Unifying Presampling via Concentration Bounds

Siyao Guo[1], Qian Li[2], Qipeng Liu[3] and Jiapeng Zhang[4]

[1] New York University Shanghai
[2] Shenzhen Institute of Computing Sciences
[3] Princeton University
[4] University of Southern California

Abstract. Auxiliary-input idealized models, such as the auxiliary-input random oracle model and the auxiliary-input random permutation model, play a critical role in assessing *non-uniform security* of symmetric-key and hash-function constructions. However, obtaining security bounds in these models are often much more challenging than in traditional idealized models.

The presampling technique, introduced by Unruh (CRYPTO' 07), generically reduces security proofs in the auxiliary-input models to a much simpler bit-fixing models. This technique has been further optimized by Coretti, Dodis, Guo, Steinberger (EUROCRYPT' 18), and generalized by Coretti, Dodis, Guo (CRYPTO' 18), resulting in powerful tools for proving non-uniform security bounds in various idealized models, including random oracle models (ROM), random permutation models (RPM), ideal cipher models (ICM) and generic group models (GGM).

We study the possibility of unifying and leveraging the presampling technique to the quantum world. To this end,

- We show that such leveraging will resolve a major open problem in quantum computing, which is closely related with the famous Aaronson-Ambainis conjecture (ITCS' 11).
- Faced with this barrier, we give a new but equivalent bit-fixing model and a simple proof of presampling techniques for arbitrary oracle distribution and access in the classical setting, including AI-ROM and AI-RPM. Our security loss matches the security loss of the best known presampling technique by Coretti et al. (EUROCRYPT' 18) for both indistinguishability and unpredictability applications. Our new proof unifies previous results by Coretti et al. (EUROCRYPT' 18) and Coretti et al. (CRYPTO' 18).
- We leverage our new classical presampling techniques to a novel "*quantum* bit-fixing version" of presampling. The security loss of our quantum bit-fixing presampling also matches the optimal security loss of the classical presampling. Using our techniques, we give the *first* post-quantum non-uniform security bounds for salted Merkle-Damgård hash functions.

1 Introduction

Practical symmetric-key and hash-function constructions are typically designed and analyzed in idealized models, such as random oracle model (ROM), random permutation model (RPM) and ideal-cipher model (ICM). Since most constructions of block ciphers and hash functions lack of solid theoretical foundations from the point of view of provable security, security bounds in idealized models provide an essential (heuristic) justification for their security. In particular, the *exact security bounds* obtained in such idealized models are often viewed as guidelines for both designers and cryptanalysis in terms of the best possible security level that can be achieved in the standard model.

Though idealized models capture all attacks that do not exploit the structure of a particular instantiation of the underlying primitive, they fail to capture preprocessing attacks. The obtained bounds in idealized model are inaccurate or not applicable at all once preprocessing is allowed. For example, Hellman [Hel80] showed a preprocessing attack that takes S bits of advice and makes T queries to a permutation $f : [N] \to [N]$ and inverts a random element of $[N]$ with probability roughly ST/N. Hence, a permutation cannot be one-way against attacks beyond $S = T = N^{1/2}$. However, it is easy to derive in RPM that a random permutation is invertible with probability at most T/N, suggesting security against attacks up to size N. Notice that the gap between N and $N^{1/2}$ matters for practical constructions. For example, while N suggests a 128-bit level security for 128-bit block cipher (e.g., 128-bit AES), $N^{1/2}$ only suggests 64-bit security.

Auxiliary-input models. To address the mismatch between idealized models and preprocessing attacks, auxiliary-input extensions of idealized models have been proposed, such as auxiliary-input random oracle model (AI-ROM), auxiliary-input random permutation model (AI-RPM) and auxiliary-input ideal cipher model (AI-ICM) [Unr07,DGK17,CDGS18,CDG18]. In AI models, an attacker is allowed to obtain arbitrary S bits of leakage about the idealized primitive before attacking the system, then use additional T queries to the primitive. Similar as that in the idealized models, security bounds obtained in AI models become the main source of justification and guidelines of the security level against *preprocessing* attacks (or more generally non-uniform attacks).

While AI models are simple extensions of well studied idealized models, they often do not offer simple and intuitive way to prove security bounds. Moreover, it becomes much more challenging to prove exact security bounds, which are demanded by practical relevance. For example, it is not straightforward how we should analyze inverting a random permutation $f : [N] \to [N]$ given S-bit advice (even for $S = 1$) and T queries in AI-RPM, let alone proving a ST/N bound, matching Hellman's attack.

The compression technique. For the specific question of permutation inversion, an optimal ST/N bound was first proved [DTT10] via the "compression paradigm", as introduced by Yao [Yao90], Gennaro and Trevisan [GT00]

(and later adopted by [Wee05]). The main idea is to argue if an attacker succeeds with "high probability" in inverting a random permutation, then we can use this attacker to build a shorter representation of (i.e., compress) the random permutation than what is possible from an information-theoretical point of view. Compression paradigm is a general technique which can be applied to different problems in auxiliary-input models. In fact, compression paradigm has been successfully applied to (auxiliary-input) random oracle models by Dodis et al. [DGK17], and (auxiliary-input) generic group model (GGM) by Corrigan-Gibbs and Kogan [CK18]. While compression based proofs often lead to optimal bounds, they are usually quite laborious. For every cryptographic construction to be analyzed, we need to carefully examine the property of the construction together with its security definition in order to compress the idealized primitive. Moreover, compression based proofs seem inapplicable to computationally secure applications. This limits the usage of this technique for analyzing more sophisticated constructions or practical constructions based on computational assumptions.

The presampling technique. A much simpler and intuitive proof for permutation inversion has been given by Coretti et al. [CDG18], by adapting the "presampling" approach taken by Coretti et al. [CDGS18] (and first introduced by [Unr07]) in the random oracle model. The presampling technique can be viewed as a general reduction from AI models to a much simpler bit-fixing model (BF) model, where the oracle can be arbitrarily fixed on some P coordinates chosen by the attacker, for some parameter P, but then the remaining coordinates are chosen at random and independently of the fixed coordinates. Importantly, the online attacker only knows the fixed coordinates. This makes BF models particularly easier to work with, because most proofs techniques for idealized models can be applied as long as we avoid the fixed coordinates.

Specifically, Coretti et al. [CDG18] and Coretti et al. [CDGS18] show that any attack with S-bit advice and T oracle queries in AI-ROM/RPM/ICM/GGM will have similar advantage in their corresponding P-BF models for an appropriately chosen P, up to an additive loss of $\delta(S,T,P) = ST/P$ (which is optimal shown by Dodis et al. [DGK17]). Moreover, for the special case of unpredictability applications (such as one-way function), one can set P to be roughly ST and achieve a multiplicative factor of 2 in the exact security.

This results in a general way for proving security in AI models. For a cryptographic application in an AI-model, we first analyze its security in the corresponding P-BF model and obtain security bounds $\varepsilon(S,T,P)$, then choose P to optimize $\delta(S,T,P) + \varepsilon(S,T,P)$. For an unpredictability application, its security in the AI model is just roughly $2 \cdot \varepsilon(S,T,ST)$, i.e., twice of its security in the (ST)-BF model. As an example, in the (ST)-BF-RPM, it can be easily shown that a random permutation $f : [N] \to [N]$ is invertible with probability at

most $O(ST/N)$ [5]. Therefore it immediately gives the optimal $O(ST/N)$ bound (matching Hellman's attack) in AI-RPM.

The presampling technique offers a much simpler approach for proving security bounds in AI models than the compression technique. By presampling techniques, Coretti et al. [CDG18] and Coretti et al. [CDG18] using simpler proofs recover the AI-ROM/RPM/GGM security bounds obtained by the compression technique [DTT10,DGK17,CK18], and give the first non-uniform bounds for a number of important practical applications (which compression appears intractable), such as (salted) Merkle-Damgård hash functions (MDHF). Chung et al. [CLMP13] study the effects of salting in the collision-resistant hash functions and argues that salting defeats preprocessing in this case.

We remark that *the optimal additive loss* and *multiplicative version* of presampling techniques in [CDG18,CDGS18] are *crucial* for obtaining exact bounds. As shown by Dodis et al. [DGK17], the presampling technique by Unruh [Unr07] with security loss $\sqrt{ST/P}$ yields sub-optimal bounds for many applications. Moreover, even with optimal additive loss, the indistinguishability version of presampling only yields $\sqrt{ST/N}$ security bounds (therefore useless for obtaining optimal bounds) for one-way functions.

A new challenge: quantum adversaries. Quantum algorithms can efficiently break many widely used assumptions for public key cryptography (such as factoring). Can they break practical symmetric-key and hash function constructions? How much security these constructions have to compromise for quantum adversaries? What if preprocessing is allowed?

To capture quantum adversaries, quantum extensions of idealized models have been considered such as the quantum random oracle model (QROM) [BDF+11], in which the attacker makes T superposition queries to the idealized primitive. Very recently, demanded by assessing post-quantum non-uniform security of symmetric-key and hash function, quantum versions of AI models have been proposed and studied [NABT15,HXY19,CLQ19,CGLQ20], in which the adversary is allowed to obtain S (qu)-bit precomputed advice about the idealized primitive.

By leveraging classical compression proofs, [NABT15,HXY19,CLQ19] obtain many security bounds. However, they only manage to analyze very basic applications such as one-way functions. Even for the basic question like inverting a random permutation with S-bit (classical) advice and T quantum queries, compression proofs give a sub-optimal bound ST^2/N. The success of presampling techniques in the classical setting motivates the main questions we study in this paper:

Can we leverage presampling techniques to the quantum setting?

[5] If the challenge $f(x)$ doesn't come from the fixed coordinates, then a proof by standard techniques bounds the probability of $f(x)$ by $O(T/N)$. The probability that $f(x)$ comes from the fixed coordinates is at most ST/N when x is uniformly chosen from $[N]$. Therefore, the overall probability of inverting $f(x)$ is $O(ST/N)$

Specifically, we hope to reduce the AI quantum models to simpler BF quantum models, then export similar proofs from quantum idealized models. Moreover, we hope to leverage security proofs proven using classical presampling techniques to the quantum setting.

Recently, Chung et al. [CGLQ20] give a new general technique for analyzing AI models for quantum adversaries. This technique reduces (Q)AI security against attackers with (quantum) advice to analyzing a multi-instance (MI) security against attackers *without* advice. They use this technique to prove $ST/N + T^2/N$ bound for inverting random functions in the AI-QROM model. Although the new technique is quite general and easier to use than compression, it inherently requires a proof of direct product type statement, to show the security of multiple-instance has an exponential decay in the number of instances. For practical symmetric key and hash function constructions, proving such statements may be challenging. On the contrast, analyzing a single-instance in the BF-model is considerably simpler.

1.1 Our Results

One natural attempt to develop quantum presampling is to leverage the optimal presampling lemma of Coretti et al. [CDGS18] for ROM (which has been exported to ICM/RPM/GGM by Coretti et al. [CDG18]).

Barriers for leveraging presampling to the quantum setting. In Section 3, we show that such leveraging has a technical barrier: it resolves a major open problem in quantum computing [AA11], namely Conjecture 1, which asserts that any quantum algorithm can be approximated on most inputs by a classical algorithm which is at most polynomially slower in terms of query complexity. This open problem, dating back to (according to [AA11]) 1999 or earlier, was included twice in Aaronson's list of "ten semi-grand challenges for quantum computing theory" [Aar05b,Aar10].

In [AA11], Aaronson and Ambainis proposed an approach, which became well-known as the Aaronson-Ambainis conjecture, towards this open problem via analysis of Boolean function. In specific, Aaronson-Ambainis conjecture asserts that any bounded low-degree function on the discrete cube has a variable with influence $\mathrm{poly}(\mathbf{Var}[f]/\deg(f))$ (see Conjecture 2). Despite much effort [DFKO06,Bac12,OY16,MA12,KK19], this open problem and the closely related Aaronson-Ambainis conjecture seem still quite open: only some special functions are confirmed [Bac12,OSSS05,MA12], and the best-known bound for general functions is still exponentially far from the conjectured bound [DFKO06,OY16,DMP17].

Unifying presampling via concentration bounds. Facing with this barrier, we revisit the presampling techniques in the classical setting. To this end, with only standard concentration bounds, we give a simpler and unified proof for the classical presampling theorems of both ROM [CDGS18] and RPM [CDG18], using an equivalent characterization of P-BF-ROM/P-BF-RPM.

Instead of viewing P-BF-ROM as a random function/oracle with at most P prefixed inputs/outputs, we define it by a classical *randomized* algorithm f making at most P queries. The random function is sampled in the following way: sample a random H, compute f^H; restart the whole procedure (including sampling a random function H) if the output of f^H is not 1. In other words, it defines a distribution over all functions conditioned on f^H outputs 1. The security game then is under the oracle access to the function H.

With the concentration bounds and the alternative definition, we show a unified proof for the classical presampling theorems. The proof is much simpler than the original proof [CDGS18], as the original proof needs to first decompose a random oracle distribution H with advice into dense distributions (a technique used in the area of communication complexity [GLM+16]), and then argue indistinguishability between a dense distribution and a uniform distribution. With almost no additional effort, the proof can be used to achieve the theorem for AI-RPM, in [CDG18]. Note that the proof is optimal, as it matches the theorem in [CDGS18], which is known to be optimal. Again the small additive security loss and the multiplicative version are particularly appealing for obtaining exact bounds.

