1$ for any $t\in\{0,1\}^n$), we have that

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

Now, 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 $i\in\{1,\dots,n\}$. 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 [Fv99], denoted 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 $\rho_0(v_i) - \rho_1(v_i)$ [Fv99]. 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

$$\begin{aligned}
SD(\rho_0(v_i), \rho_1(v_i)) &\leq \tfrac{1}{2} \|\rho_0(v_i) - \rho_1(v_i)\|_1 \\
&\leq \tfrac{1}{2^n} \left\| \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} \, |\hat{\gamma}_t\rangle_s \, {}_s\langle \hat{\gamma}_{t \oplus v_i}| \right\|_1 \\
&= \tfrac{1}{2^{n+1}} \left\| \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} (|\hat{\gamma}_t\rangle_s \, {}_s\langle \hat{\gamma}_{t \oplus v_i}| + |\hat{\gamma}_{t \oplus v_i}\rangle_s \, \langle \hat{\gamma}_t|_s) \right\|_1 \\
&\leq \tfrac{1}{2^n} \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} (\tfrac{1}{2} \| \, |\hat{\gamma}_t\rangle_s \, \langle \hat{\gamma}_{t \oplus v_i}|_s + |\hat{\gamma}_{t \oplus v_i}\rangle_s \, {}_s\langle \hat{\gamma}_t| \|_1) \\
&= \tfrac{1}{2^n} \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} \sqrt{1 - (\mathrm{Im}(\langle \hat{\gamma}_t | \hat{\gamma}_{t \oplus v_i} \rangle_s))^2} \\
&= \tfrac{1}{2^n} \textstyle\sum_{s,t} d_{t,s} d_{t \oplus v_i, s} \\
&= \tfrac{1}{2^n} \textstyle\sum_s \left( \sum_{\mathsf{hw}(t) \geq 1} d_{t,s} d_{t \oplus v_i, s} + \sum_{\mathsf{hw}(t) = 0} d_{t,s} d_{t \oplus v_i, s} \right) \\
&\leq \tfrac{1}{2^n} \textstyle\sum_s \left( \sum_{\mathsf{hw}(t) \geq 1} d_{t,s} d_{t \oplus v_i, s} + \sum_{\mathsf{hw}(t \oplus v_i) \geq 1} d_{t,s} d_{t \oplus v_i, s} \right),
\end{aligned}$$

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

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

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

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

holds for any $\alpha > 0$.

By Equation (6), we have

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

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

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

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 $\left|x'\right\rangle_{\bar{s}} = \sum_x 2^{-n/2}(-1)^{x'\cdot x}\left|x\right\rangle_s$ and $\left|x\right\rangle_s = \sum_{x'} 2^{-n/2}(-1)^{x\cdot x'}\left|x'\right\rangle_{\bar{s}}$, we have that

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

This implies that

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

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

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

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. □

### 3.2 The Interactive Proof System with a (Limited) Quantum Verifier

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 $S$. Correspondingly, 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 present our first interactive proof for the QBBC problem. The full description of the proof system is given in Protocol 1.

For our purpose, it suffices to prove the following two theorems.

**Theorem 3.1 (Completeness)** *For the promise problem* $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$, *in an honest execution of Protocol 1 between a classical verifier* $\mathcal{V}^{\mathcal{R}_F}$ *and a quantum prover* $\mathcal{P}^{\mathcal{D}}$, *the verifier* $\mathcal{V}^{\mathcal{R}_F}$ *will accept a YES instance with probability at least* $\frac{1+a}{2}$.

**Protocol 1:** Interactive Proof with a (Limited) Quantum Verifier

**Inputs:** The quantum algorithm $\mathcal{V}$ has classical access to $\mathcal{R}_F$. The quantum algorithm $\mathcal{P}$ has quantum access to $\mathcal{D}$.

**Description:**

1. The verifier $\mathcal{V}$ uniformly chooses bit-strings $x, s \xleftarrow{\$} \{0,1\}^n$ at random, and prepares a quantum state $|x\rangle$ of $n$ 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 $x, s \in \{0,1\}^n$, we can rewrite
$$|x\rangle_s = H^s |x\rangle = \frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\tilde{x} \in X} (-1)^{x^s \cdot \tilde{x}^s} |\tilde{x}\rangle .$$
Send a quantum challenge message $|\phi\rangle = |x\rangle_s$ to the prover;
2. The prover $\mathcal{P}$ sends $|\phi\rangle |0^m\rangle$ to the device $\mathcal{D}$, and returns the $n+m$ qubits $|\psi\rangle$ received from $\mathcal{D}$ to the verifier;
3. The verifier $\mathcal{V}$ first picks a bit $\delta \xleftarrow{\$} \{0,1\}$ at random, and sends back $|\psi\rangle$ to the prover if $\delta = 1$. Otherwise, make a standard measurement on $|\psi\rangle$ to obtain a pair of $\tilde{x} \in \{0,1\}^n$ and $\tilde{y} \in \{0,1\}^m$, and send a classical query $(\tilde{x}, \tilde{y})$ to the device $\mathcal{R}_F$. If $\tilde{x}^{\bar{s}} \neq x^{\bar{s}}$ (i.e., $\tilde{x}^{\bar{s}} \notin X$) or $\mathcal{R}_F(\tilde{x}, \tilde{y}) = 0$, set *accept* $= 0$ and abort, else set *accept* $= 1$ and abort.
4. The prover $\mathcal{P}$ sends a quantum query with state $|\psi\rangle$ to the device $\mathcal{D}$, and returns a state $|\zeta\rangle$ containing the first $n$ qubits of the state received from $\mathcal{D}$ to the verifier;
5. The verifier $\mathcal{V}$ applies $H^s$ to the state $|\zeta\rangle$, and obtain a state $|\zeta'\rangle$. Measure the state $|\zeta'\rangle$ to obtain $\hat{x} \in \{0,1\}^n$. If $\hat{x} \neq x$, set *accept* $= 0$, else set *accept* $= 1$.

**Output:** $\mathcal{V}$ outputs *accept*.

**Proof of Theorem 3.1.** After receiving $|\phi\rangle = |x\rangle_s$ from the verifier, the prover $\mathcal{P}$ will send a quantum query with state $|\phi\rangle |0^m\rangle$ to $\mathcal{D}$, and obtain a state $|\psi\rangle$ from $\mathcal{D}$, where
$$|\psi\rangle = \frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\tilde{x} \in X} (-1)^{x^s \cdot \tilde{x}^s} \mathcal{D}(|\tilde{x}, 0^m\rangle).$$

After receiving $|\psi\rangle$ from the prover, the verifier $\mathcal{V}$ first picks a random bit $\delta \xleftarrow{\$} \{0,1\}$. If $\delta = 0$, measuring the state $|\psi\rangle$ will 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 $\tilde{y} = F(\tilde{x})$ with probability at least $a$, which means that $\mathcal{V}$ will set *accept* $= 1$ with probability at least $a$ when $\delta = 0$. If $\delta = 1$, $\mathcal{V}$ will first send back $|\psi\rangle$ to the prover. The prover $\mathcal{P}$ will send a quantum query with state $|\psi\rangle$ to $\mathcal{D}$ (to uncompute the first application of $\mathcal{D}$), and obtain a state from $\mathcal{D}$:
$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\tilde{x} \in X} (-1)^{x^s \cdot \tilde{x}^s} |\tilde{x}\rangle |0^m\rangle = |x\rangle_s |0^m\rangle .$$

After receiving the state $|\zeta\rangle = |x\rangle_s$ containing the first $n$ qubits of the above state from $\mathcal{P}$, the verifier $\mathcal{V}$ will apply $H^s$ on the state $|\zeta\rangle$, and obtain a state $|\zeta'\rangle = |x\rangle$. Thus, measuring the state $|\zeta'\rangle$ will result in $\hat{x} = x$, which means that $\mathcal{V}$ will always set *accept* $= 1$ when $\delta = 1$.

Since $\delta$ is uniformly chosen from $\{0,1\}$, we have that $\mathcal{V}^{\mathcal{R}_F}$ will output *accept* $= 1$ with probability at least $\frac{1+a}{2}$. □

**Theorem 3.2 (Soundness)** *For the promise problem* $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$ *with constant* $\frac{15}{16} < a \leq 1$, *any classical device* $\mathcal{O}_F$ *computing* $F$, *and any QPT algorithm* $\widetilde{\mathcal{P}}^{\mathcal{D}, \mathcal{O}_F}$, *the probability that* $\mathcal{V}^{\mathcal{R}_F}$ *accepts a NO instance is at most* $\frac{31}{32} + \frac{\epsilon_1}{2} + \mathsf{negl}(n)$ *for any constant* $\max(0, b - \frac{15}{16}) < \epsilon_1 < a - \frac{15}{16}$.

**Proof of Theorem 3.2.** Note that after obtaining the first response $|\psi\rangle$ from $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ to the challenge message $|\phi\rangle = |x\rangle_s$, the verifier $\mathcal{V}$ will first uniformly pick a bit $\delta \xleftarrow{\$} \{0,1\}$ at random, and then perform different checks depending on the value of $\delta$. Let $\vartheta_0$ and $\vartheta_1$ be the probabilities that $|\psi\rangle$ passes the checks for $\delta = 0$ and $\delta = 1$, respectively. Then, the probability that $\mathcal{V}$ will output *accept* $= 1$ is $(\vartheta_0 + \vartheta_1)/2$.

Let $z = \hat{x} \oplus x$ be the difference between the random string $x \in \{0,1\}^n$ chosen in step 1 and the bit string $\hat{x} \in \{0,1\}^n$ obtained by the verifier $\mathcal{V}$ in step 5. By definition, we have that $\vartheta_1 \leq \Pr[z = 0^n]$. We now give a bound on $\vartheta_0$. Let $E$ be the event that $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ makes a classical query $\tilde{x} \in \{0,1\}^n$ to the device $\mathcal{O}_F$ such that $\tilde{x}^{\bar{s}} = x^{\bar{s}}$ (note that for any $\tilde{x}$ known for $\widetilde{\mathcal{P}}$, making a query $\tilde{x}$ to $\mathcal{O}_F$ is the best strategy to obtain a correct $F(\tilde{x})$).