Quantum presampling and applications to quantum random oracles. With the new definition, it is natural to adapt the definition of P-BF-ROM to P-BF-*QROM*. P-BF-QROM is defined by a P-query *quantum* algorithm f making superposition queries. Similarly, the random function is sampled in the following way: sample a random H, compute f^H; restart the whole procedure (including sampling a random function H) if the output of f^H is not 1.

Using the same proof for classical presampling, we obtain the quantum presampling.

Theorem 1. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if a security game G is $\varepsilon(T)$-secure in the P-BF-QROM, then it is (S, T, ε')-secure in the AI-QROM, for some ε' such that,*

$$\varepsilon'(S,T) \leq \varepsilon(T) + \frac{(S + \log \gamma^{-1})T^{\mathsf{comb}}}{P} + \gamma.$$

If G is $\varepsilon(T)$-secure in the P-BF-QROM for $P \geq (S + \log \gamma^{-1})T^{\mathsf{comb}}$, then it is (S, T, ε')-secure in the AI-QROM, for some ε' such that,

$$\varepsilon'(S,T) \leq 2 \cdot \varepsilon(T) + \gamma.$$

$T^{\mathsf{comb}} = T + T_{\mathsf{Ver}}$ is the combined query complexity and T_{Ver} is the query complexity for the challenger .

Note that it is optimal in the sense that it exactly matches the classical presampling theorem, which is optimal.

Therefore, to obtain security in the AI-QROM, one needs to obtain its security in the P-BF-QROM. We show Zhandry's compressed oracle [Zha19] would

be a useful tool for deriving bounds in the P-BF-QROM, by presenting the first non-trivial security analysis of (salted) Merkle-Damgård Hash Functions (MDHF) in the P-BF-QROM and AI-QROM.

Theorem 2. *G_{MDHF} is $\varepsilon(S,T) = \tilde{O}(ST^3/M)$-secure in the AI-QROM.*

In the classical setting, Coretti et al. [CDGS18] show an attack with advantage $\Omega(ST^2/M)$ (which is optimal), and Akshima et al. [ACDW20] show an attack for 2-block MDHF with advantage $\Omega((ST+T^2)/M)$. We observe that the attack by Akshima et al. [ACDW20] can be extended to the quantum setting, and yield an attack with advantage $ST^2/M + T^3/M$. However, it is not clear if the attack of Coretti et al. [CDGS18] can be extended to the quantum setting because of the usage of function iteration in the attack. Our bound suggests that, the speedup of quantum adversaries is limited to a factor T. Further closing this gap is an intriguing question.

1.2 Open Problems

Optimal Presampling for Quantum Advice. While our work provides a framework for the presampling technique for classical advice, we are not able to give presampling techniques for quantum advice. The difficulty comes from the fact that quantum advice would be completely destroyed once a single round of online computation was done. Note that the barrier would be overcome using the similar idea in [CGLQ20], by boosting the succeeding probability and applying Gentle Measurement Lemma [Aar05a]. However, we suspect the resulting statement may not be optimal.

Bit-Fixing Security of One-Way Permutations. While P-BF-QRPM (quantum random permutation model) is well defined following our definition for P-BF-QROM, it is not clear how to prove the security in this model. We hope one of the following two approaches would work: (1) analyzing the probability distribution of the permutations in P-BF-QRPM, and using one-way to hiding lemma [AHU19] to derive the bound for the online computation; (2) with "compressed permutation" techniques similar to Zhandry's compressed oracle techniques, a similar proof to that in the P-BF-QROM would be possible.

Closing the gap for MDHF. As discussed in the previous section, closing the gap for the security of MDHF in the AI-QROM is also an intriguing question.

2 Preliminaries

For any $n \in \mathbb{N}$, we denote $[n]$ to be the set $\{1, 2, ..., n\}$. We denote $\mathbb{Z}/n\mathbb{Z} = \{0, 1, ..., n-1\}$ as the ring of integers modulo n, and $\mathbb{F}_2 = \{0, 1\}$ as the binary finite field. For a complex vector $\mathbf{x} \in \mathbb{C}^n$, we denote the L^2-norm $|\mathbf{x}| = |\mathbf{x}|_2 = \sqrt{\sum_{i\in[n]} x_i \overline{x_i}}$. In algorithms, we denote $a \leftarrow_\$ A$ to be taking a as a uniformly independently sampled element of A.

Next, we recall some basic facts about quantum computation, and review the relevant literature on the quantum random oracle model.

2.1 Quantum Computation

A quantum system Q is defined over a finite set B of classical states. A **pure state** over Q is a unit vector in $\mathbb{C}^{|B|}$, which assigns a complex number to each element in B. In other words, let $|\phi\rangle$ be a pure state in Q, we can write $|\phi\rangle$ as a column vector:

$$|\phi\rangle = \sum_{x \in B} \alpha_x |x\rangle$$

where $\sum_{x \in B} |\alpha_x|^2 = 1$ and $\{|x\rangle\}_{x \in B}$ is called the "*computational basis*" of $\mathbb{C}^{|B|}$. The computational basis forms an orthonormal basis of $\mathbb{C}^{|B|}$. We define $\langle\phi|$ to be the row vector that is the conjugate of $|\phi\rangle$.

Given two quantum systems Q_1 over B_1 and Q_2 over B_2, we can define a *product* quantum system $Q_1 \otimes Q_2$ over the set $B_1 \times B_2$. Given $|\phi_1\rangle \in Q_1$ and $|\phi_2\rangle \in Q_2$, we can define the product state $|\phi_1\rangle \otimes |\phi_2\rangle \in Q_1 \otimes Q_2$.

We say $|\phi\rangle \in Q_1 \otimes Q_2$ is *entangled* if there does not exist $|\phi_1\rangle \in Q_1$ and $|\phi_2\rangle \in Q_2$ such that $|\phi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$. For example, consider $B_1 = B_2 = \{0, 1\}$ and $Q_1 = Q_2 = \mathbb{C}^2$, $|\phi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ is entangled. Otherwise, we say $|\phi\rangle$ is unentangled.

A state $|\phi\rangle \in Q$ can be manipulated by a unitary operator $U \in \mathbb{C}^{|B| \times |B|}$. The resulting state $|\phi'\rangle = U|\phi\rangle$. We denote the trace norm $\|U\|_{\mathrm{tr}}$ to be $\frac{1}{2} \operatorname{Tr} \sqrt{U^\dagger U}$.

We extract classical information from a quantum state $|\phi\rangle$ by performing a *measurement*. A measurement is specified by an orthonormal basis, typically the computational basis, and the probability of getting result x is $|\langle x|\phi\rangle|^2$. After the measurement, $|\phi\rangle$ "collapses" to the state $|x\rangle$ if the result is x.

For example, given the pure state $|\phi\rangle = \frac{3}{5}|0\rangle + \frac{4}{5}|1\rangle$ measured under $\{|0\rangle, |1\rangle\}$, with probability 9/25 the result is 0 and $|\phi\rangle$ collapses to $|0\rangle$; with probability 16/25 the result is 1 and $|\phi\rangle$ collapses to $|1\rangle$.

We assume quantum circuits can implement any unitary transformation (by using these basic gates, Hadamard, phase, CNOT and $\frac{\pi}{8}$ gates), in particular the following two unitary transformations:

- **Classical Computation:** Given a function $f : X \to Y$, one can implement a unitary U_f over $\mathbb{C}^{|X| \cdot |Y|} \to \mathbb{C}^{|X| \cdot |Y|}$ such that for any $|\phi\rangle = \sum_{x \in X, y \in Y} \alpha_{x,y} |x, y\rangle$,

  $$U_f |\phi\rangle = \sum_{x \in X, y \in Y} \alpha_{x,y} |x, y \oplus f(x)\rangle$$

  Here, $\oplus$ is a commutative group operation defined over Y. In particular, if f is given as a classical circuit C, there exists an efficient implementation of the unitary U_f using $|C|$ ancillas, and each gate is evaluated at most twice.
- **Quantum Fourier Transform:** For every $n \in \mathbb{N}$, the quantum Fourier transform QFT_n is a unitary operation, that is given a quantum state $|\phi\rangle = \sum_{j \in \mathbb{Z}/n\mathbb{Z}} x_j |j\rangle$, outputs $|\psi\rangle = \sum_{k \in \mathbb{Z}/n\mathbb{Z}} y_k |k\rangle$ where the sequence $\{y_k\}_k$ is the Fourier transform to the sequence $\{x_j\}_j$, i.e.

  $$y_k = \frac{1}{\sqrt{n}} \sum_{j \in \mathbb{Z}/n\mathbb{Z}} \omega_n^{jk} x_j$$

where $\omega_n = e^{2\pi i/n}$, and i is the imaginary unit.

2.2 Quantum Random Oracle Model

Here, for the completeness of the paper, we recall the background of quantum random oracle model and the compressed oracle technique introduced by [Zha19]. This section is taken verbatim from Section 2.2 of [CGLQ20].

An oracle-aided quantum algorithm can perform quantum computation as well as quantum oracle query. A quantum oracle query for an oracle $f : [N] \to [M]$ is modeled as a unitary $U_f : |x\rangle |u\rangle = |x\rangle |u + f(x)\rangle$, where $+$ denotes addition in integer ring $\mathbb{Z}/M\mathbb{Z}$ (we take the natural bijection that $M \simeq 0$, but any bijection $[M] \leftrightarrow \mathbb{Z}/M\mathbb{Z}$ suffices for our purposes).

A random oracle is a random function $H : [N] \to [M]$. The random function H is chosen at the beginning. A quantum algorithm making T oracle queries to H can be modeled as the following: it has three registers $|x\rangle, |u\rangle, |z\rangle$, where $x \in [N], u \in \mathbb{Z}/M\mathbb{Z}$ and z is the algorithm's internal working memory; it starts with some input state $|0\rangle |0\rangle |\psi\rangle$, then it applies a sequence of unitary to the state: $U_0, U_H, U_1, U_H, \cdots, U_{T-1}, U_H, U_T$ and a final measurement over computational basis. Each U_H is the quantum oracle query unitary: $U_H |x\rangle |u\rangle = |x\rangle |u + H(x)\rangle$ and U_i is the local quantum computation that is independent of H. We can always assume there is only one measurement which is a measurement on computational basis and applied at the last step of the algorithm.

2.3 Compressed Oracle

Here we briefly recall some backgrounds about compressed oracle techniques, first introduced in [Zha19]. More details are provided in Appendix A.

Intuitively, compressed oracle is an analogy of classical lazy sampling method. To simulate a random oracle, one can sample $H(x)$ for all inputs x and store everything in quantum accessible registers. Instead of recording all the information of H in the registers, Zhandry provides a solution which is useful to argue the amount of the information an algorithm knows about the random oracle.

The oracle register records a database/list that contains the output on each input x, the output is an element in $\mathbb{Z}/M\mathbb{Z} \cup \{\perp\}$, where $\perp$ is a special symbol denoting that the value is "uninitialized". The database is initialized as an empty list D_0 of length N, in other words, it is initialized as the pure state $|\emptyset\rangle := |\perp, \perp, \cdots, \perp\rangle$. Let $|D|$ denote the number of entries in D that are not $\perp$. Define $D(x)$ to be the x-th entry. Intuitively, $D(x)$ can be seen as the output of the oracle on x if $D(x) \neq \perp$; otherwise, the oracle's output on x is still undetermined.

For any D and x such that $D(x) = \perp$, we define $D \cup (x, u)$ to be the database D', such that for every $x' \neq x$, $D'(x') = D(x)$ and at the input x, $D'(x) = u$.

The compressed standard oracle is the unitary $\mathsf{CStO} := \mathsf{StdDecomp} \circ \mathsf{CStO}' \circ \mathsf{StdDecomp}$ operating on the joint system of the algorithm's registers and oracle's registers, where

- $\mathsf{CStO}'|x,u\rangle|D\rangle = |x,u+D(x)\rangle|D\rangle$ when $D(x) \neq \perp$, which writes the output of x defined in D to the u register. This operator will never be applied on an x, D where $D(x) = \perp$.
- $\mathsf{StdDecomp}(|x\rangle \otimes |D\rangle) := |x\rangle \otimes \mathsf{StdDecomp}_x |D\rangle$, where $\mathsf{StdDecomp}_x |D\rangle$ works on the x-th register of the database $D(x)$. Intuitively, it swaps a uniform superposition $\frac{1}{\sqrt{M}} \sum_y |y\rangle$ with $|\perp\rangle$ on the x-th register. Formally,
  - If $D(x) = \perp$, $\mathsf{StdDecomp}_x$ maps $|\perp\rangle$ to $\frac{1}{\sqrt{M}} \sum_y |y\rangle$, or equivalently, $\mathsf{StdDecomp}_x |D\rangle = \frac{1}{\sqrt{M}} \sum_y |D \cup (x,y)\rangle$. Intuitively, if the database does not contain information about x, it samples a fresh y as the output of x.
  - If $D(x) \neq \perp$, $\mathsf{StdDecomp}_x$ works on the x-th register, and it is an identity on $\frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |y\rangle$ for all $u \neq 0$; it maps the uniform superposition $\frac{1}{\sqrt{M}} \sum_y |y\rangle$ to $|\perp\rangle$.
    More formally, for a D' such that $D'(x) = \perp$,

    $$\mathsf{StdDecomp}_x \frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |D' \cup (x,y)\rangle = \frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |D' \cup (x,y)\rangle \text{ for any } u \neq 0,$$

    and,

    $$\mathsf{StdDecomp}_x \frac{1}{\sqrt{M}} \sum_y |D' \cup (x,y)\rangle = |D'\rangle.$$

    Since all $\frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |y\rangle$ and $|\perp\rangle$ form a basis, these requirements define a unique unitary operation.