Note that if $E$ does not happen, the probability that $\mathcal{V}$ sets *accept* $= 1$ in step 3 is at most $b$ by the assumption. Thus, we have $\vartheta_0 \leq b + \Pr[E]$. Moreover, let $\mu = 1/2 - \nu$ for some $0 < \nu < 1/2$, by the law of total probability we have that

$$\begin{aligned} \Pr[E] &= \Pr[E\,|\mathsf{hw}(\bar{s}) \leq \mu n] \cdot \Pr[\mathsf{hw}(\bar{s}) \leq \mu n] + \Pr[E\,|\mathsf{hw}(\bar{s}) > \mu n] \cdot \Pr[\mathsf{hw}(\bar{s}) > \mu n] \\ &\leq \Pr[\mathsf{hw}(\bar{s}) \leq \mu n] + \Pr[E|\mathsf{hw}(\bar{s}) > \mu n]. \end{aligned} \tag{9}$$

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

$$\Pr[E\,|\mathsf{hw}(\bar{s}) > \mu n] \leq \frac{\ell}{2^{\mathsf{hw}(\bar{s}) - I(x^{\bar{s}},e)}} \leq \frac{\ell}{2^{\mu n - I(x,e)}}, \tag{10}$$

where $\ell = \mathsf{poly}(n)$ is the number of classical $\mathcal{O}_F$ queries made by $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ (since $I(x,e)$ bounds the information that $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$ obtains about $x$ from above). Thus, we have that

$$\vartheta_0 \leq b + e^{-n\nu^2} + \frac{\ell}{2^{\mu n - I(x,e)}}. \tag{11}$$

Since the choice of $\delta$ is random and independent from $x$ and $s$, the algorithm $\widetilde{\mathcal{P}}$ cannot obtain more information about $(x,s)$ when $\delta = 0$. We can use Lemma 3.1 to establish a connection between $I(x,e)$ and $\vartheta_1$ (because for $\delta = 1$, the verifier will send back the state $|\psi\rangle$ to the prover, one can think that $\widetilde{\mathcal{P}}$ directly outputs $|\zeta\rangle$ after receiving the query $|\phi\rangle$ from the verifier):

$$I(x,e) \leq n(\alpha + \frac{1}{\alpha}\sum_{\mathsf{hw}(z)\geq 1} \Pr[z]) = n(\alpha + \frac{1}{\alpha}(1 - \Pr[z = 0^n])) \leq n(\alpha + \frac{1}{\alpha}(1 - \vartheta_1)), \tag{12}$$

where $\alpha > 0$ is an arbitrary real. Let $0 < \epsilon_1 < 1/16$ be an arbitrary constant. We now proceed the proof by discussing the value of $\vartheta_1$.

If $\vartheta_1 < 15/16 + \epsilon_1$, we immediately have

$$\frac{\vartheta_0 + \vartheta_1}{2} < \frac{31}{32} + \frac{\epsilon_1}{2}. \tag{13}$$

If $15/16 + \epsilon_1 \leq \vartheta_1 \leq 1$, we claim that there exists a constant $\epsilon_2 > 0$, s.t., $I(x,e) \leq (1/2 - \epsilon_2)n$. If $\vartheta_1 = 1$, this obviously holds as $\alpha > 0$ can be an arbitrary constant in Eq (12). If $15/16 + \epsilon_1 < \vartheta_1 < 1$, we have $I(x,e) \leq 2n\sqrt{1 - \vartheta_1}$ by setting $\alpha = \sqrt{1 - \vartheta_1}$ in Eq (12). Using the fact that $\sqrt{1 - \vartheta_1} \leq \sqrt{1/16 - \epsilon_1}$, we have that claim that there exists a constant $\epsilon_2 > 0$, s.t., $I(x,e) \leq (1/2 - \epsilon_2)n$. By appropriately choosing a constant $0 < \nu < \epsilon_2$, we can have $\mu n - I(x,e) = \epsilon_3 n$ for some constant $\epsilon_3 > 0$, which means that $\vartheta_0 \leq b + \mathsf{negl}(n)$ by Eq (11). Thus, we have

$$\frac{\vartheta_0 + \vartheta_1}{2} \leq \frac{1 + b}{2} + \mathsf{negl}(n). \tag{14}$$

By choosing constant $\epsilon_1$, s.t., $\max(0, b - \frac{15}{16}) < \epsilon_1 < a - \frac{15}{16}$, we can always have that $\frac{\vartheta_0+\vartheta_1}{2} < \frac{31}{32} + \frac{\epsilon_1}{2} + \mathsf{negl}(n)$ holds. This completes the proof. □

# 4 Interactive Proof with a Fully Classical Verifier

At a high level, the verifier $\mathcal{V}$ in Section 3 needs quantum capabilities for two main goals: 1) generating a quantum state with a verifiable hidden set $X$; and 2) measuring the quantum state from the prover to obtain a correct pair $(x, F(x))$ for $x \in X$. In this section, we show that under the LWE assumption, a classical verifier can achieve the same goals by delegating the quantum computations to the prover. For this, we will first present a protocol for generating a quantum state with a verifiable hidden set in Section 4.1 by modifying the randomness expansion protocol in [BCM$^+$18], and an oblivious measurement protocol in Sec. 4.2 by tailoring the protocol in [Mah18a].

## 4.1 Generation of Quantum State with Verifiable Hidden Set

Let $\lambda$ be a security parameter. Let $n = \mathsf{poly}(\lambda)$, and let $\gamma, q > 0$ be functions of $\lambda$ and $n$. Let $\mathcal{F}$ be an extended NTCF family, and $\mathcal{G}$ be its corresponding trapdoor injective family. Now, we will give a protocol which allows a classical verifier $\mathcal{V}$ and a quantum prover $\mathcal{P}$ to cooperatively generate a state with a verifiable hidden set such that the prover holds the state without knowing the corresponding hidden set held by the verifier. For our purpose, we describe the protocol in two phases: the generation phase and the verification phase as depicted in Protocol 2.

We have the following two theorems for Protocol 2.

**Theorem 4.1** *If $\mathcal{F}$ is an extended NTCF family, and $\mathcal{G}$ is the corresponding trapdoor injective family. Let $s = s_1 \dots s_n \in \{0,1\}^n$ be the secret bit string chosen by $\mathcal{V}$, and let $\hat{Y} = (\hat{y}_1, \dots, \hat{y}_n) \in \mathcal{Y}^n$ be the set output by $\mathcal{P}$ in the generation phase. Moreover, let $\hat{x}_{r_i,\hat{y}_i} = \mathsf{INV}_{\mathcal{F}}(t_i, r_i, \hat{y}_i)$ for any $r_i \in \{0,1\}$ if $s_i = 1$, and $(\hat{r}_i, \hat{x}_{\hat{r}_i,\hat{y}_i}) = \mathsf{INV}_{\mathcal{G}}(t_i, \hat{y}_i)$ otherwise. Then, in an honest execution of Protocol 2 between a classical verifier $\mathcal{V}$ and a quantum algorithm $\mathcal{P}$, the quantum state $|\phi\rangle$ obtained by $\mathcal{P}$ in the generation phase is within negligible trace distance from the following state:*

$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\substack{r = r_1 \dots r_n \in R_{s,\hat{Y}}, \\ \hat{X}_{r,\hat{Y}} = \hat{x}_{r_1,\hat{y}_1} \dots \hat{x}_{r_n,\hat{y}_n} \in \mathcal{X}^n}} |r\rangle \left|\hat{X}_{r,\hat{Y}}\right\rangle,$$

*where $R_{s,\hat{Y}} = \{r = r_1 \dots r_n \in \{0,1\}^n : s_i = 0 \wedge r_i = \hat{r}_i\}$. Moreover, the probability that $\mathcal{V}$ outputs $1$ is negligibly close to $1$.*

**Proof.** By definition, the quantum state $|\phi\rangle$ is obtained by directly measuring the register containing $Y$ of the state $|\phi_0\rangle$. We first show that $|\phi_0\rangle$ is within negligible trace distance from the following state:

$$\left|\phi_0^{(n)}\right\rangle = \frac{1}{2^{n/2}|\mathcal{X}|^{n/2}} \sum_{\substack{r = r_1 \dots r_n \in \{0,1\}^n, \\ X = x_1 \dots x_n \in \mathcal{X}^n, \\ Y = y_1 \dots y_n \in \mathcal{Y}^n}} \alpha'_{X,Y} |r\rangle |X\rangle |Y\rangle,$$

where $\alpha'_{X,Y} = \prod_i \sqrt{\hat{g}_{k_i,r_i}(x_i)(y_i)}$, $\hat{g}_{k_i,b} = f_{k_i,b}$ if $s_i = 1$, otherwise $\hat{g}_{k_i,b} = g_{k_i,b}$. Clearly, $\alpha'_{X,Y}$ only differs from $\alpha_{X,Y}$ at position $s_i = 1$. Now, we define a set of states $\left|\phi_0^{(1)}\right\rangle, \dots, \left|\phi_0^{(n)}\right\rangle$ such that $\left|\phi_0^{(1)}\right\rangle = |\phi_0\rangle$, and for $i = 1, \dots, n$ the state $\left|\phi_0^{(i)}\right\rangle$ is obtained from $\left|\phi_0^{(i-1)}\right\rangle$ by replacing $g'_{k_i,b}$ with $\hat{g}_{k_i,b}$. As $n$ is a polynomial in $\lambda$, in order to show that the trace distance between $|\phi_0\rangle$ and $\left|\phi_0^{(n)}\right\rangle$ is negligible, it suffices to show that for any $i > 1$, the trace distance between $\left|\phi_0^{(i)}\right\rangle$ and $\left|\phi_0^{(i-1)}\right\rangle$ is negligible in $\lambda$, namely,

$$\left\| \left|\phi_0^{(i)}\right\rangle - \left|\phi_0^{(i-1)}\right\rangle \right\|_{tr} \leq \mathsf{negl}(\lambda).$$

**Protocol 2:** Generation of Quantum State with Verifiable Hidden Set

**Inputs:** The classical algorithm $\mathcal{V}$ inputs an integer $n$ and a real $q \in (0,1)$. The quantum algorithm $\mathcal{P}$ inputs an integer $n$.

**Generation:** This is to generate a quantum state held by $\mathcal{P}$.

1. For $i = 1, \dots, n$, $\mathcal{V}$ selects $s_i \in \{0,1\}$ such that $\Pr[s_i = 1] = q$. Then, it generates $(k_i, t_i) \leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda)$ if $s_i = 1$, otherwise $(k_i, t_i) \leftarrow \mathsf{GEN}_{\mathcal{G}}(1^\lambda)$. Send the keys $K = (k_1, \dots, k_n)$ to $\mathcal{P}$.

2. $\mathcal{P}$ first applies the Hadamard operation to $n$ qubits containing 0 to obtain a state
$$\frac{1}{2^{n/2}} \sum_{r = r_1 \dots r_n \in \{0,1\}^n} |r\rangle = \frac{1}{2^{n/2}} \sum_{r = r_1 \dots r_n \in \{0,1\}^n} |r_1\rangle \dots |r_n\rangle .$$
Then, for $i = 1, \dots, n$, it applies the $\mathsf{SAMP}_{\mathcal{F}} = \mathsf{SAMP}_{\mathcal{G}}$ procedure in superposition with $k_i$ and the $i$-th qubit containing $r_i$ as input. Let $g'_{k_i,b} = f'_{k_i,b}$ if $s_i = 1$, and $g'_{k_i,b} = g_{k_i,b}$ otherwise. $\mathcal{P}$ will obtain a state
$$|\phi_0\rangle = \frac{1}{2^{n/2}|\mathcal{X}|^{n/2}} \sum_{\substack{r = r_1 \dots r_n \in \{0,1\}^n, \\ X = x_1 \dots x_n \in \mathcal{X}^n, \\ Y = y_1 \dots y_n \in \mathcal{Y}^n}} \alpha_{X,Y} |r\rangle |X\rangle |Y\rangle ,$$
where $\alpha_{X,Y} = \prod_i \sqrt{g'_{k_i,r_i}(x_i)(y_i)}$. Finally, $\mathcal{P}$ measures the registers containing $Y$ to obtain a set $\hat{Y} = (\hat{y}_1, \dots, \hat{y}_n) \in \mathcal{Y}^n$ and a state $|\phi\rangle$. Send $\hat{Y}$ to $\mathcal{V}$.

**Verification:** This is to verify the quantum state $|\phi\rangle$ holding by $\mathcal{P}$.

3. Let $\mathsf{cst} = 1$ be a consistence flag held by the verifier. For $i = 1, \dots, n$, $\mathcal{P}$ and $\mathcal{V}$ works as follows:

   3.1 $\mathcal{V}$ randomly chooses $c_i \xleftarrow{\$} \{0,1\}$ if $s_i = 1$, otherwise set $c_i = 1$. Then, it sends $c_i$ to $\mathcal{P}$.

   3.2 $\mathcal{P}$ performs the following computation depending on the value of $c_i$:

   (a) In case $c_i = 0$, evaluate the function $J$ on the qubits containing $x_i$, and then apply a Hadamard transform to the $w+1$ qubits containing $r_i$ and $x_i$. Measure the $w+1$ registers to obtain a pair $(\hat{u}_i, \hat{d}_i) \in \{0,1\} \times \{0,1\}^w$. Send $(\hat{u}_i, \hat{d}_i)$ to $\mathcal{V}$;

   (b) In case $c_i = 1$, measure the $w+1$ qubits containing $(r_i, x_i)$ to obtain a pair $(\hat{r}_i, \hat{x}_i) \in \{0,1\} \times \mathcal{X}$. Send $(\hat{r}_i, \hat{x}_i)$ to $\mathcal{V}$.

   3.3 $\mathcal{V}$ performs the following computation depending on the value of $c_i$:

   (a) In case $c_i = 0$, randomly choose $W_i \xleftarrow{\$} \{0,1\}$ if $\hat{d}_i \notin \hat{G}_{y_i} = G_{k_i,0,\hat{x}_{0,\hat{y}_i}} \cap G_{k_i,1,\hat{x}_{1,\hat{y}_i}}$. Otherwise, set $W_i = 1$ if $\hat{d}_i \cdot (J(\hat{x}_{0,\hat{y}_i}) \oplus J(\hat{x}_{1,\hat{y}_i})) = \hat{u}_i$, and $W_i = 0$ if not, where $\hat{x}_{b,\hat{y}_i} = \mathsf{INV}_{\mathcal{F}}(t_i, b, \hat{y}_i)$ for $b \in \{0,1\}$.

   (b) In case $c_i = 1$, set $W_i$ as the value returned by $\mathsf{CHK}_{\mathcal{F}}(k_i, \hat{r}_i, \hat{x}_i, \hat{y}_i)$ if $s_i = 1$, otherwise set $\mathsf{cst} = 0$ if $\mathsf{CHK}_{\mathcal{G}}(k_i, \hat{r}_i, \hat{x}_i, \hat{y}_i) = 0$.

**Outputs:** $\mathcal{V}$ outputs 0 if $\sum_{i:s_i=1} W_i < (1-\gamma)qn$ or $\mathsf{cst} = 0$, and 1 otherwise.

This obviously holds for $s_i = 0$, because in this case $g'_{k_i,b} = \hat{g}_{k_i,b}$ and $\left|\phi_0^{(i)}\right\rangle = \left|\phi_0^{(i-1)}\right\rangle$. Thus, we only have to consider the case $s_i = 1$, where $g'_{k_i,b} = f'_{k_i,b}$ and $\hat{g}_{k_i,b} = f_{k_i,b}$. As for every $k$ and $b \in \{0,1\}$, $E_{x \xleftarrow{\$} \mathcal{X}}[H^2(f_{k,b}(x), f'_{k,b}(x))] \le \mu(\lambda)$ for some negligible function $\mu(\cdot)$ by the assumption, we have that
$$\left\| \left|\phi_0^{(i)}\right\rangle - \left|\phi_0^{(i-1)}\right\rangle \right\|_{tr} \le \mathsf{negl}(\lambda).$$

By the property of trace distance, we have that $|\phi\rangle$ is within negligible trace distance from the state obtained by measuring the state

$$|\psi\rangle = \frac{1}{2^{n/2}|\mathcal{X}|^{n/2}} \sum_{\substack{r = r_1 \dots r_n \in \{0,1\}^n, \\ X = x_1 \dots x_n \in \mathcal{X}^n, \\ Y = y_1 \dots y_n \in \mathcal{Y}^n}} \alpha'_{X,Y} |r\rangle |X\rangle |Y\rangle .$$

to obtain $\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_n\}$. By the injective pair property of $\mathcal{F}$ and the disjoint trapdoor injective pair property of $\mathcal{G}$, we have that $|\phi\rangle$ is within negligible trace distance from the state

$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\substack{r = r_1 \dots r_n \in R_{s,\hat{Y}}, \\ \hat{X}_{r,\hat{Y}} = \hat{x}_{r_1,\hat{y}_1} \dots \hat{x}_{r_n,\hat{y}_n} \in \mathcal{X}^n}} |r\rangle \left| \hat{X}_{r,\hat{Y}} \right\rangle .$$

This completes the proof of the first claim.

As for the second claim, by the Chernoff bound it suffices to show that $W_i = 1$ almost always holds for $s_i = 1$. By the definition of an extended NTCF, $\mathcal{D}$ will set $W_i = 0$ for an honest $\mathcal{P}$ only if $\mathcal{P}$ outputs a $\hat{d}_i \notin \hat{G}_{\hat{y}_i}$, which by item 4(a) in Definition 2.2 happens with negligible probability. This completes the proof. □

**Theorem 4.2** *If $\mathcal{F}$ is an extended NTCF family, and $\mathcal{G}$ is the corresponding trapdoor injective family. Let $s = s_1 \dots s_n \in \{0,1\}^n$ be the secret bit string chosen by $\mathcal{V}$, and let $\hat{Y} = (\hat{y}_1, \dots, \hat{y}_n) \in \mathcal{Y}^n$ be the set output by $\mathcal{P}$ in the generation phase. Let $\hat{r} \in \{0,1\}^{\mathsf{hw}(\bar{s})}$ be the string obtained by concatenating each bit $\hat{r}_i \in \{0,1\}$ output by $\mathcal{P}$ satisfying $s_i = 0$ in the verification phase, and let $R_{s,\hat{Y}} = \{r = r_1 \dots r_n \in \{0,1\}^n : r^{\bar{s}} = \hat{r}\}$. Let $E$ be the event that $\mathcal{P}$ outputs a $r \in R_{s,\hat{Y}}$ before the verification phase. Let $A$ be the event that $\mathcal{V}$ outputs 1 in the verification phase. Then, for any QPT algorithm $\mathcal{P}$, we have that* $\Pr[E|A] \leq \mathsf{negl}(\lambda)$.

**Proof.** We first consider a modification of Protocol 2 where $\mathcal{V}$ also generates the keys by running $(k_i, t_i) = \mathsf{GEN}_{\mathcal{F}}(1^\lambda)$ for all $s_i = 0$. Then, we can similarly define events $E'$ and $A'$ the same as $E$ and $A$ for the modified protocol. By the assumption that no QPT algorithm can distinguish the keys generated by $\mathsf{GEN}_{\mathcal{F}}(1^\lambda)$ from that generated by $\mathsf{GEN}_{\mathcal{G}}(1^\lambda)$. We have the $|\Pr[E'|A'] - \Pr[E|A]| \leq \mathsf{negl}(\lambda)$ for any QPT algorithm $\mathcal{P}$ (note that the verifier $\mathcal{V}$ will not use $t_i$ for $s_i = 0$ during the whole interactions).

Thus, it suffices to show that $\Pr[E'|A'] \leq \mathsf{negl}(\lambda)$. We note that the above modified protocol is essentially identical to the randomness expansion protocol in [BCM+18] except the following two differences:

1. The verifier in the randomness expansion protocol [BCM+18] will only generate a fresh $k_i$ if $i = 1$ or $s_{i-1} = 1$, and will reuse the same key $k_i = k_{i+1} = \cdots = k_j$ for any $i < j$ satisfying $s_i = \cdots = s_{j-1} = 0$ and $s_j = 1$. This is necessary for saving the random bits to generate the keys, because the length of the output random bits of a meaningful randomness expansion protocol must be longer than that of the input ones (which is not required in our protocol);

2. At the end of the randomness expansion protocol [BCM+18], the verifier will either reject and abort if $\sum_{i:s_i=1} W_i < (1-\gamma)qn$, or output a string $\hat{r} \in \{0,1\}^{\mathsf{hw}(\bar{s})}$ obtained by concatenating $\hat{r}_i \in \{0,1\}$ for all $s_i = 0$. Besides, a consistence flag $\mathsf{cst}$ is added in Protocol 2 to ensure that the bit $\hat{r}_i \in \{0,1\}$ returned by the prover for some $s_i = 0$ in the verification phase is always equal to the one computed by the verifier using $(\hat{r}_i, \hat{x}_{\hat{r}_i,\hat{y}_i}) = \mathsf{INV}_{\mathcal{G}}(t_i, \hat{y}_i)$ from $\hat{y}_i$ (we note that this is achieved by simply checking if $\mathsf{CHK}_{\mathcal{G}}(k_i, \hat{r}_i, \hat{x}_i, \hat{y}_i) = 0$ without using the trapdoor $t_i$, that similar checks are implicitly done in [BCM+18]).