A quantum algorithm making T oracle queries to a compressed oracle can be modeled as the following: the algorithm has three registers $|x\rangle, |u\rangle, |z\rangle$, where $x \in [N], u \in \mathbb{Z}/M\mathbb{Z}$ and z is the algorithm's internal working memory; it starts with some input state $|0\rangle |0\rangle |\psi\rangle$; the joint state of the algorithm and the compressed oracle is $|0\rangle |0\rangle |\psi\rangle \otimes |\emptyset\rangle$. It then applies a sequence of unitary to the state: U_0, CStO, U_1, CStO, $\cdots$, U_{T-1}, CStO, U_T and a final measurement over computational basis.

Zhandry proves that, the quantum random oracle model and the compressed standard oracle model are perfectly indistinguishable by any *unbounded* quantum algorithm.

In this work, we only consider query complexity, and thus simulation efficiency is irrelevant to us. Looking ahead, we simulate a random oracle as a compressed standard oracle to help us analyze security games with the help from the following lemmas. Both lemmas are proven in [Zha19,CGLQ20].

The first lemma gives a general formulation of the overall state of $\mathcal{A}$ and the compressed standard oracle after $\mathcal{A}$ makes T queries, even conditioned on arbitrary measurement results. Looking ahead, it gives a characterization of P-BF-QROM (defined in Section 4.1) if the oracle is simulated as a compressed standard oracle.

Lemma 1. *If $\mathcal{A}$ makes at most T queries to a compressed standard oracle, assuming the overall state of $\mathcal{A}$ and the compressed standard oracle is $\sum_{z,D} \alpha_{z,D} |z\rangle_{\mathcal{A}} |D\rangle_H$ where $|z\rangle$ is $\mathcal{A}$'s registers and $|D\rangle$ is the oracle's registers, then it only has support on all D such that $|D| \leq T$. In other words, the overall state can be written as,*

$$\sum_{z,D:|D|\leq T} \alpha_{z,D} |z\rangle_{\mathcal{A}} \otimes |D\rangle_H.$$

Moreover, it is true even if the state is conditioned on arbitrary outcomes (with non-zero probability) of $\mathcal{A}$'s intermediate measurements.

The second lemma provides a quantum analogue of lazy sampling in the classical ROM.

Lemma 2 (Lemma 5 in [Zha19]). *Let H be a random oracle from $[N] \to [M]$. Consider a quantum algorithm $\mathcal{A}$ making queries to the standard oracle and outputting tuples $(x_1, \cdots, x_k, y_1, \cdots, y_k, z)$. Supposed the random function H is measured after $\mathcal{A}$ produces its output. Let R be an arbitrary set of such tuples. Suppose with probability p, $\mathcal{A}$ outputs a tuple such that (1) the tuple is in R and (2) $H(x_i) = y_i$ for all i. Now consider running $\mathcal{A}$ with the compressed standard oracle CStO, and suppose the database D is measured after $\mathcal{A}$ produces its output. Let p' be the probability that (1) the tuple is in R and (2) $D(x_i) = y_i$ (in particular, $D(x_i) \neq \perp$) for all i. Then $\sqrt{p} \leq \sqrt{p'} + \sqrt{k/M}$.*

Moreover, it is true even if it is conditioned on arbitrary outcomes (with non-zero probability) of $\mathcal{A}$'s intermediate measurements.

2.4 Security Game with Classical Advice

In this paper, we only focus on the case where advice is classical. Therefore in the rest of the presentation, "advice" simply means "classical advice". The following definitions are defined in [CGLQ20].

Definition 1 (Algorithm with Advice). *An (S, T) (query) classical/quantum algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with (oracle-dependent) advice consists of two procedures:*

- *let $H, \tilde{H}$ be two oracles accessed by $\mathcal{A}_1, \mathcal{A}_2$ respectively in the offline and online phases;*
- *$\alpha \leftarrow \mathcal{A}_1(H)$, which is an arbitrary (unbounded) function of H, and outputs an S-bit α;*
- *$|\mathsf{ans}\rangle \leftarrow \mathcal{A}_2^{\tilde{H}}(\alpha, \mathsf{ch})$, which is an unbounded algorithm that takes advice α, a challenge ch, makes at most T (classical or quantum respectively) queries to $\tilde{H}$, and outputs an answer, which we measure in the computational basis to obtain the classical answer ans.*

Note that we do not need to distinguish if $\mathcal{A}_1$ is classical or quantum because as long as $\mathcal{A}_1$ is unbounded, they have the same computational power. We say $\mathcal{A}$ is quantum if $\mathcal{A}_2$ makes quantum queries and otherwise $\mathcal{A}$ is classical. In this work, we will mainly focus on $\mathcal{A}$ being quantum and the case of $\mathcal{A}$ being classical will be provided mainly in the preliminary Section 2.5.

Below, we will use the words "adversary" and "algorithm" interchangeably, especially when we consider interactive security games shortly after.

Definition 2 (Security Game). *Let H be a random oracle $[N] \to [M]$. A (non-interactive) security game $G = (C)$ is specified by a challenger $C = (\mathsf{Samp}, \mathsf{Query}, \mathsf{Ver})$, where:*

1. $\mathsf{ch} \leftarrow \mathsf{Samp}^H(r)$ *is a classical algorithm that takes randomness $r \in R$ as input, and outputs a challenge* ch.
2. $\mathsf{Query}^H(r, \cdot)$ *is a deterministic classical algorithm that hardcodes the randomness r and provides adversary's online queries*[6].
3. $b \leftarrow \mathsf{Ver}^H(r, \mathsf{ans})$ *is a deterministic classical algorithm that takes the input* ans *and outputs a decision b indicating whether the game is won.*

For every algorithm with advice, i.e. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define

$$\mathcal{A} \Longleftrightarrow C(H) := \mathsf{Ver}^H\left(r, \mathcal{A}_2^{\tilde{H}}(\mathcal{A}_1(H), \mathsf{Samp}^H(r))\right)$$

to be the binary variable indicating whether $\mathcal{A}$ successfully makes the challenger output 1, or equivalently if $\mathcal{A}$ wins the security game, where $\tilde{H}(\cdot) := \mathsf{Query}^H(r, \cdot)$. Additionally, we define T_{Ver} be the query complexity of computing Ver^H.

Definition 3 (Security in the AI-ROM/AI-QROM). *We define the security in the AI-ROM/AI-QROM of a security game $G = (C)$ to be*

$$\delta = \delta(S, T) := \sup_{\mathcal{A}} \Pr_{H,r,\mathcal{A}}\left[\mathcal{A} \Longleftrightarrow C(H) = 1\right],$$

where $\mathcal{A}$ in the probability denotes the randomness of the algorithm, and supremum is taken over all classical or quantum (S, T) algorithm $\mathcal{A}$ in the AI-ROM or AI-QROM respectively.

Additionally, we say a security game G is δ-secure if its security is at most δ.

Definition 4. *We call the security game a* ***decision*** *game if* $\mathsf{ans} \in \{0, 1\}$.

Definition 5 (Advantage against Decision Games). *We define the advantage of $\mathcal{A}$ for a decision game G to be*

$$\varepsilon = \varepsilon(S, T) := \delta(S, T) - 1/2,$$

if it has winning probability $\delta(S, T)$.

Definition 6 (Best Advantage of Decision Games). *We define the best advantage of a decision game G in AI-ROM/AI-QROM to be $\varepsilon(S, T) := \delta(S, T) - 1/2$ if G has security $\delta(S, T)$ in AI-ROM/AI-QROM.*

[6] As an example, for most applications, $\mathsf{Query}^H(r, \cdot) = H(\cdot)$.

2.5 Presampling Techniques for Random Oracles

We recall classical presampling techniques for random oracles in [CDGS18].

Definition 7 ((N, M)-source). *An (N, M)-source is a random variable X with range $[M]^N$.*

An oracle $\mathcal{O} : [N] \to [M]$ can be represented by a string in $[M]^N$. In the rest of the work, we will sometimes treat an oracle as an element in $[M]^N$. Drawing an oracle from a certain distribution is equivalent to sampling a random variable from the corresponding (N, M)-source.

Definition 8 (P-bit-fixing). *An (N, M)-source is called* P-bit-fixing *if it is fixed on at most P coordinates and uniform on the rest.*

They then defined security in the P-BF-ROM.

Definition 9 (P-BF-ROM). *A security game in the P-BF-ROM consists the following two procedures:*

- *Before the challenging phase, the offline algorithm $\mathcal{A}_1$ runs a (randomized) algorithm to generate a list $\mathcal{L} = \{(x_i, y_i)\}_{i\in[P]}$ containing at most P input-output pairs (all x_is are distinct).*
- *In the challenging phase, the security game is executed with an online algorithm $\mathcal{A}_2$ and oracle access to H. H is a function drawn from the P-bit-fixing distribution and the prefixed inputs/outputs are $\mathcal{L}$.*

Note that $\mathcal{A}_2$ knows the strategy of $\mathcal{A}_1$, but nothing else.

Remark 1. In [CDGS18], the definition of P-BF-ROM allows $\mathcal{A}_2$ to obtain the list $\mathcal{L}$ generated by $\mathcal{A}_1$. In our definition, $\mathcal{A}_2$ only knows the strategy of the offline algorithm $\mathcal{A}_1$. We observe that Definition 9 is a weaker definition and is enough for deriving their main theorem Theorem 3.

The following lemma was given in [CDGS18]. It shows that a random oracle distribution conditioned on advice is very close to a convex combination of P-bit-fixing distributions.

Lemma 3. *Let X be distributed uniformly over $[M]^N$ and $Z := f(X)$, where $f : [M]^N \to \{0,1\}^S$ is an arbitrary function. For any $\gamma > 0$ and $P \in \mathbb{N}$, there exists a family $\{Y_z\}_{z\in\{0,1\}^S}$ of convex combinations Y_z of P-bit-fixing (N, M)-sources such that for any classical distinguisher $\mathcal{D}$ taking an S-bit input and querying at most $T < P$ coordinates of its oracle,*

$$\left|\Pr\left[\mathcal{D}^X(f(X)) = 1\right] - \Pr\left[\mathcal{D}^{Y_{f(X)}}(f(X)) = 1\right]\right| \;\le\; \frac{(S + \log 1/\gamma) \cdot T}{P} + \gamma$$

and

$$\Pr\left[\mathcal{D}^X(f(X)) = 1\right] \;\le\; 2^{(S+\log 1/\gamma)T/P} \cdot \Pr\left[\mathcal{D}^{Y_{f(X)}}(f(X)) = 1\right] + \gamma.$$

Note that the case of getting $X, Z := f(X)$ is the AI-ROM, and the case of getting Y_Z, Z is the P-BF-ROM. The lemma implies the two main theorems (Theorem 5, 6) of [CDGS18].

Theorem 3. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if a security game G is $\varepsilon(T)$-secure in the P-BF-ROM, then it is (S, T, ε')-secure in the AI-ROM, for some ε' such that,*

$$\varepsilon'(S,T) \leq \varepsilon(T) + \frac{(S + \log \gamma^{-1})T^{\mathsf{comb}}}{P} + \gamma.$$

If G is $\varepsilon(T)$-secure in the P-BF-ROM for $P \geq (S + \log \gamma^{-1})T^{\mathsf{comb}}$, then it is (S, T, ε')-secure in the AI-ROM, for some ε' such that,

$$\varepsilon'(S,T) \leq 2 \cdot \varepsilon(T) + \gamma.$$

$T^{\mathsf{comb}} = T + T_{\mathsf{Ver}}$ is the combined query complexity and T_{Ver} is the query complexity for the challenger .

Therefore, with the above lemma and theorems, [CDGS18] prove the security of several cryptographic applications in the AI-ROM. The idea is to first switch to the P-BF-ROM and then argue its security in this model. To prove the security of one-way functions (OWF) in the AI-ROM, they can instead argue the security in the P-BF-ROM, which is much easier to argue than that in the AI-ROM. Informally, if the challenge y is not in the list $\mathcal{L}$, to invert y in the P-BF-ROM is as difficult as that in the ROM. Therefore, the overall security is at most $(P + T)/\min\{N, M\}$ in the P-BF-ROM. Combining with Theorem 3, they get the desired bound for the security of OWF in the AI-ROM.

2.6 Aaronson-Ambainis Conjecture

A major open problem in quantum computing is whether super-polynomial quantum speedups need structures of inputs. The following conjecture captures this question.