Clearly, the way of using a fresh $k_i$ for all $i \in \{1, \ldots, n\}$ will not give more advantage to $\mathcal{P}$, and the output of $\mathcal{V}$ will not affect the view of $\mathcal{P}$. In fact, by almost the same proof as that for [BCM$^+$18, Proposition 8.9], we can show that for any QPT algorithm $\mathcal{P}$, there is a negligible function $\epsilon(\lambda)$ such that if $\mathcal{V}$ outputs 1, then the $\epsilon$-smooth min-entropy of $\hat{r}$ conditioned on all the information obtained by $\mathcal{P}$ before the verification phase is at least $O(n)$ under appropriate choices of parameters. We refer the reader to [BCM$^+$18] for the details. Thus, for the modified protocol, the probability that $\mathcal{P}$ outputs a string $r \in R_{s,\hat{Y}} = \{r = r_1 \ldots r_n \in \{0,1\}^n : r^{\bar{s}} = \hat{r}\}$ before the verification phase, conditioned on the event that $\mathcal{V}$ outputs 1, is negligible under appropriate choices of parameters, namely, $\Pr[E'|A'] \leq \mathsf{negl}(\lambda)$. This completes the proof. □

## 4.2 Oblivious Measurement on Quantum State

Let $\lambda$ be the security parameter. Let $\mathcal{F}$ be an extended NTCF family associated with a corresponding trapdoor injective family $\mathcal{G}$. Suppose that there is a classical verifier $\mathcal{V}$ holding a secret bit $\delta \in \{0,1\}$ and a quantum prover $\mathcal{P}$ holding an arbitrary state $|\psi\rangle$ (which does not necessarily generated by $\mathcal{P}$). We now give a protocol between $\mathcal{V}$ and $\mathcal{P}$ such that depending on the value of $\delta$, $\mathcal{V}$ either obtains the measurement outcome of $|\psi\rangle$ or holds some information that can help $\mathcal{P}$ to compute a state with trace distance negligibly close to the input state $|\psi\rangle$, without leaking the information of $\delta$ to $\mathcal{P}$ (i.e., $\mathcal{P}$ does not know which case it is for $\mathcal{V}$). The full description of the protocol is given in Protocol 3, which is based on the protocol in [Mah18a].

**Protocol 3:** Oblivious Measurement on Quantum State

**Inputs:** The classical algorithm $\mathcal{V}$ inputs an integer $n$ and a bit $\delta \in \{0,1\}$. The quantum algorithm $\mathcal{P}$ inputs a quantum state $|\psi\rangle = \sum_{r=r_1 \ldots r_n \in \{0,1\}^n} \beta_r |r_1, \ldots, r_n\rangle$.

**Description:** This is the description of the oblivious measurement protocol.

1. For $i = 1, \ldots, n$, $\mathcal{V}$ generates $(k_i, t_i) \leftarrow \mathsf{GEN}_{\mathcal{G}}(1^\lambda)$ if $\delta = 0$, otherwise $(k_i, t_i) \leftarrow \mathsf{GEN}_{\mathcal{F}}(1^\lambda)$. Then, send $K = \{k_i\}_{i \in \{1,\ldots,n\}}$ to $\mathcal{P}$.
2. For $i = 1, \ldots, n$, $\mathcal{P}$ first applies the $\mathsf{SAMP}_{\mathcal{F}} = \mathsf{SAMP}_{\mathcal{G}}$ procedure in superposition with $k_i$ and the $i$-th qubit containing $r_i$ of $|\psi\rangle$ as input. Let $g'_{k_i,b} = g_{k_i,b}$ if $\delta = 0$, otherwise $g'_{k_i,b} = f'_{k_i,b}$. This will lead to the following state
$$\frac{1}{|\mathcal{X}|^{n/2}} \sum_{\substack{r = r_1 \ldots r_n \in \{0,1\}^n, \\ X = x_1 \ldots x_n \in \mathcal{X}^n, \\ Y = y_1 \ldots y_n \in \mathcal{Y}^n}} \alpha_{X,Y} |r\rangle |X\rangle |Y\rangle,$$
where $\alpha_{X,Y} = \beta_r \prod_i \sqrt{g'_{k_i,r_i}(x_i)(y_i)}$. Then, $\mathcal{P}$ measures the $\mathcal{Y}$ registers to obtain a set $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_n) \in \mathcal{Y}^n$ and a state $|\psi'\rangle$. Send $\hat{Y}$ to $\mathcal{V}$.
3. $\mathcal{V}$ computes $(\hat{r}_i, \hat{x}_{\hat{r}_i,\hat{y}_i}) = \mathsf{INV}_{\mathcal{G}}(t_i, \hat{y}_i)$ for each $\hat{y}_i \in \mathcal{Y}$ if $\delta = 0$.

**Outputs:** $\mathcal{V}$ outputs $\hat{r} = \hat{r}_1 \ldots \hat{r}_n$ if $\delta = 0$, and $KT = \{(k_i, t_i)\}$ otherwise; $\mathcal{P}$ outputs a state $|\psi'\rangle$.

We have the following two theorems for Protocol 3.

**Theorem 4.3** *In an honest execution of Protocol 3 between a classical verifier $\mathcal{V}$ with input $\delta$ and a quantum algorithm $\mathcal{P}$ with input a state $|\psi\rangle = \sum_{r=r_1 \ldots r_n \in \{0,1\}^n} \beta_r |r_1, \ldots, r_n\rangle$, we have*

- *In case $\delta = 0$, $\mathcal{V}$ obtains $\hat{r} \in \{0,1\}^n$ as a result of a standard measurement of $|\psi\rangle$;*
- *In case $\delta = 1$, there is an efficient quantum algorithm* Rec *which takes $KT = \{(k_i, t_i)\}$ and $|\psi'\rangle$ as inputs, outputs a state $|\psi''\rangle$ within negligible trace distance from $|\psi\rangle$.*

**Proof.** As the protocol basically applies the same operations to each qubit of $|\psi\rangle$ independently. It suffices to consider the first qubit of $|\psi\rangle$. Without loss of generality, we can rewrite $|\psi\rangle = \sum_{r_1\in\{0,1\}} \hat{\beta}_{r_1} |r_1\rangle |\psi_{r_1}\rangle$. Applying the $\mathsf{SAMP}_{\mathcal{F}} = \mathsf{SAMP}_{\mathcal{G}}$ procedure in superposition with $k_1$ and the first qubit containing $r_1$ as input will lead to a state either

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{\substack{r_1 \in \{0,1\},\\ x\in\mathcal{X}, y\in\mathcal{Y}}} \hat{\beta}_{r_1} \sqrt{g_{k_1,r_1}(x_1)(y_1)}\, |r_1\rangle |\psi_{r_1}\rangle |x_1\rangle |y_1\rangle$$

for $\delta = 0$, or

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{\substack{r_1 \in \{0,1\},\\ x\in\mathcal{X}, y\in\mathcal{Y}}} \hat{\beta}_{r_1} \sqrt{f'_{k_1,r_1}(x_1)(y_1)}\, |r_1\rangle |\psi_{r_1}\rangle |x_1\rangle |y_1\rangle ,$$

which then is within negligible trace distance of the following state:

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{\substack{r_1 \in \{0,1\},\\ x\in\mathcal{X}, y\in\mathcal{Y}}} \hat{\beta}_{r_1} \sqrt{f_{k_1,r_1}(x_1)(y_1)}\, |r_1\rangle |\psi_{r_1}\rangle |x_1\rangle |y_1\rangle ,$$

because $E_{x\xleftarrow{\$}\mathcal{X}}[H^2(f_{k,b}(x), f'_{k,b}(x))] \leq \mu(\lambda)$ for some negligible function $\mu(\cdot)$ by the assumption. Measuring the register containing $y_1$ will obtain $\hat{y}_1 \in \mathcal{Y}$. Let $(\hat{r}_1, x_{\hat{r}_1,\hat{y}_1}) = \mathsf{INV}_{\mathcal{G}}(t_1, \hat{y}_1)$ if $\delta = 0$, otherwise $\hat{x}_{r_1,\hat{y}_1} = \mathsf{INV}_{\mathcal{F}}(t_1, r_1, \hat{y}_1)$ for $r_1 \in \{0,1\}$. If $\delta = 0$, with probability $(\hat{\beta}_{\hat{r}_1})^2$ the remaining state $|\psi'\rangle$ held by $\mathcal{P}$ is

$$|\hat{r}_1\rangle |\phi_{\hat{r}_1}\rangle |x_{\hat{r}_1,\hat{y}_1}\rangle |\hat{y}_1\rangle$$

by the disjoint trapdoor injective pair property of $\mathcal{G}$. Otherwise, the remaining state $|\psi'\rangle$ is within negligible trace distance of the following state:

$$\sum_{r_1\in\{0,1\}} \hat{\beta}_{r_1} |r_1\rangle |\phi_{r_1}\rangle |\hat{x}_{r_1,\hat{y}_1}\rangle |\hat{y}_1\rangle$$

by the trapdoor and injective pair property of $\mathcal{F}$. Clearly, if $\delta = 0$, $\mathcal{V}$ will obtain $\hat{r}_1$ with probability $(\hat{\beta}_{\hat{r}_1})^2$, which is the same as directly measuring the first qubit of $|\psi\rangle = \sum_{r_1\in\{0,1\}} \hat{\beta}_{r_1} |r_1\rangle |\psi_{r_1}\rangle$. Otherwise, let $\mathsf{Rec}$ be the quantum algorithm which computes the inverting algorithm $\hat{x}_{r_1,\hat{y}_1} = \mathsf{INV}_{\mathcal{F}}(t_1, r_1, \hat{y}_1)$. Clearly, given $t_1$ and $|\psi'\rangle$ as inputs, $\mathsf{Rec}$ can be used to uncompute the register containing $\hat{x}_{r_1,\hat{y}_1}$ of the above state, and obtain a state within negligible trace distance from the input state $|\psi\rangle$. This completes the proof. □

**Theorem 4.4** *If $\mathcal{F}$ is an extended NTCF family, and $\mathcal{G}$ is the corresponding trapdoor injective family $\mathcal{G}$, then for a uniformly random $\delta \in \{0,1\}$ held by $\mathcal{V}$ and any QPT algorithm $\mathcal{P}$, the probability that $\mathcal{P}$ outputs $\delta' = \delta$ is negligibly close to* $1/2$ *after interacting with $\mathcal{V}$.*

**Proof.** Consider a modification of Protocol 3 where $\mathcal{V}$ also generates the keys by running $(k_i, t_i) = \mathsf{GEN}_{\mathcal{F}}(1^\lambda)$ for $\delta = 0$. Then, $\mathcal{P}$ cannot obtain any information of $\delta$ by interacting with $\mathcal{V}$ in the modified protocol. Thus, the probability that $\mathcal{P}$ output $\delta' = \delta$ is exactly $1/2$. By the fact that $\mathsf{SAMP}_{\mathcal{F}} = \mathsf{SAMP}_{\mathcal{G}}$, and that no QPT algorithm $\mathcal{P}$ can distinguish the keys generated by $\mathsf{GEN}_{\mathcal{F}}(1^\lambda)$ from that generated by $\mathsf{GEN}_{\mathcal{G}}(1^\lambda)$, we have the view of $\mathcal{P}$ in the modified protocol and that in Protocol 3 are computationally indistinguishable. This means that the probability that $\mathcal{P}$ outputs $\delta' = \delta$ is negligibly close to $1/2$, which completes the proof. □

## 4.3 The Interactive Proof System with a Fully Classical Verifier

Let $\lambda$ be the security parameter. Let $n = \mathsf{poly}(\lambda)$. Let $\mathcal{F}$ be an extended NTCF family associated with a corresponding trapdoor injective family $\mathcal{G}$. Now, we are ready to present the proof system with a classical verifier $\mathcal{V}^{\mathcal{R}_F}$ and a quantum prover $\mathcal{P}^{\mathcal{D}}$ such that for a YES instance, the probability that $\mathcal{V}^{\mathcal{R}_F}$ outputs 1, after interacting with an honest prover $\mathcal{P}^{\mathcal{D}}$, is at least $\frac{1+a}{2} - \mathsf{negl}(n)$, while for a NO instance, the probability that $\mathcal{V}^{\mathcal{R}_F}$ outputs 1, after interacting with any QPT algorithm $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$, is at most $\frac{1+b}{2} + \mathsf{negl}(n)$. The full description of the proof system is given in Protocol 4. We have the following theorems for Protocol 4.

**Theorem 4.5** *If $\mathcal{F}$ is an extended NTCF family, and $\mathcal{G}$ is the corresponding trapdoor injective family. Then, for the promise problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$, in an honest execution of Protocol 4 between a classical verifier $\mathcal{V}^{\mathcal{R}_F}$ and a quantum prover $\mathcal{P}^{\mathcal{D}}$, the probability that $\mathcal{V}^{\mathcal{R}_F}$ accepts a YES instance is at least $\frac{1+a}{2} - \mathsf{negl}(n)$.*

**Proof.** Note that before executing Protocol 3, $\mathcal{P}$ will obtain a state $|\phi\rangle$ within negligible trace distance from the following state

$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\substack{r = r_1 \ldots r_n \in R_{s,\hat{Y}}, \\ \hat{X}_{r,\hat{Y}} = \hat{x}_{r_1,\hat{y}_1} \ldots \hat{x}_{r_n,\hat{y}_n} \in \mathcal{X}^n}} \mathcal{D}(|r\rangle |0^m\rangle) \left|\hat{X}_{r,\hat{Y}}\right\rangle .$$

If $\delta = 0$, by Theorem 4.3 measuring the above state will obtain a pair $(\hat{r}, F(\hat{r}))$ satisfying $\hat{r} \in R_{s,\hat{Y}}$ with probability at least $a$, which means that $\mathcal{V}$ will set *accept* $= 1$ with probability at least $a - \mathsf{negl}(n)$. If $\delta = 1$, by Theorem 4.3 we have that $\mathcal{P}$ can compute a state $|\phi'\rangle$ within negligible trace distance from the following state

$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\substack{r = r_1 \ldots r_n \in R_{s,\hat{Y}}, \\ \hat{X}_{r,\hat{Y}} = \hat{x}_{r_1,\hat{y}_1} \ldots \hat{x}_{r_n,\hat{y}_n} \in \mathcal{X}^n}} |r\rangle \left|\hat{X}_{r,\hat{Y}}\right\rangle ,$$

which is within negligible trace distance from the state $|\phi\rangle$ obtained by $\mathcal{P}$ after executing the generation phase of Protocol 2. As $\mathcal{V}$ and $\mathcal{P}$ will execute the verification phase of Protocol 2 with $|\phi'\rangle$ as input, the probability that *accept* $= 1$ in step 5 is negligibly close to 1 by Theorem 4.1.

Since $\delta \in \{0, 1\}$ is uniformly chosen at random, the probability that $\mathcal{V}^{\mathcal{R}_F}$ accepts a YES instance is at least $\frac{1+a}{2} - \mathsf{negl}(n)$. This completes the proof. □

**Theorem 4.6** *If $\mathcal{F}$ is an extended NTCF family, and $\mathcal{G}$ is the corresponding trapdoor injective family. Then, for the promise problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_F, a, b)$, any classical device $\mathcal{O}_F$ computing $F$, and any QPT algorithm $\widetilde{\mathcal{P}}^{\mathcal{D},\mathcal{O}_F}$, the probability that $\mathcal{V}^{\mathcal{R}_F}$ accepts a NO instance is at most $\frac{1+b}{2} + \mathsf{negl}(n)$.*

**Proof.** Let $E$ be the event that $\widetilde{\mathcal{P}}$ makes a classical query with some $\hat{r} \in R_{s,\hat{Y}}$ to $\mathcal{O}_F$ at the end of executing Protocol 4 in step 3. By Theorem 4.4, we have that $|\Pr[E|\delta = 0] - \Pr[E|\delta = 1]| \leq \mathsf{negl}(n)$ because $\mathcal{P}$ cannot obtain any useful information about $\delta$ held by $\mathcal{V}$.

Let $\mathrm{acc}_\delta$ be the event that $\mathcal{V}$ outputs 1 for a fixed $\delta \in \{0, 1\}$. As $\delta$ is uniformly chosen by $\mathcal{V}$, the probability that $\mathcal{V}$ outputs 1 is equal to $\frac{\Pr[\mathrm{acc}_0] + \Pr[\mathrm{acc}_1]}{2}$. By the assumption that $b$ is not smaller than the probability for any QPT algorithm to correctly guess $F(x)$ without knowing $x$, we have that $\Pr[\mathrm{acc}_0] \leq \Pr[E|\delta = 0] + b$. Moreover, by Theorem 4.2 we have that $\Pr[E|\delta = 1, \mathrm{acc}_1] = \mathsf{negl}(n)$, as otherwise there is a QPT algorithm which breaks the security of Protocol 2 by internally running Protocol 3 before the verification phase of Protocol 2. By the law of total probability, we have that

$$\Pr[E|\delta = 1] = \Pr[E|\delta = 1, \mathrm{acc}_1] \Pr[\mathrm{acc}_1] + \Pr[E|\delta = 1, \neg\, \mathrm{acc}_1] \Pr[\neg\, \mathrm{acc}_1].$$

**Protocol 4:** Interactive Proof with a Fully Classical Verifier

**Inputs:** The classical verifier $\mathcal{V}$ has classical access to $\mathcal{R}_F$. The quantum prover $\mathcal{P}$ has quantum access to $\mathcal{D}$.

**Description:** This is the full description of protocol.

1. $\mathcal{V}$ executes the generation phase of Protocol 2 with $\mathcal{P}$. After this, $\mathcal{V}$ will hold a string $s \in \{0,1\}^n$, a set $KT = \{(k_1, t_i)\}$ of keys and trapdoors, and a set $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_n\}$. Let $\hat{x}_{r_i,\hat{y}_i} = \mathsf{INV}_{\mathcal{F}}(t_i, r_i, \hat{y}_i)$ for any $r_i \in \{0,1\}$ if $s_i = 1$, otherwise $(\hat{r}_i, \hat{x}_{\hat{r}_i,\hat{y}_i}) = \mathsf{INV}_{\mathcal{G}}(t_i, \hat{y}_i)$. Then, $\mathcal{P}$ will hold a state $|\phi\rangle$ within negligible trace distance from the following state:
$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\substack{r = r_1 \ldots r_n \in R_{s,\hat{Y}}, \\ \hat{X}_{r,\hat{Y}} = \hat{x}_{r_1,\hat{y}_1} \ldots \hat{x}_{r_n,\hat{y}_n} \in \mathcal{X}^n}} |r\rangle \left|\hat{X}_{r,\hat{Y}}\right\rangle .$$
where $R_{s,\hat{Y}} = \{r = r_1 \ldots r_n \in \{0,1\}^n : s_i = 0 \wedge r_i = \hat{r}_i\}$.
2. By inserting a register containing $|0^m\rangle$ after the register containing $r$, $\mathcal{P}$ makes a quantum query with the register containing the first $n+m$ qubits to the device $\mathcal{D}$, which will return a state $|\psi\rangle$ within negligible trace distance from the following state:
$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\substack{r = r_1 \ldots r_n \in R_{s,\hat{Y}}, \\ \hat{X}_{r,\hat{Y}} = \hat{x}_{r_1,\hat{y}_1} \ldots \hat{x}_{r_n,\hat{y}_n} \in \mathcal{X}^n}} \mathcal{D}(|r\rangle |0^m\rangle) \left|\hat{X}_{r,\hat{Y}}\right\rangle .$$
3. $\mathcal{V}$ randomly chooses a bit $\delta \in \{0,1\}$, and executes Protocol 3 with $\mathcal{P}$ to measure the register containing the first $(n+m)$ qubits of $|\psi\rangle$. In the case $\delta = 0$, $\mathcal{V}$ will obtain a pair $(\hat{r}, \hat{h}) \in \{0,1\}^n \times \{0,1\}^m$. It sets *accept* $= 0$ and aborts if $\hat{r} \notin R_{s,\hat{Y}}$ or $\mathcal{R}_F(\hat{r}, \hat{h}) = 0$. Otherwise, it sets *accept* $= 1$ and aborts. In the case $\delta = 1$, $\mathcal{V}$ will obtain a set $KT' = \{(k'_i, t'_i)\}_{i \in \{1,\ldots,n+m\}}$ of keys and trapdoors; $\mathcal{P}$ will obtain a state $|\psi'\rangle$. $\mathcal{V}$ sends $KT' = \{(k'_i, t'_i)\}_{i \in \{1,\ldots,n+m\}}$ to $\mathcal{P}$.
4. $\mathcal{P}$ computes a state $|\psi''\rangle$ within negligible trace distance from the following state
$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\substack{r = r_1 \ldots r_n \in R_{s,\hat{Y}}, \\ \hat{X}_{r,\hat{Y}} = \hat{x}_{r_1,\hat{y}_1} \ldots \hat{x}_{r_n,\hat{y}_n} \in \mathcal{X}^n}} \mathcal{D}(|r\rangle |0^m\rangle) \left|\hat{X}_{r,\hat{Y}}\right\rangle ,$$
by using the algorithm $\mathsf{Rec}$ in Theorem 4.3 with $KT'$ and $|\psi'\rangle$ as inputs. Then, it makes a quantum query with the register containing the first $(n+m)$ qubits to $\mathcal{D}$. By omitting the middle $m$ qubits after the register containing $r$, this will lead to a state $|\phi'\rangle$ within negligible trace distance from the following state
$$\frac{1}{\sqrt{2^{\mathsf{hw}(s)}}} \sum_{\substack{r = r_1 \ldots r_n \in R_{s,\hat{Y}}, \\ \hat{X}_{r,\hat{Y}} = \hat{x}_{r_1,\hat{y}_1} \ldots \hat{x}_{r_n,\hat{y}_n} \in \mathcal{X}^n}} |r\rangle \left|\hat{X}_{r,\hat{Y}}\right\rangle .$$
5. $\mathcal{V}$ executes the verification phase of Protocol 2 with $\mathcal{P}$ using the state $|\phi'\rangle$ as input. Set *accept* $= 1$ if $\mathcal{V}$ output 1 in this execution.