Conjecture 1 (folklore, see [AA11]). Let $\mathcal{A}$ be a quantum algorithm making T queries to a Boolean input $x = (x_1, \cdots, x_n)$. For any $\varepsilon > 0$, there is a deterministic classical algorithm that makes $\mathrm{poly}(T, 1/\varepsilon, 1/\delta)$ queries to the x_i's, and that approximates $\mathcal{A}$'s acceptance probability within an additive error ε on a $(1 - \delta)$ fraction of inputs.

This conjecture is a central open problem in the area of quantum computing [Aar05b,Aar10]. In the paper [AA11], Aaronson and Ambainis proposed a new conjecture (a.k.a Aaronson-Ambainis conjecture) which is sufficient to affirm Conjecture 1. In specific, they conjectured that any low-degree function $f : \{-1, 1\}^n \to [0, 1]$ has an influential variable.

Conjecture 2 ([AA11]). Let $f : \{-1,1\}^n \to [0,1]$ be a degree-d polynomial. We define its variance as $\mathbf{Var}[f] := \mathbb{E}_x[f(x)^2] - (\mathbb{E}_x[f(x)])^2$. For each $i \in [n]$, its influence is defined as $I_i(f) := \mathbb{E}_x\left[\left(f(x) - f(x^i)\right)^2\right]$, where x^i is the string obtained by flipping the i-th bit of x. Then there is a coordinate $i \in [n]$ such that

$$I_i(f) = \left(\mathbf{Var}\left[f\right]/d\right)^{O(1)}.$$

Despite much effort [DFKO06,Bac12,OY16,MA12,KK19,LZ20], both Conjecture 1 and Conjecture 2 are still quite open. Special classes were confirmed by several papers [Bac12,OSSS05,MA12]. The best known bound for general functions is still exponentially far from conjectured [DFKO06,OY16,DMP17].

Li and Zhang [LZ20] provided an equivalent form[7] of Conjecture 1, which seems easier to prove and will be used in this paper. Similar ideas are also explored by Keller and Klein [KK19]. Given a (classical or quantum) distinguisher $\mathcal{A}$, let $\mathbb{E}[\mathcal{A}] = \mathbb{E}_X\left[\Pr[\mathcal{A}^X = 1]\right]$ and $\mathbf{Var}[\mathcal{A}] = \mathbb{E}_X\left[\Pr[\mathcal{A}^X = 1] - \mathbb{E}[\mathcal{A}]\right]^2$. Here, X is uniformly distributed over $[M]^N$.

Conjecture 3. Let $\mathcal{A}$ be a quantum distinguisher that makes T queries to an oracle $[N] \to \{0,1\}$. Then there exists a $\mathrm{poly}(T/\mathbf{Var}[\mathcal{A}])$-bit-fixing $(2,N)$-source Y (i.e., there is a list $\mathcal{L}$ containing at most $\mathrm{poly}(T/\mathbf{Var}[\mathcal{A}])$ input-output pairs, and Y is uniformly distributed over $\{0,1\}^N$ conditioned on some coordinates are fixed according to $\mathcal{L}$) such that

$$\left|\Pr\left[\mathcal{A}^Y = 1\right] - \mathbb{E}[\mathcal{A}]\right| \geq \mathrm{poly}(\mathbf{Var}\left[\mathcal{A}\right]/T).$$

For the sake of completeness, we present the proof of the equivalence between Conjecture 1 and Conjecture 3 in the appendix. The nontrivial direction is to show how Conjecture 3 implies Conjecture 1. It follows the general strategy of the argument of Midrijanis [Mid04] which shows that any Boolean function can be computed by a classical decision tree of depth at most the block sensitivity times the polynomial degree.

2.7 Concentration Bounds

The following claim and lemmas of concentration bounds will be used in our proof. We prove them in this section.

Claim 1. *Let $X_1, \ldots, X_N$ be indicators (potentially correlated, binary random variables). Let $Y_1, \ldots, Y_g$ be binary variables such that each Y_i is uniformly randomly sampled from $X_1, \ldots, X_N$. Suppose that*

$$\Pr[Y_1 = 1 \wedge \cdots \wedge Y_g = 1] \leq \alpha^g,$$

[7] In fact, our setting is a little different from that in [LZ20], in which $\mathcal{A}$ can be any degree-d bounded polynomial. However, the proof in [LZ20] can be generalized without effort to our setting.

then

$$\Pr\left[\sum_{i\in[N]} X_i \geq \delta N\right] \leq \left(\frac{\alpha}{\delta}\right)^g.$$

Proof. Let E denote the event $Y_1 = 1 \wedge \cdots \wedge Y_g = 1$. We have,

$$\Pr\left[\sum_{i\in[N]} X_i \geq \delta N\right] \leq \frac{\Pr[E]}{\Pr\left[E \,\middle|\, \sum_{i\in[N]} X_i \geq \delta N\right]} \leq \frac{\alpha^g}{\delta^g},$$

where the second inequality is because the probability that $Y_1, \ldots, Y_g$ are all 1 is at least δ^g conditioning on that there are at least δN ones among $X_1, \ldots, X_N$. □

We first define random variables $Y_{<i}$: $Y_{<i} = 1$ if and only if $Y_1 = Y_2 = \cdots = Y_{i-1} = 1$. $Y_{<1}$ is always equal to 1. We then show two concentration bounds using the claim above. The first one is a multiplicative bound and the second one is an additive bound.

Lemma 4. *Define X_i, Y_i as in Claim 1. Let S', T, g be arbitrary integers, and $P := gT$. Suppose that, for every $i \in [g]$,*

$$\Pr[Y_i = 1 | Y_{<i} = 1] \leq \varepsilon,$$

then,

$$\Pr\left[\frac{1}{N}\sum_{i\in[N]} X_i \geq 2^{S'T/P} \cdot \varepsilon\right] \leq 2^{-S'}.$$

Proof. Let $\alpha := \varepsilon$, and $\delta := 2^{S'T/P} \cdot \varepsilon$. Note that,

$$\Pr[Y_1 = 1 \wedge \cdots \wedge Y_g = 1] = \prod_{i=1}^{g} \Pr[Y_i = 1 | Y_{<i} = 1] \leq \alpha^g.$$

By Claim 1,

$$\Pr\left[\sum_{i\in[N]} X_i \geq \delta N\right] \leq \left(\frac{\alpha}{\delta}\right)^g = \left(\frac{\varepsilon}{2^{S'T/P} \cdot \varepsilon}\right)^g = 2^{-S'}.$$

□

Lemma 5. *Define X_i, Y_i as in Claim 1. Let S', T, g be arbitrary integers, and $P := gT$. Suppose that, for every $i \in [g]$,*

$$\Pr[Y_i = 1 | Y_{<i} = 1] \leq \varepsilon,$$

then,

$$\Pr\left[\frac{1}{N}\sum_{i\in[N]} X_i \geq \varepsilon + \frac{S'T}{P}\right] \leq 2^{-S'}.$$

Proof. Let $\alpha := \varepsilon$, and $\delta := \varepsilon + S'T/P$. We assume that $\varepsilon + S'T/P \leq 1$, otherwise the statement is trivially true. Note that,

$$\Pr[Y_1 = 1 \wedge \cdots \wedge Y_g = 1] = \prod_{i=1}^{g} \Pr[Y_i = 1 | Y_{<i} = 1] \leq \alpha^g \ .$$

By Claim 1,

$$\begin{aligned} \Pr\left[\sum_{i\in[N]} X_i \geq \delta N\right] &\leq \left(\frac{\varepsilon}{\varepsilon + S'T/P}\right)^g \\ &\leq \left(1 - \frac{S'T}{P}\right)^g \\ &\leq 2^{-S'}, \end{aligned}$$

where the second inequality uses the assumption that $\varepsilon + S'T/P \leq 1$, the third inequality uses the fact $1 - x \leq 2^{-x}$ for any $x \geq 0$ and $P = gT$. □

3 Barriers for Leveraging Presampling Techniques

As we have seen the simple and easy-to-use tools (presampling techniques) in the preliminary Section 2.5, we ask the question: *is it possible to leverage Lemma 3 (and Theorem 3) to the quantum world*? The following conjecture formally states that the presampling technique could reduce security proofs in AI-QROM to those in the simpler "P-BF-QROM"[8]. The conjecture requires a much weaker bound than that in Lemma 3.

Conjecture 4. Let X be distributed uniformly over $[M]^N$ and $Z := f(X)$, where $f : [M]^N \to \{0,1\}^S$ is an arbitrary function. For any $P \in \mathbb{N}$, there exists a family $\{Y_z\}_{z\in\{0,1\}^S}$ of convex combinations Y_z of P-bit-fixing (N, M)-sources such that for any quantum distinguisher $\mathcal{A}$ taking an S-bit input and making T quantum queries of its oracle,

$$\left|\Pr[\mathcal{A}^X(f(X)) = 1] - \Pr[\mathcal{A}^{Y_{f(X)}}(f(X)) = 1]\right| \leq h(S) \cdot T \cdot \left(\frac{\log M}{P}\right)^C .$$

Here C is a universal constant and $h : \mathbb{N} \to \mathbb{R}^+$ can be any function.

Note that this conjecture is weaker than Section 2.5 in the sense that the dependency on S can be arbitrary, but Lemma 3 is polynomial in S.

In this section, we show that even requiring a much weaker bound (Conjecture 4) implies Conjecture 1, which reveals a barrier for leveraging Lemma 3 to the quantum world.

[8] We have not defined what is P-BF-QROM yet. Since we will give a barrier and the following Conjecture 4 does not require a formal definition, we will not formally define it in this section.

Lemma 6. *Conjecture 4 implies Conjecture 3, then Conjecture 1.*

Proof. In fact, we will prove Conjecture 3 only assuming that Conjecture 4 holds for $S = 1$. Let $\mathcal{A}$ be a quantum distinguisher that makes T queries of an oracle in $\{0,1\}^N$. We will show that there exists a $\mathrm{poly}(T/\mathbf{Var}[\mathcal{A}])$-bit-fixing source Y such that the gap between $\Pr[\mathcal{A}^Y = 1]$ and $\mathbb{E}[\mathcal{A}]$ is at least $\sigma/4$. Here, $\sigma = \sqrt{\mathbf{Var}[\mathcal{A}]}$.

The basic idea is as follows. Let $f : \{0,1\}^N \to \{0,1\}$ indicate whether the acceptance probability of $\mathcal{A}$ access to the oracle $\mathcal{O} \in \{0,1\}^N$ is high (say, $f(\mathcal{O}) = 1$ if and only if $\Pr[\mathcal{A}^{\mathcal{O}} = 1] - \mathbb{E}[\mathcal{A}] \geq \sigma/2$). Let $\mathcal{A}_1$ be another quantum distinguisher which (i) takes the bit $f(\mathcal{O})$ as advice, (ii) simulates $\mathcal{A}$ if $f(\mathcal{O}) = 1$, and (iii) makes no queries and rejects if $f(\mathcal{O}) = 0$. On one hand, $\mathcal{A}_1$ and $\mathcal{A}$ have the same acceptance probability when access to any $\mathcal{O} \in f^{-1}(1)$. On the other hand, according to Conjecture 4, for an oracle randomly sampled from $f^{-1}(1)$, $\mathcal{A}_1$ has the similar acceptance probability with oracle access to some bit-fixing source.