**Output:** $\mathcal{V}$ outputs *accept*.

As $\Pr[\mathrm{acc}_1] \leq 1$, we have that $\Pr[E|\delta = 1] = \Pr[E|\delta = 1, \neg\,\mathrm{acc}_1] \Pr[\neg\,\mathrm{acc}_1] + \mathsf{negl}(n)$. This means that $\Pr[E|\delta = 0] = \Pr[E|\delta = 1, \neg\,\mathrm{acc}_1] \Pr[\neg\,\mathrm{acc}_1] + \mathsf{negl}(n)$. By the following inequality,
$$\Pr[\mathrm{acc}_0] + \Pr[\mathrm{acc}_1] \leq b + \Pr[E|\delta = 1, \neg\,\mathrm{acc}_1] \Pr[\neg\,\mathrm{acc}_1] + \Pr[\mathrm{acc}_1] + \mathsf{negl}(n) \leq 1 + b + \mathsf{negl}(n),$$

the probability that $\mathcal{V}$ accepts a NO instance is at most $\frac{\Pr[\mathrm{acc}_0]+\Pr[\mathrm{acc}_1]}{2} \leq \frac{1+b}{2} + \mathsf{negl}(n)$. This completes the proof. □

# 5 Application: Separation between ROM and QROM

In this section, we show that the problem of distinguishing the random oracle model (ROM) and the quantum ROM (QROM) is a natural $\mathsf{QBBC}$ problem.

In the ROM, all parties, including the adversary, are given access to an "idealized" random function (i.e., a random oracle, RO). Since its introduction [BR93], the ROM has been widely used to design and analyze many well-known schemes such as the OAEP encryption [BR95] and the full-domain hash (FDH) signature [BR96]. Although most "honestly-designed" ROM schemes seem to keep the security in practice, the soundness of ROM has been questioned by the literatures [CGH04, Nie02, MRH04]. The first separation between the ROM and the standard model was given by Canetti, Goldreich and Halevi [CGH04], 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.

In 2011, the authors of [BDF$^+$11] found that the classical ROM may even be problematic for quantum adversaries, and introduced the quantum ROM (QROM) where honest parties (e.g., the cryptosystems) still access the RO in a classical way, but the adversary is explicitly allowed to make quantum queries to the RO. They justified the necessity of QROM by presenting an artificial identification protocol which is secure in the ROM but is insecure in the QROM. However, the separation in [BDF$^+$11] heavily relies on a set of "timing assumptions". This is because their artificial identification protocol basically uses a (somewhat) trivial distinguisher between ROM and QROM. Specifically, the distinguisher is built upon the gap in finding a collision of an $m$-bit output hash function between using the birthday attack with $O(2^{m/2})$ classical queries and using the Grover algorithm with $O(2^{m/3})$ quantum queries [Gro96, BHT98]. Since the query gap is only polynomial (as $2^{m/2}$ can be naturally written as a polynomial of $2^{m/3}$), their argument requires extra "timing assumptions" (e.g., "unit time" and "zero time" assumptions) to ensure that the running time of the identification protocol is longer than $O(2^{m/3})$ "unit time" for a QROM adversary to run the Grover algorithm [Gro96], but is shorter than $O(2^{m/2})$ "unit time" for a ROM adversary to carry out the birthday attack. *This leaves a nine-year open question of finding a standard separation between ROM and QROM.*

Note that the only difference between ROM and QROM is that the adversary in the QROM can make quantum queries to the RO while that in the ROM can only make classical queries. This can be naturally seen as that the adversary in the QROM has a quantum device $\mathcal{D}$ correctly computing an idealized random function $\mathcal{O} : \{0,1\}^n \to \{0,1\}^m$, while that in the ROM only has a trivial quantum device $\mathcal{D}$ simply guessing the output of $\mathcal{O}(\cdot)$ at each point. As the honest parties are given classically access to the RO $\mathcal{O}(\cdot)$ (and thus has a natural device $\mathcal{R}_{\mathcal{O}}$ deciding the input-output relation of $\mathcal{O}(\cdot)$), the problem of distinguishing ROM and QROM is a natural promise problem $\mathsf{QBBC}(\mathcal{D}, \mathcal{R}_{\mathcal{O}}, 1, \frac{1}{2^m})$.

By Theorems 4.5 and 4.6, we immediately have that there is an efficient classical distinguisher $\mathcal{V}'$ (obtained by repeatedly running the verifier $\mathcal{V}$ in Protocol 4 a polynomial number of times) for ROM and QROM, such that it almost always outputs 1 after interacting with a QROM adversary performing the strategy of an honest prover $\mathcal{P}$, and 0 after interacting with any adversary in the ROM. This distinguisher $\mathcal{V}'$ can be used as a building block to construct cryptosystems that are secure in the ROM but insecure in the QROM, as it allows to embed some malicious behaviors that can only be utilized by an adversary in the QROM: given a secure cryptosystem $\mathcal{C}$, one can construct another cryptosystem $\mathcal{C}'$ which first internally runs the distinguisher $\mathcal{V}'$ to detect if the adversary runs in the QROM, and then performs normally as $\mathcal{C}$ does if $\mathcal{V}'$ outputs 0, otherwise behaves maliciously (e.g., directly outputting the secret key to the adversary) if $\mathcal{V}'$ outputs 1.

In the following, we give a concrete counter-example using identification schemes.

**Definition 5.1 (Identification Scheme)** *An identification scheme* $\Pi_{ID}$ *consists of three PPT algorithms* $(\mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ *such that:*

- *given a security parameter* $\lambda$ *as input, the key generation algorithm* $\mathsf{KeyGen}$ *outputs a public key* $\mathsf{pk}$ *and a secret key* $\mathsf{sk}$*, namely,* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*;*
- $\mathsf{Prove}$ *and* $\mathsf{Verify}$ *are interactive algorithms. After interacting with the prover algorithm* $\mathsf{Prove}(\mathsf{sk})$ *with input a secret key* $\mathsf{sk}$*, the verification algorithm* $\mathsf{Verify}(\mathsf{pk})$ *which takes a public key* $\mathsf{pk}$ *as input will output a bit* $b = 1/0$ *indicating whether "accept" or "reject".*

Denote by $\langle \mathsf{Prove}(\mathsf{sk}), \mathsf{Verify}(\mathsf{pk}) \rangle$ the output of $\mathsf{Verify}(\mathsf{pk})$. For correctness, we require that for all $\lambda$, and all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, the algorithm $\mathsf{Verify}(\mathsf{pk})$ will always output 1 (i.e., "accept") after interacting with $\mathsf{Prove}(\mathsf{sk})$, namely,

$$\Pr\Big[\langle \mathsf{Prove}(\mathsf{sk}), \mathsf{Verify}(\mathsf{pk}) \rangle = 1 : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)\Big] = 1,$$

where the probability is taken over all randomness used by algorithms $\mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify}$.

An active adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against identification schemes consists of a pair of interactive algorithms, which works in two stages. In the first stage, the adversary runs algorithm $\mathcal{A}_1(\mathsf{pk})$ to interact with an honest prover $\mathsf{Prove}(\mathsf{sk})$ by acting as a verifier, and outputs some "secret" information $\tau$ learned from the interactions. In the second stage, the adversary runs algorithm $\mathcal{A}_2(\tau)$ to interact with an honest verifier $\mathsf{Verify}(\mathsf{pk})$ by acting as a prover, and tries to impersonate $\mathsf{Prove}(\mathsf{sk})$ to a verifier. The security of identification schemes requires that for any efficient algorithm $\mathcal{A}$, it is unable to falsely impersonate $\mathsf{Prove}(\mathsf{sk})$ to a verifier. For our purpose, we focus on post-quantum identification schemes, where the adversary $\mathcal{A}$ can be any QPT algorithm, but is only allowed to classically interact with the schemes (i.e., the interfaces of the schemes are still classical).

**Definition 5.2 (Active Security)** *An identification scheme* $\Pi_{ID} = (\mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ *is actively secure, if the following is negligible for all QPT adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*:*

$$\Pr\Big[\langle \mathcal{A}_2(\tau), \mathsf{Verify}(\mathsf{pk}) \rangle = 1 : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda), \tau \leftarrow \mathcal{A}_1^{\langle \mathsf{Prove}(\mathsf{sk}), \cdot \rangle}(\mathsf{pk})\Big],$$

*where the oracle* $\langle \mathsf{Prove}(\mathsf{sk}), \cdot \rangle$ *allows* $\mathcal{A}_1$ *to interact with an honest prover* $\mathsf{Prove}(\mathsf{sk})$ *by acting as a verifier.*

Let $\mathcal{O} : \{0,1\}^n \to \{0,1\}^m$ be any RO. Let $\mathcal{R}_\mathcal{O}$ be a device deciding the input-output relation of $\mathcal{O}(\cdot)$, which can be implemented by using a single classical query to $\mathcal{O}(\cdot)$ for each input. In particular, given an input pair $(x, y) \in \{0,1\}^n \times \{0,1\}^m$, the device $\mathcal{R}_\mathcal{O}$ first sends a query $x$ to the RO $\mathcal{O}(\cdot)$, which will return $\hat{y} = \mathcal{O}(x)$. If $y = \hat{y}$, $\mathcal{R}_\mathcal{O}(x, y)$ returns 1, and 0 otherwise. Let $\mathcal{V}^{\mathcal{R}_\mathcal{O}}$ be the verifier with a device $\mathcal{R}_\mathcal{O}$ in Protocol 4. Let $\widetilde{\mathcal{P}}$ be a dummy prover for $\mathcal{V}^{\mathcal{R}_\mathcal{O}}$ which always returns $0^\ell$ if $\mathcal{V}^{\mathcal{R}_\mathcal{O}}$ is expected to receive a bit string of length $\ell$. Let $\widetilde{\mathcal{V}}^{\mathcal{R}_\mathcal{O}}$ be the algorithm obtained by repeatedly running $\mathcal{V}^{\mathcal{R}_\mathcal{O}}$ for $N = \mathsf{poly}(n)$ times, and outputting 1 if in more than $\frac{7N}{8}$ of the repetitions $\mathcal{V}^{\mathcal{R}_\mathcal{O}}$ outputs 1. Let $\widetilde{\mathcal{P}}'$ be the corresponding dummy algorithm for $\widetilde{\mathcal{V}}$, obtained by repeatedly running $\widetilde{\mathcal{P}}$ for $N = \mathsf{poly}(n)$ times.

We now construct an identification scheme $\Pi'_{ID} = (\mathsf{KeyGen}', \mathsf{Prove}', \mathsf{Verify}')$ from any actively secure $\Pi_{ID} = (\mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ and algorithms $\widetilde{\mathcal{V}}^{\mathcal{R}_\mathcal{O}}$ and $\widetilde{\mathcal{P}}'$.

- The key generation algorithm $\mathsf{KeyGen}'$ works the same as $\mathsf{KeyGen}$, namely, it simply runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$, and outputs $(\mathsf{pk}, \mathsf{sk})$;

- The algorithm $\mathsf{Prove}'(\mathsf{sk})$ first internally executes $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$. If $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ outputs 1, it sends $\mathsf{sk}$ to the verifier and aborts, otherwise it runs $\mathsf{Prove}(\mathsf{sk})$ normally.

- The algorithm $\mathsf{Verify}'(\mathsf{pk})$ first internally executes $\widetilde{\mathcal{P}}'$. It outputs 1 if it receives a secret key $\mathsf{sk}$ corresponding to $\mathsf{pk}$ after this execution. Otherwise, it runs $\mathsf{Verify}(\mathsf{pk})$ normally, and outputs 1 if and only if $\mathsf{Verify}(\mathsf{pk})$ accepts.

Clearly, the identification scheme $\Pi'_{ID} = (\mathsf{KeyGen}', \mathsf{Prove}', \mathsf{Verify}')$ is correct by the correctness of $\Pi_{ID} = (\mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$, because for any honest prover $\mathsf{Prove}'(\mathsf{sk})$, the algorithm $\mathsf{Verify}'(\mathsf{pk})$ will always output 1 no matter if the internal subroutine $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ of $\mathsf{Prove}'(\mathsf{sk})$ outputs 1 or not. We now prove that $\Pi'_{ID} = (\mathsf{KeyGen}', \mathsf{Prove}', \mathsf{Verify}')$ is secure in the ROM.

**Theorem 5.1** *Let $\mathcal{O} : \{0,1\}^n \to \{0,1\}^m$ be any RO for sufficiently large $n$. If $\Pi_{ID} = (\mathsf{KeyGen}, \mathsf{Prove}, \mathsf{Verify})$ is actively secure, then the modified scheme $\Pi'_{ID} = (\mathsf{KeyGen}', \mathsf{Prove}', \mathsf{Verify}')$ is actively secure in the ROM. In particular, if there exists an efficient adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ breaking the security of $\Pi'_{ID}$ with probability $\delta$, then there exists another efficient adversary of $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ breaking the security of $\Pi_{ID}$ with probability at least $\delta - \mathsf{negl}(n)$.*

**Proof.**

We now construct $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ from $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ as follows. Given a public key $\mathsf{pk}$ as input, algorithm $\mathcal{A}_1^{\langle \mathsf{Prove}(\mathsf{sk}), \cdot \rangle}(\mathsf{pk})$ internally runs $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ and $\mathcal{A}'_1(\mathsf{pk})$, and simulates a prover $\mathsf{Prove}'(\mathsf{sk})$ for $\mathcal{A}'_1(\mathsf{pk})$ as follows:

- If $\mathcal{A}'_1(\mathsf{pk})$ sends a message intended for the subroutine $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$, it sends the message to its simulated $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$. If $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ outputs 1, $\mathcal{A}_1$ directly aborts. Otherwise, it continues as an honest prover by using $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$.

- If $\mathcal{A}'_1(\mathsf{pk})$ sends a message intended for the subroutine $\mathsf{Prove}(\mathsf{sk})$, it sends the message to its own oracle $\langle \mathsf{Prove}(\mathsf{sk}), \cdot \rangle$, and returns whatever it obtains from the oracle to $\mathcal{A}'_1(\mathsf{pk})$.

- If $\mathcal{A}'_1(\mathsf{pk})$ produces an output $\tau$ and aborts, $\mathcal{A}_1$ returns $\tau$ as its own output and aborts.

Given $\tau$ as input, algorithm $\mathcal{A}_2(\tau)$ internally runs $\widetilde{\mathcal{P}}'$ and $\mathcal{A}'_2(\tau)$ to interact with an honest verifier $\mathsf{Verify}(\mathsf{pk})$ by simulating as a verifier $\mathsf{Verify}'(\mathsf{pk})$ for $\mathcal{A}'_2(\tau)$ as follows:

- If $\mathcal{A}'_2$ sends a message intended for the subroutine $\widetilde{\mathcal{P}}'$, $\mathcal{A}_2$ sends the message to the simulated $\widetilde{\mathcal{P}}'$, and returns whatever it obtains from $\widetilde{\mathcal{P}}'$ to $\mathcal{A}'_2$.

- If $\mathcal{A}'_2$ sends a secret key $\mathsf{sk}$ corresponding to $\mathsf{pk}$ and aborts, $\mathcal{A}_2$ directly runs $\mathsf{Prove}(\mathsf{sk})$ to interact with $\mathsf{Verify}(\mathsf{pk})$;

- If $\mathcal{A}'_2$ sends a message intended for the subroutine $\mathsf{Verify}(\mathsf{pk})$, $\mathcal{A}_2$ sends the message to $\mathsf{Verify}(\mathsf{pk})$ and returns whatever it obtains from $\mathsf{Verify}(\mathsf{pk})$ to $\mathcal{A}'_2$.

Let $E$ be the event that $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ outputs 1. Clearly, if $E$ does not happen, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ simulates a perfect environment for $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$. In this case, the probability that $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ breaks the security of $\Pi_{ID}$ is the same as that $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ breaks the security of $\Pi'_{ID}$. Thus, it suffices to show that $\Pr[E] = \mathsf{negl}(n)$.

Note that the adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ can only classically access the RO $\mathcal{O}(\cdot)$, and that $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ is obtained by repeatedly running $\mathcal{V}^{\mathcal{R}_{\mathcal{O}}}$ for $N = \mathsf{poly}(n)$ times. By Theorem 4.6, the probability that $\mathcal{V}^{\mathcal{R}_{\mathcal{O}}}$ outputs 1 after interacting with any adversary $\mathcal{A}'_1$ given only classical access to $\mathcal{O}(\cdot)$ is at most $\frac{1+1/2^m}{2} + \mathsf{negl}(n) \le \frac{3}{4} + \mathsf{negl}(n)$. As $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ will only output 1 if in more than $\frac{7N}{8}$ of the repetitions $\mathcal{V}^{\mathcal{R}_{\mathcal{O}}}$ outputs 1, we have that $\Pr[E] \le e^{-N/150} = \mathsf{negl}(n)$ for sufficiently large $n$ and $N$ by the Chernoff bound. This completes the proof. □

**Theorem 5.2** *Let $\mathcal{O} : \{0,1\}^n \to \{0,1\}^m$ be any RO for any integers $n$ and $m$. The modified scheme $\Pi'_{ID} = (\mathsf{KeyGen}', \mathsf{Prove}', \mathsf{Verify}')$ is insecure in the QROM.*

**Proof.** Let $\mathcal{D}$ be a quantum device computing $\mathcal{O}$, i.e., $\mathcal{D} : |x\rangle |0^m\rangle \to |x\rangle |\mathcal{O}(x)\rangle$. Let $\widetilde{\mathcal{P}}^{\mathcal{D}}$ is obtained by repeatedly running the prover algorithm $\mathcal{P}^{\mathcal{D}}$ in Protocol 4 for $N = \mathsf{poly}(n)$ times. By Theorem 4.5, the probability that $\mathcal{V}^{\mathcal{R}_{\mathcal{O}}}$ outputs 1 after interacting with $\mathcal{P}^{\mathcal{D}}$ is at least $1 - \mathsf{negl}(n)$. Thus, the probability that $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ outputs 1 after interacting with $\widetilde{\mathcal{P}}^{\mathcal{D}}$ is at least $1 - e^{-N/150} = 1 - \mathsf{negl}(n)$ for sufficiently large $n$ and $N$ by the Chernoff bound.

We now construct an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ breaking the active security of $\Pi'_{ID}$ in the QROM. Note that $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is given quantum access to $\mathcal{O}(\cdot)$. This means that $\mathcal{A}_1$ can internally runs $\widetilde{\mathcal{P}}^{\mathcal{D}}$ by answering each query to $\mathcal{D}$ using a quantum query to $\mathcal{O}(\cdot)$. $\mathcal{A}_1$ works as follows to obtain a secret key $\mathsf{sk}$ from $\mathsf{Prove}'(\mathsf{sk})$:

- If $\mathsf{Prove}'(\mathsf{sk})$ sends a message intended for the subroutine $\widetilde{\mathcal{P}}'$, $\mathcal{A}_1$ sends the message to $\widetilde{\mathcal{P}}^{\mathcal{D}}$, and returns whatever it obtains from $\widetilde{\mathcal{P}}^{\mathcal{D}}$ to $\mathsf{Prove}'(\mathsf{sk})$;
- If $\mathsf{Prove}'(\mathsf{sk})$ sends a secret key $\mathsf{sk}$ corresponding to $\mathsf{pk}$ and aborts, $\mathcal{A}_1$ directly outputs $\tau = \mathsf{sk}$ and aborts;
- If $\mathsf{Prove}'(\mathsf{sk})$ sends a message intended for the subroutine $\mathsf{Verify}(\mathsf{pk})$, $\mathcal{A}_1$ outputs $\tau = \perp$ and aborts.