More formally, let X be uniformly distributed over $\{0,1\}^N$. For simplicity of notations, we abbreviate $\Pr[\mathcal{A}^{\mathcal{O}} = 1]$ to $\mathcal{A}^{\mathcal{O}}$. Noting that $|\mathcal{A}^{\mathcal{O}} - \mathbb{E}[\mathcal{A}]| \leq 1$ for any $\mathcal{O} \in \{0,1\}^N$, we have

$$\begin{aligned}
\sigma^2 &= \mathbb{E}_X\left[\left|\mathcal{A}^X - \mathbb{E}[\mathcal{A}]\right|^2\right] \\
&\leq \Pr_X\left[|\mathcal{A}^X - \mathbb{E}[\mathcal{A}]| \geq \sigma/2\right] + \Pr_X\left[|\mathcal{A}^X - \mathbb{E}[\mathcal{A}]| \leq \sigma/2\right] \cdot \sigma^2/4 \\
&\leq \Pr_X\left[|\mathcal{A}^X - \mathbb{E}[\mathcal{A}]| \geq \sigma/2\right] + \sigma^2/4.
\end{aligned}$$

So $\Pr_X\left[|\mathcal{A}^X - \mathbb{E}[\mathcal{A}]| \geq \sigma/2\right] \geq 3\sigma^2/4$. By symmetry, we assume

$$\Pr_X\left[\mathcal{A}^X - \mathbb{E}[\mathcal{A}] \geq \sigma/2\right] \geq 3\sigma^2/8. \tag{1}$$

Let $f : \{0,1\}^N \to \{0,1\}$ be defined as follows: $f(X) = 1$ if and only if $\mathcal{A}^X - \mathbb{E}[\mathcal{A}] \geq \sigma/2$. Inequality 1 says that $\Pr_X[f(X) = 1] \geq 3\sigma^2/8$. Let X_1 be the distribution of X conditioned on $f(X) = 1$. Let $\{Y_0, Y_1\}$ be the family of convex combinations of P-bit-fixing sources guaranteed by Conjecture 4. Let $\mathcal{A}_1$ be another quantum distinguisher that (i) takes a 1-bit input, (ii) simulates $\mathcal{A}$ if the input bit is 1, and (iii) makes no queries and rejects if the input bit is 0. It has that

$$\frac{h(1)\cdot T}{P^C} \geq \left|\mathbb{E}_X\left[\mathcal{A}_1^X(f(X))\right] - \mathbb{E}_X\left[\mathcal{A}_1^{Y_{f(X)}}(f(X))\right]\right| \geq \Pr_X[f(X) = 1] \cdot \left|\mathcal{A}^{X_1} - \mathcal{A}^{Y_1}\right|$$

That is, $\left|\mathcal{A}^{X_1} - \mathcal{A}^{Y_1}\right| \leq 8h(1)\cdot T/(3\sigma^2 P^C)$. In particular, there is a P-bit-fixing source Y such that $\left|\mathcal{A}^{X_1} - \mathcal{A}^{Y}\right| \leq 8h(1)\cdot T/(3\sigma^2 P^C)$. Let $P = \lceil \left(\frac{32h(1)\cdot T}{3\sigma^3}\right)^{1/C} \rceil$, then $8h(1)\cdot T/(3\sigma^2 P^C) \leq \sigma/4$. Finally, by the triangle inequality,

$$\left|\mathcal{A}^Y - \mathbb{E}[\mathcal{A}]\right| \geq \left|\mathcal{A}^{X_1} - \mathbb{E}[\mathcal{A}]\right| - \left|\mathcal{A}^Y - \mathcal{A}^{X_1}\right| \geq \sigma/2 - \sigma/4 = \sigma/4.$$

This completes the proof. □

4 Unifying Presampling via Concentration Bounds

As discussed in the last section, the natural extension of Lemma 3 does not work in the quantum world, otherwise we can prove AA conjecture. In this section, we will give a much simpler proof for (classical) Theorem 3 directly, using only concentration bounds, which also unifies the proof for both AI-ROM [CDGS18] and AI-RPM (random permutation model) [CDG18]. The core of the proof is to use an equivalent characterization of the P-BF-ROM. We will then generalize this definition for AI-QROM in the next section.

4.1 A New Characterization of Bit-Fixing

The P-BF-ROM fixes at most P input-output pairs of a random oracle. The failed attempt in the last section tries to classically fix P input-output pairs of a quantum random oracle (which will be queried in superposition later). To overcome the barrier, we may need to '*quantumly*' fix P input-output pairs and avoid the AA conjecture barrier. However, it is not clear how to 'fix quantumly' or 'fix in superposition'.

We realize that the P-BF-ROM can be defined by a bounded query algorithm.

Definition 10 (P-BF-ROM). *A security game in the P-BF-ROM consists the following two procedures:*

- *Before the challenging phase, the offline adversary $\mathcal{A}_1$ prepares a (randomized) algorithm f, and then interacts with a challenger:*
  1. *The challenger samples a random function H;*
  2. *$\mathcal{A}_1$ computes f^H which makes at most P queries to H.*
  3. *$\mathcal{A}_1$ gets a single bit output z of f^H. If $z \neq 1$, it restarts the whole procedure (including sampling a new random function H at the beginning).*
- *In the challenging phase, the security game is executed with an online algorithm $\mathcal{A}_2$ and oracle access to the function H.*

Note that the algorithm f can be inefficient, including running time of f and time for sampling a random H conditioned on $f^H = 1$, except the number of queries are bounded by P.

Definition 10 says that the oracle distribution in the online phase is determined by a P-query bounded algorithm in the pre-computation stage, conditioned on the output of the algorithm f^H being 1. This definition can be easily extended to P-BF-RPM, by simply replacing H with a random permutation.

Next, we show that the P-BF-ROM defined above is exactly equivalent to that defined in Definition 9. In other words, any oracle distribution in the online phase that can be generated in the offline phase of Definition 9, can also be generated in Definition 10, and vice versa.

Lemma 7. *Definition 9 is equivalent to Definition 10.*

Proof. We first show the easy direction: any oracle distribution in the online phase that can be generated in the offline phase of Definition 9, can also be generated in Definition 10.

Assume an algorithm g samples a list $\mathcal{L}$ of at most P input-output pairs and $\mathcal{L}$ defines the P-bit-fixing oracle distribution in Definition 9. We show such a distribution can be generated by conditioning on some algorithm f^H outputting 1. Let f be the following algorithm:

- f runs g as a subroutine and obtains $\mathcal{L} = \{(x_i, y_i)\}$ for at most P distinct x_is.
- f^H queries $x_1, x_2, \cdots$ one by one and it outputs 1 if and only if for all i, $H(x_i) = y_i$.

It is easy to see that the oracle distribution defined by f in Definition 10 is the same as that defined by g in Definition 9.

Now we focus on the opposite direction: any oracle distribution in the online phase that can be generated in the offline phase of Definition 10, can also be generated in Definition 9.

We first assume f is a *deterministic* algorithm. Without loss of generality, f will never query the same input twice as it can simply record all queries it made. A transcript τ of f is defined as a set containing all input-output pairs queried by f. Each transcript will be marked as accepting or rejecting depending on whether f outputs 1 or 0 respectively.

Note that every pair of transcripts τ, τ' is 'disjoint'. Namely, for any τ, τ', there always exists an input x and $y \neq y'$ such that $(x, y) \in \tau$ and $(x, y') \in \tau'$. Moreover, let X_τ be the oracle distribution that is compatible with τ but everywhere else is sampled uniformly at random. Then X_τ and $X_{\tau'}$ are disjoint. We notice that $\{X_\tau\}_\tau$ is indeed a partition of all possible oracles.

Therefore, we can construct the algorithm g as follows:

- g uses f as a subroutine. It obtains all transcripts $\mathcal{T} = \{\tau\}$.
- g samples a transcript τ with probability $M^{-|\tau|}$. Note that $M^{-|\tau|} = |X_\tau|/M^N$, because $\{X_\tau\}_\tau$ is a partition of all possible oracles, we have

$$\sum_{\tau \in \mathcal{T}} M^{-|\tau|} = 1.$$

- If τ is not an accepting transcript, g restarts everything. Otherwise, it outputs $\mathcal{L} = \tau$.

The distribution generated by g is simply bit-fixing sources corresponding to all accepting transcripts. We observe that it is a uniform distribution over all oracles in $\{M_\tau\}$ for τ being an accepting transcript. This is exactly the distribution defined by f.

If f is a randomized algorithm, we construct g in the following way:

– g uses f as a subroutine. It first samples uniform randomness r. It obtains all transcripts $\mathcal{T} = \{\tau\}$ corresponding to $f(;r)$ (which is deterministic).
– g samples a transcript τ with probability $M^{-|\tau|}$.
– If τ is not an accepting transcript, g restarts everything (including sampling randomness r). Otherwise, it outputs $\mathcal{L} = \tau$.

The proof is almost identical to the deterministic case. □

4.2 A Simpler Proof for Theorem 3

We reprove Theorem 3 using concentration bounds. The proof is much simpler than the original proof [CDGS18], as the original proof needs to first decompose a random oracle distribution H with advice into dense distributions (a technique used in the area of communication complexity [GLM+16]), and then argue indistinguishability between a dense distribution and a uniform distribution.

We first recall the theorem.

Theorem 3. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if a security game G is $\varepsilon(T)$-secure in the P-BF-ROM, then it is (S, T, ε')-secure in the AI-ROM, for some ε' such that,*

$$\varepsilon'(S,T) \leq \varepsilon(T) + \frac{(S + \log \gamma^{-1})T^{\mathsf{comb}}}{P} + \gamma.$$

If G is $\varepsilon(T)$-secure in the P-BF-ROM for $P \geq (S + \log \gamma^{-1})T^{\mathsf{comb}}$, then it is (S, T, ε')-secure in the AI-ROM, for some ε' such that,

$$\varepsilon'(S,T) \leq 2 \cdot \varepsilon(T) + \gamma.$$

$T^{\mathsf{comb}} = T + T_{\mathsf{Ver}}$ is the combined query complexity and T_{Ver} is the query complexity for the challenger .

Proof (Reprove Theorem 3).

Let G be a security game with random coin space R. As defined in Definition 2, randomness $i \in R$ is for generating a challenge.

We first prove the second half of the theorem. Fix any (S, T) algorithm $\mathcal{A}$ for G. For a given advice $\alpha \in \{0,1\}^S$, let X_i^α be the random variable indicating if $\mathcal{A}(\alpha, \cdot)$ wins the game G with randomness $i \in R$. More precisely, X_i^α is the following:

– H is sampled at the beginning;
– $\mathcal{A}(\alpha)$ plays the game G, where the challenge ch is sampled by $\mathsf{Samp}^H(i)$ for this fixed i;
– $X_i^\alpha = 1$ if and only if the game is won by $\mathcal{A}(\alpha)$.

Note that X_i^α and $X_{i'}^\alpha$ use the same random H.

Similarly we define Y_j^α to be the random variable that is uniformly at random sampled from $\{X_i^\alpha\}_{i \in R}$. Y_j^α is the random variable indicating if an algorithm $\mathcal{A}(\alpha)$ wins the game in the j-th round, with a uniformly chosen challenge.

We also define $Y^{\alpha}_{<j}$ in a similar way in Section 2.7: it is 1 if and only if all $Y^{\alpha}_1 = \cdots = Y^{\alpha}_{j-1} = 1$. $Y^{\alpha}_{<j}$ is the random variable indicating if an algorithm $\mathcal{A}(\alpha)$ wins all games in the first $(j-1)$ rounds, with uniformly chosen challenges for each round.

Since G is ε-secure in the P-BF-ROM for $P \geq (S + \log \gamma^{-1})T^{\mathsf{comb}} = gT^{\mathsf{comb}}$, we have the following claim:

Claim 2.

$$\Pr\left[Y^{\alpha}_j = 1 \,|\, Y^{\alpha}_{<j} = 1\right] \leq \varepsilon \text{ for all } j \leq g := (S + \log \gamma^{-1}).$$

Proof. Fixing a $j \leq g$. Let f be an algorithm that computes $Y^{\alpha}_{<j}$. We know that $Y^{\alpha}_{<j} = 1$ if and only if $Y^{\alpha}_1 = \cdots = Y^{\alpha}_{j-1} = 1$. To compute each Y^{α}_k for $k \in \{1, 2, \cdots, j-1\}$, the total number of queries to make is $(T + T_{\mathsf{Ver}})$. Thus, the total number of queries to compute $Y^{\alpha}_{<j}$ (or compute f) is at most $(j-1)(T + T_{\mathsf{Ver}}) = (j-1)T^{\mathsf{comb}} < gT^{\mathsf{comb}}$.

Thus, the oracle distribution conditioned on f outputting 1 is a distribution generated in the P-BF-ROM for $P \geq (S + \log \gamma^{-1})T^{\mathsf{comb}}$. Because G is ε-secure in the P-BF-ROM, by definition we have,

$$\Pr\left[Y^{\alpha}_j = 1 \,|\, Y^{\alpha}_{<j} = 1\right] = \Pr_H\left[Y^{\alpha}_j = 1 \,|\, f^H = 1\right] \leq \varepsilon.$$

It holds for all $j \leq g$. □

By Lemma 4, for any advice α, let $S' = S + \log \gamma^{-1}$, we have that

$$\Pr\left[\frac{1}{|R|}\sum_{i \in [R]} X^{\alpha}_i \geq 2\varepsilon\right] \leq 2^{-S'} = 2^{-S} \cdot \gamma.$$

Applying union bound, we have

$$\Pr\left[\exists \alpha \in \{0,1\}^S, \frac{1}{|R|}\sum_{i \in [R]} X^{\alpha}_i \geq 2\varepsilon\right] \leq \gamma.$$

Therefore, we have for any (S, T) algorithm $\mathcal{A}$,

$$\Pr\left[\exists \alpha \in \{0,1\}^S, \mathcal{A}(\alpha, \cdot) \text{ wins the game}\right] \leq 2\varepsilon + \gamma.$$

We finish the proof for the second part.

We then prove the first half of the theorem. If $P < (S + \log \gamma^{-1})T^{\mathsf{comb}}$, the statement is trivially true. Otherwise, let $g = P/T^{\mathsf{comb}}$.

Fix any (S, T) algorithm $\mathcal{A}$ for G. For a given advice $\alpha \in \{0,1\}^S$, we define X^{α}_i, Y^{α}_j and $Y^{\alpha}_{<j}$ as above.