Given $\tau = \mathsf{sk}$ as input, algorithm $\mathcal{A}_2(\tau)$ performs the same as $\mathsf{Prove}'(\mathsf{sk})$ to interact with $\mathsf{Verify}(\mathsf{pk})$. Otherwise, $\mathcal{A}_2(\tau)$ directly aborts.

It suffices to show that

$$\Pr\Big[\langle \mathcal{A}_2(\tau), \mathsf{Verify}'(\mathsf{pk})\rangle = 1 : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}'(1^\lambda), \tau \leftarrow \mathcal{A}_1^{\langle \mathsf{Prove}'(\mathsf{sk}), \cdot\rangle}(\mathsf{pk})\Big] \geq 1 - \mathsf{negl}(n),$$

which in turn can be proved by showing that $\mathcal{A}_1$ will output $\tau = \mathsf{sk}$ with probability at least $1 - \mathsf{negl}(n)$. As $\mathsf{Prove}'(\mathsf{sk})$ will output $\mathsf{sk}$ if $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$ outputs 1 and $\mathcal{A}_1$ essentially runs $\widetilde{\mathcal{P}}^{\mathcal{D}}$ to interact with $\widetilde{\mathcal{V}}^{\mathcal{R}_{\mathcal{O}}}$, the probability that $\mathsf{Prove}'(\mathsf{sk})$ sends a secret key $\mathsf{sk}$ to $\mathcal{A}_1$ is at least $1 - \mathsf{negl}(n)$ for sufficiently large $n$ and $N$. This means that $\mathcal{A}_1$ will output $\tau = \mathsf{sk}$ with probability at least $1 - \mathsf{negl}(n)$. By the correctness of $\Pi'_{ID}$, we complete the proof. □

# References

[ABOE10] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In *Innovations in Comuter Science - ICS 2010, Tsinghua University , Beijing, China, January 5-7, 2010. Proceedings*, pages 453–469, 2010.

[ABOEM17] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv preprint arXiv:1704.04487*, 2017.

[ACGH19] Gorjan Alagic, Andrew M Childs, Alex B Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. *arXiv*, pages arXiv–1911, 2019.

[AV13] Dorit Aharonov and Umesh Vazirani. *Is quantum mechanics falsifiable? A computational perspective on the foundations of quantum mechanics.* Computability: Turing, Gödel, Church, and Beyond. MIT Press, 2013.

[BBB+06] Eli Biham, Michel Boyer, P. Oscar Boykin, Tal Mor, and Vwani Roychowdhury. A proof of the security of quantum key distribution. *Journal of Cryptology*, 19(4):381–439, Oct 2006.

[BCM$^+$18] Z. Brakerski, P. Christiano, U. Mahadev, U. Vazirani, and T. Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 320–331, 2018.

[BDF$^+$11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In DongHoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, pages 41–69. Springer, Heidelberg, 2011.

[BFK09] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 517–526, 2009.

[BHT98] Gilles Brassard, Peter HØyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In Cláudio L. Lucchesi and Arnaldo V. Moura, editors, *LATIN'98: Theoretical Informatics*, pages 163–169. Springer, Heidelberg, 1998.

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low t-gate complexity. In *Annual Cryptology Conference*, pages 609–629. Springer, 2015.

[BL08] Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78:012352, Jul 2008.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS 1993*, pages 62–73. ACM, 1993.

[BR95] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT'94*, pages 92–111. Springer, Heidelberg, 1995.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures-how to sign with RSA and Rabin. In Ueli Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, pages 399–416. Springer, Heidelberg, 1996.

[Bra18] Zvika Brakerski. Quantum fhe (almost) as secure as classical. In *Annual International Cryptology Conference*, pages 67–95. Springer, 2018.

[Bro18] Anne Broadbent. How to verify a quantum computation. *Theory of Computing*, 14(11):1–37, 2018.

[CCKW18] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. On the possibility of classical client blind quantum computing. *arXiv preprint arXiv:1802.08759*, 2018.

[CCKW19] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. Qfactory: classically-instructed remote secret qubits preparation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 615–645. Springer, 2019.

[CCY20] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In *Theory of Cryptography*, 2020.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, July 2004.

[CGJV19] Andrea Coladangelo, Alex B. Grilo, Stacey Jeffery, and Thomas Vidick. Verifier-on-a-leash: New schemes for verifiable delegated quantum computation, with quasilinear resources. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 247–277, Cham, 2019. Springer International Publishing.

[Chi05] Andrew M. Childs. Secure assisted quantum computation. *Quantum Info. Comput.*, 5(6):456466, September 2005.

[CHSH69] John F Clauser, Michael A Horne, Abner Shimony, and Richard A Holt. Proposed experiment to test local hidden-variable theories. *Physical review letters*, 23(15):880, 1969.

[DSS16] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *Annual International Cryptology Conference*, pages 3–32. Springer, 2016.

[FHM18] Joseph F. Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Phys. Rev. Lett.*, 120:040501, Jan 2018.

[Fit17] Joseph F Fitzsimons. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3(1):1–11, 2017.

[FK17] Joseph F. Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Phys. Rev. A*, 96:012303, Jul 2017.

[Fv99] C. A. Fuchs and J. van de Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, May 1999.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[GKK19] Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of quantum computation: An overview of existing approaches. *Theory of computing systems*, 63(4):715–808, 2019.

[GKW15] Alexandru Gheorghiu, Elham Kashefi, and Petros Wallden. Robustness and device independence of verifiable blind quantum computing. *New Journal of Physics*, 17(8):083040, 2015.

[GMMR13] Vittorio Giovannetti, Lorenzo Maccone, Tomoyuki Morimae, and Terry G Rudolph. Efficient universal blind quantum computation. *Physical review letters*, 111(23):230501, 2013.

[Gri19] Alex B Grilo. A simple protocol for verifiable delegation of quantum computation in one round. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC 1996*, pages 212–219. ACM, 1996.

[GV19] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1024–1033. IEEE, 2019.

[GWK17] Alexandru Gheorghiu, Petros Wallden, and Elham Kashefi. Rigidity of quantum steering and one-sided device-independent verifiable quantum computation. *New Journal of Physics*, 19(2):023043, feb 2017.

[HKSE17] D Hangleiter, M Kliesch, M Schwarz, and J Eisert. Direct certification of a class of quantum simulations. *Quantum Science and Technology*, 2(1):015004, feb 2017.

[HM15] Masahito Hayashi and Tomoyuki Morimae. Verifiable measurement-only blind quantum computing with stabilizer testing. *Phys. Rev. Lett.*, 115:220502, Nov 2015.

[HPDF15] Michal Hajdušek, Carlos A Pérez-Delgado, and Joseph F Fitzsimons. Device-independent verifiable blind quantum computation. *arXiv preprint arXiv:1502.02563*, 2015.

[HT19] Masahito Hayashi and Yuki Takeuchi. Verifying commuting quantum computations via fidelity estimation of weighted graph states. *New Journal of Physics*, 21(9):093060, 2019.

[Ji16] Zhengfeng Ji. Classical verification of quantum proofs. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 885–898, 2016.

[KKR06] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[Mah18a] U. Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267, 2018.

[Mah18b] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 332–338. IEEE, 2018.

[McK16] Matthew McKague. Interactive proofs for BQP via self-tested graph states. *Theory of Computing*, 12(3):1–42, 2016.

[MF16] Tomoyuki Morimae and Joseph F Fitzsimons. Post hoc verification with a single prover. *arXiv preprint arXiv:1603.06046*, 2016.

[MPDF13] Atul Mantri, Carlos A Pérez-Delgado, and Joseph F Fitzsimons. Optimal blind quantum computation. *Physical review letters*, 111(23):230502, 2013.

[MRH04] Ueli Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *Theory of Cryptography*, pages 21–39. Springer, Heidelberg, 2004.

[MTH17] Tomoyuki Morimae, Yuki Takeuchi, and Masahito Hayashi. Verification of hypergraph states. *Physical Review A*, 96(6):062321, 2017.

[MV20] Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. *arXiv preprint arXiv:2001.09161*, 2020.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, pages 111–126. Springer, Heidelberg, 2002.

[NIS16] NIST. Post-quantum cryptography standardization, 2016. `http://csrc.nist.gov/groups/ST/post-quantum-crypto/submission-requirements/index.html`.

[NV16] Anand Natarajan and Thomas Vidick. Robust self-testing of many-qubit states. *arXiv preprint arXiv:1610.03574*, 2016.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93. ACM, 2005.

[RUV13] Ben W Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496(7446):456–460, 2013.

[TM18] Yuki Takeuchi and Tomoyuki Morimae. Verification of many-qubit states. *Physical Review X*, 8(2):021060, 2018.

[Vid20] Thomas Vidick. Verifying quantum computations at scale: A cryptographic leash on quantum devices. *Bulletin of the American Mathematical Society*, 57(1):39–76, 2020.

[YZ20a] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. Cryptology ePrint Archive, Report 2020/1270, 2020. `https://eprint.iacr.org/2020/1270`.

[YZ20b] Takashi Yamakawa and Mark Zhandry. A note on separating classical and quantum random oracles. Cryptology ePrint Archive, Report 2020/787, 2020. `https://eprint.iacr.org/2020/787`.

[ZH19a] Huangjun Zhu and Masahito Hayashi. Efficient verification of hypergraph states. *Phys. Rev. Applied*, 12:054047, Nov 2019.

[ZH19b] Huangjun Zhu and Masahito Hayashi. Efficient verification of pure quantum states in the adversarial scenario. *Physical Review Letters*, 123(26):260504, 2019.

[ZYF$^+$19] Jiang Zhang, Yu Yu, Dengguo Feng, Shuqin Fan, and Zhenfeng Zhang. On the (quantum) random oracle methodology: New separations and more. Cryptology ePrint Archive, Report 2019/1101, 2019. `https://eprint.iacr.org/2019/1101`.

[ZZ20] Huangjun Zhu and Haoyu Zhang. Efficient verification of quantum gates with local operations. *Physical Review A*, 101(4):042316, 2020.