Since G is ε-secure in the P-BF-ROM, similar to Claim 2, we have,

$$\Pr\left[Y^{\alpha}_j = 1 \,|\, Y^{\alpha}_{<j} = 1\right] \leq \varepsilon \text{ for all } j \leq g = P/T^{\mathsf{comb}}.$$

By Lemma 5, for any advice α, let $S' = S + \log \gamma^{-1}$, we have that

$$\Pr\left[\frac{1}{|R|}\sum_{i\in R} X_i^\alpha \geq \varepsilon + S'T^{\mathsf{comb}}/P\right] \leq 2^{-S'} = 2^{-S}\cdot\gamma.$$

Applying union bound, we have

$$\Pr\left[\exists\alpha\in\{0,1\}^S, \frac{1}{|R|}\sum_{i\in R} X_i^\alpha \geq \varepsilon + S'T^{\mathsf{comb}}/P\right] \leq \gamma.$$

Therefore, we have for any (S,T) algorithm $\mathcal{A}$,

$$\Pr\left[\exists\alpha\in\{0,1\}^S, \mathcal{A}(\alpha,\cdot) \text{ wins the game}\right] \leq \varepsilon + \frac{(S+\gamma^{-1})T^{\mathsf{comb}}}{P} + \gamma.$$

□

Note that if we assume the underlying G is secure in the P-BF-RPM, we can prove its security in the AI-RPM with the same parameter.

5 Applications to AI-QROM

In this section, we leverage presampling techniques to the quantum setting, and obtain a presampling lemma for quantum oracles (Theorem 1). To illustrate the power of the presampling techniques, we give the *first* post-quantum non-uniform security bounds for salted Merkle-Damgård hash functions (Theorem 2).

5.1 Presampling Techniques for Quantum Random Oracles

The classical P-BF-ROM is defined by a P-query classical algorithm f. We now extend it to the quantum case. The quantum P-BF-QROM is similarly defined by a P-query quantum algorithm.

Definition 11 (P-BF-QROM). *A security game in the P-BF-QROM consists the following two procedures:*

- *Before the challenging phase, the offline adversary $\mathcal{A}_1$ prepares a quantum algorithm f, and then interacts with a challenger:*
  1. *The challenger samples a random function H;*
  2. *$\mathcal{A}_1$ computes f^H which makes at most P superposition queries to H.*
  3. *$\mathcal{A}_1$ gets a single bit output z of f^H. If $z \neq 1$, it restarts the whole procedure (including sampling a new random function H at the beginning).*
- *In the challenging phase, the security game is executed with an online algorithm $\mathcal{A}_2$ and oracle access to the function H.*

Note that the algorithm f can be inefficient, including running time of f and time for sampling a random H conditioned on $f^H = 1$, except the number of queries are bounded by P.

Equivalently, the definition says that the oracle distribution in the online phase is determined by a P-query bounded quantum algorithm in the pre-computation stage, conditioned on the output of the algorithm f^H being 1.

Note that a random oracle distribution defined by a P-query f outputting 1, can be described by a joint state as in Lemma 1 if the random oracle is simulated as a compressed oracle. This will be useful when we prove security in the P-BF-QROM.

With the definition above, we can lift Theorem 3 to the quantum setting.

Theorem 1. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if a security game G is $\varepsilon(T)$-secure in the P-BF-QROM, then it is (S, T, ε')-secure in the AI-QROM, for some ε' such that,*

$$\varepsilon'(S,T) \leq \varepsilon(T) + \frac{(S + \log \gamma^{-1})T^{\mathsf{comb}}}{P} + \gamma.$$

If G is $\varepsilon(T)$-secure in the P-BF-QROM for $P \geq (S + \log \gamma^{-1})T^{\mathsf{comb}}$, then it is (S, T, ε')-secure in the AI-QROM, for some ε' such that,

$$\varepsilon'(S,T) \leq 2 \cdot \varepsilon(T) + \gamma.$$

$T^{\mathsf{comb}} = T + T_{\mathsf{Ver}}$ is the combined query complexity and T_{Ver} is the query complexity for the challenger .

The proof is identical to that for Theorem 3, except $X_i^\alpha, Y_j^\alpha, Y_{<j}^\alpha$ are defined for a quantum algorithm $\mathcal{A}$ with a classical advice α. Therefore, we omit the proof here.

By replacing H with a random permutation, the definition can be easily extended to P-BF-QRPM. We present a similar presampling theorem for AI-QRPM. More details are provided in Appendix C.

5.2 Post-quantum Non-uniform Security of Merkle-Damgård Hash Functions (MDHF)

Collision resistant hash functions are an important cryptographic primitive. Let H be a (collision resistant) hash function. It is required that finding two distinct inputs $x \neq x'$ such that $H(x) = H(x')$ is hard. However, this definition can not be achieved in the AI-QROM. An attack would simply find a collision in the pre-processing stage and make the security trivial. Thus in practice, one considers a family of collision resistant functions, with a public key called salt that determines which function is chosen. More formally, a hash function is $H : [K] \times [N] \to [M]$ that takes a salt $a \in [K]$ and an input $x \in [N]$. Its collision resistance is defined as, given a uniformly random $a \xleftarrow{\$} [K]$, finding two distinct $x \neq x'$ such that $H(a, x) = H(a, x')$ is hard.

In practice, a hash function usually takes inputs of different lengths. Many hash functions used, including MD5, SHA-2, are based on the Merkle-Damgård construction. It transforms a hash function with fixed input lengths to a hash

function with arbitrary input lengths (as long as the length is still a polynomial). More formally, let H be a collision resistant hash function with fixed input lengths, modeled as a random oracle $H : [M] \times [N] \to [M]$. Note that the salt space $[K]$ is the same as its image $[M]$. Let a message $y = (y_1, \cdots, y_B)$ be a B-block message with each $y_i \in [N]$. The function $H_{\mathsf{MD}}(a, y)$ evaluates as the follows:

$$H_{\mathsf{MD}}(a, y) = H^B_{\mathsf{MD}}(a, (y_1, \cdots, y_B)) = \begin{cases} H(H^{B-1}_{\mathsf{MD}}(a, (y_1, \cdots, y_{B-1})), y_B) & B > 1 \\ H(a, y_1) & B = 1 \end{cases}$$

In other words, it applies the fixed-length hash function H on the salt a and the first block y_1 to get a_2 as the salt for the next step; it then applies H on a_2 and y_2 to get a_3 and so on.

The security game $G_{\mathsf{MDHF}} = (C_{\mathsf{MDHF}})$ is defined as the following, where the challenger C_{MDHF} is specified by these procedures:

- $\mathsf{Samp}^H(r)$: it takes $r \in [M]$ as randomness and outputs a salt $a = r$;
- $\mathsf{Query}^H(a, \cdot) = H(\cdot)$;
- $\mathsf{Ver}^H(a, (x, x')) = 1$ if and only if $x \neq x'$ and $H_{\mathsf{MD}}(a, x) = H_{\mathsf{MD}}(a, x')$.

In other words, an algorithm gets access to the random oracle H in the preprocessing stage; in the online phase, it has the advice computed in the preprocessing stage and is given a random salt a; its goal is to find $x \neq x'$ (either they are of different lengths or they are different inputs of the same length) such that $H_{\mathsf{MD}}(a, x) = H_{\mathsf{MD}}(a, x')$.

In the AI-ROM, a tight bound $\tilde{O}(S/M+T^2/M)$ for the case $B = 1$ was proven by [DGK17]. Later Dodis *et al.* [CDGS18] proved a tight bound $\tilde{O}(ST^2/M)$ for the general MDHF case. More recently, [ACDW20] studied finding short collisions of MDHFs in the AI-ROM. In the rest of the section, we are going to show the first non-trivial bound in the AI-QROM.

We prove the following theorem:

Theorem 2. *G_{MDHF} is $\varepsilon(S, T) = \tilde{O}(ST^3/M)$-secure in the AI-QROM.*

In order to prove the theorem, we show the following lemma. Combining with Theorem 1, we have the first non-trivial bound for the security of MDHF in the AI-QROM.

Lemma 8. *G_{MDHF} is $O((PT^2 + T^3)/M)$-secure in the P-BF-QROM.*

Proof. In the P-BF-QROM, as stated in Lemma 1, the overall state of an algorithm and the oracle conditioned on the measurement of the first P queries is

$$|\psi_0\rangle = \sum_{z, D:|D|\leq P} \alpha_{z,D} |z\rangle |D\rangle .$$

For every salt $a \in [M]$, define a projection Q_a that finds if a is in the database D. In other words,

$$Q_a = \sum_{z,D:\exists x, D(a,x)\neq\perp} |z, D\rangle\langle z, D|.$$

Thus, the probability that a fixed salt a in D is $p_a = |Q_a |\psi_0\rangle|^2$. Since $|\psi_0\rangle$ only has support on all databases D with at most P entries, each z, D will contribute $|\alpha_{z,D}|^2$ to at most P different probabilities p_a. Therefore, if a random challenging salt a is chosen, the probability of a in the database is at most $\mathbb{E}_a[p_a] = \frac{1}{M}\sum_a p_a \leq \frac{P}{M}$.

In the online phase, the algorithm and the challenger are doing the follows:

- The challenger samples a random salt a and gives it to $\mathcal{A}$;
- $\mathcal{A}$ upon receiving a, for $i = 1, \cdots, T$,
  - It applies a unitary U_{i-1} (depends on a), $|\psi'_i\rangle = (U_{i-1} \otimes I)|\psi_{i-1}\rangle$;
  - It makes an oracle query to H (i.e CStO), $|\psi_i\rangle = \mathsf{CStO}|\psi'_i\rangle$.
- $\mathcal{A}$ measures its registers and outputs distinct $\{(x_i, y_i)\}_{i=1}^{B}$ and $\{(x'_i, y'_i)\}_{i=1}^{B'}$. It wins if and only if they form an MDHF collision respect to a: let $y_0 = y'_0 = a$, it should satisfy: (1) $\forall i \in [B], H(y_{i-1}, x_i) = y_i$; (2) $\forall j \in [B'], H(y'_{j-1}, x'_j) = y'_j$; (3) $y_B = y'_{B'}$.

From Lemma 2, let the probability that $\mathcal{A}$ finds an MDHF collision as described above be q_a, the probability that D contains an MDHF collision be q'_a, we have $\sqrt{q_a} \leq \sqrt{q'_a} + \sqrt{\frac{B+B'}{M}}$. Without loss of generality we can assume $B + B' \leq T$, therefore $\sqrt{q_a} \leq \sqrt{q'_a} + \sqrt{\frac{T}{M}}$.

To bound q_a, we only need to focus on the probability q'_a that D contains an MDHF collision. Define R_a be a projection that check if D has an MDHF collision with respect to a. We observe that $|R_a|\psi_0\rangle| \leq |Q_a|\psi_0\rangle|$, because a database contains an MDHF collision with respect to a only if it contains entries starting with a.

First, we know that applying a unitary only on $\mathcal{A}$'s register does not affect the projection R_a:

Lemma 9. *$|R_a|\psi'_i\rangle| = |R_a|\psi_{i-1}\rangle|$ for all $i \in [T]$.*

Proof. By the definition of $|\psi'_i\rangle$, we have $|R_a|\psi'_i\rangle| = |R_a(U_{i-1}\otimes I)|\psi_{i-1}\rangle|$. Since R_a is a projection applied on the second half of the state but U_{i-1} is applied only on the first half of the state, it does not affect the overall probability. Therefore, $|R_a|\psi'_i\rangle| = |R_a|\psi_{i-1}\rangle|$. □

Lemma 10. *$|R_a|\psi_i\rangle| \leq |R_a|\psi'_i\rangle| + \sqrt{\frac{P+i-1}{M}}$ for all $i \in [T]$.*

Proof. By definitions of $|\psi_i\rangle$ and $|\psi'_i\rangle$, we have:

$$\begin{aligned}
|R_a|\psi_i\rangle| &= |R_a U_H|\psi'_i\rangle| \\
&= |R_a U_H (I - R_a + R_a)|\psi'_i\rangle| \\
&\leq |R_a U_H (I - R_a)|\psi'_i\rangle| + |R_a U_H R_a|\psi'_i\rangle|
\end{aligned}$$

The second term $|R_a U_H R_a |\psi'_i\rangle| \leq |U_H R_a |\psi'_i\rangle| = |R_a |\psi'_i\rangle|$. For the first term $|R_a U_H (I - R_a) |\psi'_i\rangle|$, the state $(I - R_a) |\psi'_i\rangle$ has support only on databases D that do not contain any MDHF collision.

We first show the classical statement and translate it into the quantum world using compressed oracle techniques. Classically, by making a new query $(\tilde{a}, \tilde{x})$, the only possibility that an MDHF collision appears in a database D which previously did not contain any MDHF collision is the following case: assume the resulting database contains distinct $\{(x_i, y_i)\}_{i=1}^{B}$ and $\{(x'_i, y'_i)\}_{i=1}^{B'}$ (assuming $y_0 = y'_0 = a$) which form an MDHF collision; the query $(\tilde{a}, \tilde{x})$ must be part of either of $\{(x_i, y_i)\}_{i=1}^{B}$ or $\{(x'_i, y'_i)\}_{i=1}^{B'}$; in other words, there must exist either an $i \in [B]$ or a $j \in [B']$ such that $(\tilde{a}, \tilde{x}, H(\tilde{a}, \tilde{x})) = (y_{i-1}, x_i, y_i)$ or (y'_{j-1}, x'_j, y'_j). Since D previously did not contain an MDHF collision, $(\tilde{a}, \tilde{x})$ was not in the database. Since D only contains $|D|$ entries and $H(\tilde{a}, \tilde{x})$ is completely random (by classical lazy sampling argument), the probability of having an MDHF collision is at most $|D|/M$.

Quantumly, the above analysis can be applied using compressed oracle technique. The same arguments have been applied in analyzing finding pre-images [Zha19,CGLQ20] and collision-finding [Zha19,LZ19,CGLQ20]. Using this statement, $|R_a U_H (I - R_a) |\psi'_i\rangle|$ is at most $\sqrt{\frac{P+i-1}{M}}$ where $(P + i - 1)$ is the largest cardinality of all databases with non-zero support. □

Therefore, combining it with the above lemmas, we conclude that:

$$|R_a |\psi_T\rangle| \leq |R_a |\psi_{T-1}\rangle| + \sqrt{\frac{P+T-1}{M}} \leq \sum_{i=1}^{T} \sqrt{\frac{P+i-1}{M}} + |R_a |\psi_0\rangle|.$$

By Lemma 2, we have,

$$\begin{aligned}
\sqrt{q_a} &\leq \sqrt{q'_a} + \sqrt{T/M} \\
&= |R_a |\psi_T\rangle| + \sqrt{T/M} \\
&\leq \sum_{i=1}^{T} \sqrt{\frac{P+i-1}{M}} + |R_a |\psi_0\rangle| + \sqrt{T/M} \\
&\leq \sum_{i=1}^{T} \sqrt{\frac{P+i-1}{M}} + |Q_a |\psi_0\rangle| + \sqrt{T/M}.
\end{aligned}$$

By Cauchy-Schwarz,

$$q_a \leq O\left((T + PT^2 + T^3)/M\right) + 2 \cdot |Q_a |\psi_0\rangle|^2 = O\left((PT^2 + T^3)/M\right) + 2p_a.$$

Averaging over a, $\mathbb{E}_a[q_a] \leq O\left((PT^2 + T^3)/M\right) + \mathbb{E}_a[p_a] = O\left((PT^2 + T^3)/M\right)$. Thus MDHF is $O\left(\frac{PT^2+T^3}{M}\right)$-secure in the P-BF-QROM. □

5.3 Post-quantum Non-uniform Security of One-way Functions (OWF)

In this section, we show the simplicity and generality of our theorem by reproving results in [CGLQ20]. We only prove one of the main results in [CGLQ20], namely the almost optimal bound of OWF in the AI-QROM. Other results can be reproved with almost no extra effort, in a similar way.

The security game $G_{\mathsf{OWF}} = (C_{\mathsf{OWF}})$ is defined as follows, where the challenger C_{OWF} is specified by the following procedures:

- $\mathsf{Samp}^H(r)$: which takes randomness $r = x \in [N]$ and outputs the challenge $\mathsf{ch} = y = H(x)$.
- $\mathsf{Query}^H(r, x')$: it ignores the randomness and simply outputs $H(x')$.
- $\mathsf{Ver}^H(r, x')$: it outputs 1 if and only if $H(x') = H(x)$ where $x = r$.

Namely, the challenger samples a random input x and the challenge is $y = H(x)$. An adversary wins the game if and only if it finds any preimage of y.

We reprove the following theorem.

Theorem 7. *G_{OWF} is $\varepsilon(S,T) = \tilde{O}((ST+T^2)/\min\{N,M\})$-secure in the AI-QROM.*

By Theorem 1, we only need to prove its security in the P-BF-QROM.

Lemma 11. *G_{OWF} is $O((P+T^2)/\min\{N,M\})$-secure in the P-BF-QROM.*

Proof. To prove it, we first recall the lemma 1.5 in [CGLQ20]. Note that although the original statement for the following lemma only considers H having the same domain and range, it indeed works for any $H : [N] \to [M]$ and the proofs are in Lemma 5.6 and Lemma 5.9 of [CGLQ20].

Lemma 12 (Lemma 1.5, [CGLQ20]). *For any quantum algorithm making q_0+q queries to a random function $H : [N] \to [M]$, if $H(x)$ is sampled and given after the q_0-th query, conditioned on arbitrary outcomes (with non-zero probability) of the algorithm's measurement during the first q_0 queries, the probability of inverting $H(x)$ is at most $O((q_0+q^2)/\min\{N,M\})$.*

By letting the computation for the first q_0 queries to be an evaluation of f and measuring if $f^H = 1$, we realize it is exactly the statement for its security in the q_0-BF-QROM. By letting $q_0 = P$ and $q = T$, we prove our lemma. □

References

[AA11] Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. In *Proc. of Innovations in Theoretical Computer Science Conference (ITCS)*, pages 338–352, 2011.

[Aar05a] Scott Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1(1):1–28, 2005.

[Aar05b] Scott Aaronson. Ten semi-grand challenges for quantum computing theory. http://www.scottaaronson.com/writings/qchallenge.html, 2005.

[Aar10] Scott Aaronson. Updated version of "ten semi-grand challenges for quantum computing theory". http://www.scottaaronson.com/blog/?p=471, 2010.

[ACDW20] Akshima, David Cash, Andrew Drucker, and Hoeteck Wee. Time-space tradeoffs and short collisions in merkle-damgård hash functions. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 157–186. Springer, 2020.

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

[Bac12] Arturs Backurs. Influences in low-degree polynomials. https://www.scottaaronson.com/showcase2/report/arturs-backurs.pdf, 2012.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.

[CDG18] Sandro Coretti, Yevgeniy Dodis, and Siyao Guo. Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In *Annual International Cryptology Conference*, pages 693–721. Springer, 2018.

[CDGS18] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. Random oracles and non-uniformity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 227–258. Springer, 2018.

[CGLQ20] Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. Tight quantum time-space tradeoffs for function inversion. *arXiv preprint arXiv:2006.05650*, 2020.

[CK18] Henry Corrigan-Gibbs and Dmitry Kogan. The discrete-logarithm problem with preprocessing. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 415–447. Springer, 2018.

[CLMP13] Kai-Min Chung, Huijia Lin, Mohammad Mahmoody, and Rafael Pass. On the power of nonuniformity in proofs of security. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 389–400, 2013.

[CLQ19] Kai-Min Chung, Tai-Ning Liao, and Luowen Qian. Lower bounds for function inversion with quantum advice. *arXiv preprint arXiv:1911.09176*, 2019.

[DFKO06] Irit Dinur, Ehud Friedgut, Guy Kindler, and Ryan O'Donnell. On the fourier tails of bounded functions over the discrete cube. In *Proc. of Symposium on Theory of Computing (STOC)*, pages 437–446, 2006.

[DGK17] Yevgeniy Dodis, Siyao Guo, and Jonathan Katz. Fixing cracks in the concrete: Random oracles with auxiliary input, revisited. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 473–495, 2017.

[DMP17] Andreas Defant, Mieczysław Mastyło, and Antonio Pérez. On the fourier spectrum of functions on boolean cubes. *Mathematische Annalen*, (3), 2017.

[DTT10] Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and prgs. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, pages 649–665, 2010.

[GLM+16] Mika Goos, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, 2016.

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 305–313, 2000.

[Hel80] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Trans. Information Theory*, 26(4):401–406, 1980.

[HXY19] Minki Hhan, Keita Xagawa, and Takashi Yamakawa. Quantum random oracle model with auxiliary input. In *AsiaCrypt*. Springer, 2019.

[KK19] Nathan Keller and Ohad Klein. Quantum speedups need structure. *CoRR*, abs/1911.03748, 2019.

[LZ19] Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 189–218. Springer, 2019.

[LZ20] Qian Li and Jiapeng Zhang. Small ℓ_1-norm extension of functions over the boolean cube, and aaronson-ambainis conjecture. private communication, 2020.

[MA12] Montanaro and Ashley. Some applications of hypercontractive inequalities in quantum information theory. *Journal of Mathematical Physics*, 53(12), 2012.

[Mid04] Gatis Midrijanis. Exact quantum query complexity for total boolean functions. 2004.

[NABT15] Aran Nayebi, Scott Aaronson, Aleksandrs Belovs, and Luca Trevisan. Quantum lower bound for inverting a permutation with advice. *Quantum Information & Computation*, 15(11&12):901–913, 2015.

[NS94] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994.

[OSSS05] Ryan O'Donnell, Michael E. Saks, Oded Schramm, and Rocco A. Servedio. Every decision tree has an influential variable. In *Proc. of Symposium on Foundations of Computer Science (FOCS)*, pages 31–39, 2005.

[OY16] Ryan O'Donnell and Zhao Yu. Polynomial bounds for decoupling, with applications. In *Proc. of Conference on Computational Complexity (CCC)*, 2016.

[Unr07] Dominique Unruh. Random oracles and auxiliary input. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 205–223. Springer, 2007.

[Wee05] Hoeteck Wee. On obfuscating point functions. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 523–532, 2005.

[Yao90] AC-C Yao. Coherent functions and program checkers. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 84–94, 1990.

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

A More on Quantum Random Oracle Model

This section is taken verbatim from Section 2.3 of [CGLQ20]. In this subsection, we recall the technique introduced by Zhandry [Zha19]. We will explain how to purify a random oracle in the quantum setting first, and then give equivalent forms of a quantum random oracle, namely standard oracle StO, phase oracle PhO and compressed standard oracle CStO. All these oracles are equivalent in the sense that for every (even unbounded) algorithm making queries to one of these oracles, the output distribution of the algorithm is exactly identical regardless of which oracle is given. Then Zhandry shows dealing with compressed standard oracle is usually easier. Roughly speaking, Zhandry shows that with compressed standard oracle, one could quantify the amount of the information about the random oracle learned by any quantum algorithm, analogous to the lazy sampling technique that is very commonly used for classical random oracles.

Note that in Zhandry's work [Zha19], they originally only considered output of size $M = 2^m$, and implementing a quantum random oracle as $U_H : |x\rangle |u\rangle = |x\rangle |u \oplus H(x)\rangle$ where $\oplus$ is bit-wise XOR, or the addition over $\mathbb{F}_2^m$. Therefore, their description of the compressed oracle technique is different since the range is defined as $\mathbb{F}_2^m$ instead of $\mathbb{Z}/M\mathbb{Z}$ considered in this paper. Two oracles are equivalent as we can simulate one with the other using two queries. For the completeness of the paper, we will reprove some of the useful lemmas under the integer ring $\mathbb{Z}/M\mathbb{Z}$.

Also note that Zhandry also showed that a compressed oracle can be efficiently implemented by a quantum computer, i.e. the running time is only polynomial in the number of queries and $\log N, \log M$. In this work, since we mainly consider query complexity and for presentation, we ignore the issue of efficiency for a simpler presentation.

Purification: standard oracle. Let H be a random oracle $[N] \to [M]$. The function H is sampled at the very beginning, or equivalently, initially we prepare a maximally mixed state $\eta \sum_H |H\rangle \langle H|$ up to some normalization factor η, and

each query can be implemented by another unitary U, which reads the function H and applies U_H. However, we can "purify" the random oracle, meaning that we can replace the mixed state of $|H\rangle$ with a uniform superposition of all possible functions, i.e. $\sqrt{\eta}\sum_H |H\rangle$. Consider the truth table of H, that is $|H\rangle = |H(1)\rangle |H(2)\rangle \cdots |H(N)\rangle$. Let $\mathcal{A}$ be any quantum algorithm. We say the algorithm can query the standard oracle if we treat the algorithm's registers and $|H\rangle$ as a whole system, initialized as $|0\rangle |\psi\rangle \otimes \frac{1}{M^{N/2}} \sum_H |H\rangle$. An oracle query StO in this purified state is defined as,

$$\mathsf{StO} |x\rangle |u\rangle |z\rangle \otimes |H\rangle = |x\rangle |u + H(x)\rangle |z\rangle \otimes |H\rangle ,$$

where $|x\rangle, |u\rangle$ are the input and output register, $|z\rangle$ is an arbitrary working register and $|H\rangle$ is the random oracle. Each local quantum computation is $U_i \otimes I$ which only operates on $\mathcal{A}$'s registers $|x\rangle |u\rangle |z\rangle$. Therefore the computation of any $\mathcal{A}$ can be described as a sequence of: $U_0 \otimes I$, StO, $\cdots$, $U_{T-1} \otimes I$, StO, $U_T \otimes I$, and a final computational measurement over $\mathcal{A}$'s register. The following proposition tells that the output distribution using a standard oracle is exactly the same as using a random oracle.

Lemma 13 ([Zha19, Lemma 2]). *Let $\mathcal{A}$ be an (unbounded) quantum algorithm making oracle queries. The output of $\mathcal{A}$ given a random function H is exactly identical to the output of $\mathcal{A}$ given access to a standard oracle. Therefore, a random oracle with quantum query access can be perfectly simulated as a standard oracle.*

Phase kickback: phase oracle. Define a unitary V as $(I_x \otimes \mathsf{QFT}_M^\dagger \otimes I_H)$ which applies $\mathsf{QFT}_M^\dagger$ on the output register $|u\rangle$. Define the phase oracle operator $\mathsf{PhO} := V^\dagger \cdot \mathsf{StO} \cdot V$.

$$\begin{aligned}
\mathsf{PhO} |x\rangle |u\rangle \otimes |H\rangle &= V^\dagger \cdot \mathsf{StO} \cdot \frac{1}{\sqrt{M}} \sum_y \omega_M^{-uy} |x\rangle |y\rangle \otimes |H\rangle \\
&= V^\dagger \cdot \frac{1}{\sqrt{M}} \sum_y \omega_M^{-uy} |x\rangle |y + H(x)\rangle \otimes |H\rangle \\
&= \frac{1}{M} \sum_{y,y'} \omega_M^{-uy+(y+H(x))y'} |x\rangle |y'\rangle \otimes |H\rangle \\
&= \frac{1}{M} \omega_M^{uH(x)} \sum_{y,y'} \omega_M^{(y+H(x))(y'-u)} |x\rangle |y'\rangle \otimes |H\rangle \\
&= |x\rangle |u\rangle \otimes \omega_M^{uH(x)} |H\rangle .
\end{aligned}$$

Similarly, we override the notation PhO such that for any auxiliary register $|z\rangle$, $\mathsf{PhO} |x\rangle |u\rangle |z\rangle \otimes |H\rangle = |x\rangle |u\rangle |z\rangle \otimes \omega_M^{uH(x)} |H\rangle$.

Observing that $VV^\dagger = I$, the following lemma tells that we can efficiently convert between a standard oracle algorithm and a phase oracle algorithm.

Lemma 14 ([Zha19, Lemma 3]). *Let $\mathcal{A}$ be an (unbounded) quantum algorithm making queries to a standard oracle. Let $\mathcal{B}$ be the algorithm that is identical to $\mathcal{A}$, except it performs V and $V^\dagger$ before and after each query. Then the output distributions of $\mathcal{A}$ (given access to a standard oracle) and $\mathcal{B}$ (given access to a phase oracle) are identical. Therefore, a quantum random oracle can be perfectly simulated as a phase oracle.*

We then have the following lemma for the phase oracle, that formulates the behavior of a quantum algorithm making at most T queries to the phase oracle. We have seen that every PhO query will add a phase to the $|H\rangle$ register, i.e., $\mathsf{PhO}\,|x\rangle\,|u\rangle \otimes |H\rangle = |x\rangle\,|u\rangle \otimes \omega_M^{uH(x)}\,|H\rangle$. Define D as a truth table, or equivalently a vector in $(\mathbb{Z}/M\mathbb{Z})^N$ and $D(x)$ be the x-th entry of D. Define $|D|$ be the number of non-zero entries in D. For any D, we define $|\phi_D\rangle = \frac{1}{M^{N/2}} \sum_H \omega_M^{\langle D,H\rangle}\,|H\rangle$ for all $D \in (\mathbb{Z}/M\mathbb{Z})^N$ where $\langle D, H\rangle$ is defined to be the inner product $\sum_{x\in[N]} D(x)H(x)$. Note that we will only use this inner product on the exponent of ω_M so it is irrelevant whether we are computing it on the integer ring or the ring modulo M.

Lemma 15. *Let $\mathcal{A}$ be a quantum algorithm making at most T queries to a phase oracle. The overall state of $\mathcal{A}$ and the phase oracle can be written as $\sum_{z,D:|D|\le T} \alpha_{z,D}\,|z\rangle \otimes \frac{1}{M^{N/2}} \sum_H \omega_M^{\langle D,H\rangle}\,|H\rangle = \sum_{z,D:|D|\le T} \alpha_{z,D}\,|z\rangle \otimes |\phi_D\rangle$.*

Moreover, it is true even if the state is conditioned on arbitrary outcomes (with non-zero probability) of $\mathcal{A}$'s intermediate measurements.

Compressed standard oracle. Intuitively, compressed oracle is an analogy of classical lazy sampling method. Instead of recording all the information of H in the registers (like what it does in the standard oracle or the phase oracle), Zhandry provides a better solution which is useful to argue the amount of the information an algorithm knows about the random oracle.

The oracle register records a database/list that contains the output on each input x, the output is an element in $\mathbb{Z}/M\mathbb{Z} \cup \{\perp\}$, where $\perp$ is a special symbol denoting that the value is "uninitialized". The database is initialized as an empty list D_0 of length N, in other words, it is initialized as the pure state $|\emptyset\rangle := |\perp, \perp, \cdots, \perp\rangle$. Let $|D|$ denote the number of entries in D that are not $\perp$. Define $D(x)$ to be the x-th entry.

For any D and x such that $D(x) = \perp$, we define $D \cup (x, u)$ to be the database D', such that for every $x' \neq x$, $D'(x') = D(x)$ and at the input x, $D'(x) = u$.

The compressed standard oracle is the unitary $\mathsf{CStO} := \mathsf{StdDecomp} \circ \mathsf{CStO}' \circ \mathsf{StdDecomp}$, where

- $\mathsf{CStO}'\,|x, u\rangle|D\rangle = |x, u + D(x)\rangle|D\rangle$ when $D(x) \neq \perp$, which writes the output of x defined in D to the u register. This operator will never be applied on an x, D where $D(x) = \perp$.
- $\mathsf{StdDecomp}(|x\rangle \otimes |D\rangle) := |x\rangle \otimes \mathsf{StdDecomp}_x\,|D\rangle$, where $\mathsf{StdDecomp}_x\,|D\rangle$ works on the x-th register of the database $D(x)$. Intuitively, it swaps a uniform superposition $\frac{1}{\sqrt{M}} \sum_y |y\rangle$ with $|\perp\rangle$ on the x-th register. Formally,

- If $D(x) = \perp$, $\mathsf{StdDecomp}_x$ maps $|\perp\rangle$ to $\frac{1}{\sqrt{M}}\sum_y |y\rangle$, or equivalently, $\mathsf{StdDecomp}_x|D\rangle = \frac{1}{\sqrt{M}}\sum_y |D \cup (x,y)\rangle$. Intuitively, if the database does not contain information about x, it samples a fresh y as the output of x.
- If $D(x) \neq \perp$, $\mathsf{StdDecomp}_x$ works on the x-th register, and it is an identity on $\frac{1}{\sqrt{M}}\sum_y \omega_M^{uy}|y\rangle$ for all $u \neq 0$; it maps the uniform superposition $\frac{1}{\sqrt{M}}\sum_y |y\rangle$ to $|\perp\rangle$.
  More formally, for a D' such that $D'(x) = \perp$,

  $$\mathsf{StdDecomp}_x \frac{1}{\sqrt{M}}\sum_y \omega_M^{uy}|D' \cup (x,y)\rangle = \frac{1}{\sqrt{M}}\sum_y \omega_M^{uy}|D' \cup (x,y)\rangle \text{ for any } u \neq 0,$$

  and,

  $$\mathsf{StdDecomp}_x \frac{1}{\sqrt{M}}\sum_y |D' \cup (x,y)\rangle = |D'\rangle.$$

  Since all $\frac{1}{\sqrt{M}}\sum_y \omega_M^{uy}|y\rangle$ and $|\perp\rangle$ form a basis, these requirements define a unique unitary operation.

Zhandry proves that, StO and CStO are perfectly indistinguishable by any *unbounded* quantum algorithm.

Lemma 16 ([Zha19, Lemma 4]). *Let $\mathcal{A}$ be an (unbounded) quantum algorithm making oracle queries. The output of $\mathcal{A}$ given access to the standard oracle is exactly identical to the output of $\mathcal{A}$ given access to a compressed standard oracle.*

Combining this lemma with Lemma 13, we obtain the following corollary.

Corollary 1. *A quantum random oracle can be perfectly simulated as a compressed standard oracle.*

In this work, we only consider query complexity, and thus simulation efficiency is irrelevant to us. Looking ahead, we simulate a random oracle as a compressed standard oracle to help us analyze security of different games with the help from the following lemmas.

The first lemma gives a general formulation of the overall state of $\mathcal{A}$ and the compressed standard oracle after $\mathcal{A}$ makes T queries, analogous to Lemma 15 for phase oracle.

Lemma 1. *If $\mathcal{A}$ makes at most T queries to a compressed standard oracle, assuming the overall state of $\mathcal{A}$ and the compressed standard oracle is $\sum_{z,D} \alpha_{z,D}|z\rangle_{\mathcal{A}}|D\rangle_H$ where $|z\rangle$ is $\mathcal{A}$'s registers and $|D\rangle$ is the oracle's registers, then it only has support on all D such that $|D| \leq T$. In other words, the overall state can be written as,*

$$\sum_{z,D:|D|\leq T} \alpha_{z,D}|z\rangle_{\mathcal{A}} \otimes |D\rangle_H.$$

Moreover, it is true even if the state is conditioned on arbitrary outcomes (with non-zero probability) of $\mathcal{A}$'s intermediate measurements.

The second lemma provides a quantum analogue of lazy sampling in the classical ROM.

Lemma 2 (Lemma 5 in [Zha19]). *Let H be a random oracle from $[N] \to [M]$. Consider a quantum algorithm $\mathcal{A}$ making queries to the standard oracle and outputting tuples $(x_1, \cdots, x_k, y_1, \cdots, y_k, z)$. Supposed the random function H is measured after $\mathcal{A}$ produces its output. Let R be an arbitrary set of such tuples. Suppose with probability p, $\mathcal{A}$ outputs a tuple such that (1) the tuple is in R and (2) $H(x_i) = y_i$ for all i. Now consider running $\mathcal{A}$ with the compressed standard oracle CStO, and suppose the database D is measured after $\mathcal{A}$ produces its output. Let p' be the probability that (1) the tuple is in R and (2) $D(x_i) = y_i$ (in particular, $D(x_i) \neq \perp$) for all i. Then $\sqrt{p} \leq \sqrt{p'} + \sqrt{k/M}$.*

Moreover, it is true even if it is conditioned on arbitrary outcomes (with non-zero probability) of $\mathcal{A}$'s intermediate measurements.

B Equivalence between Conjecture 1 and Conjecture 3

We first present some notations and basic facts in Boolean function analysis that will be used. Let $f : \{-1,1\}^n \to \mathbb{R}$ be a function, we make the following notations.

1. We define the zero vector as $\mathbf{0} := (0, \ldots, 0) \in \mathbb{R}^n$.
2. The expectation of f is defined as $f(\mathbf{0}) := \mathbb{E}_{x \sim \{-1,1\}^n}[f(x)]$
3. The variance of f is defined as $\mathbf{Var}[f] := \mathbb{E}_x\left[(f(x) - f(\mathbf{0}))^2\right]$.
4. The degree of f is defined as $\deg(f) = \max\{|S| : \widehat{f}(S) \neq 0\}$. Here $\widehat{f}(S) := \mathbb{E}_x[f(x)\prod_{i \in S} x_i]$ represents the Fourier coefficient of f.
5. For every set $J \subseteq [n]$, we denote its complement as $\bar{J} := [n] \setminus J$.
6. For each $J \subseteq [n]$ and $x'_J \in \{-1,1\}^J$, the restricted function $f|_{x'_J} : \{-1,1\}^{\bar{J}} \to \mathbb{R}$ is defined as $f|_{x'_J}(x_{\bar{J}}) = f(x'_J, x_{\bar{J}})$. Besides, we denote as $f(x_J, \mathbf{0}_{\bar{J}}) = \mathbb{E}_{x_{\bar{J}}}[f(x_J, x_{\bar{J}})]$.
7. For every $x \in \{-1,1\}^n$ and $J \subset [n]$, let $x^J \in \{-1,1\}^n$ denote the string obtained by flipping all bits in J.

Definition 12 (Smooth function). *Let t be a positive integer and $\delta > 0$. A function $f : \{-1,1\}^n \to \mathbb{R}$ is called (t,δ)-smooth if for any $|J| \leq t$ and $x_J \in \{-1,1\}^J$, it has that*

$$|f(x_J, \mathbf{0}_{\bar{J}}) - f(\mathbf{0})| \leq \delta.$$

Definition 13. *Let $f : \{-1,1\}^n \to \mathbb{R}$ be a function. For each $x \in \{-1,1\}^n$, the block-sensitivity of f at x is defined as*

$$\mathrm{bs}(f,x) = \max_{k, J_1 \sqcup \cdots \sqcup J_k} \sum_{i \in [k]} |f(x^{J_i}) - f(x)|/2.$$

The block-sensitivity of f is defined as $\max_x\{\mathrm{bs}(f,x)\}$.

Nisan and Szegedy [NS94] proved that $\mathrm{bs}(f) = O(\deg(f)^2)$ for any Boolean function $f : \{-1,1\}^n \to \{0,1\}$. This result can be further generalized to bounded functions, see [KK19] for a formal proof.

Claim 3 ([NS94,KK19]). *Let $f : \{-1,1\}^n \to [0,1]$ be a bounded degree-d polynomial, then $\mathrm{bs}(f) = O(d^2)$.*

Let $f : \{-1,1\}^n \to [0,1]$ be the acceptance probability of some quantum algorithm that makes T queries to a Boolean string $x = (x_1, \cdots, x_n)$, a basic result shown by Beals et al. [BBC$^+$01] says that $\deg(f) \leq 2T$. Furthermore, according to Claim 3, we have $\mathrm{bs}(f) = O(T^2)$.

It is more convenient to use the following restates of Conjecture 1 and Conjecture 3.

Conjecture 5 (restate of Conjecture 1). There is a constant $C > 0$. For the acceptance probability $f : \{-1,1\}^n \to [0,1]$ of any quantum algorithm that makes T queries to a Boolean string $x = (x_1, \cdots, x_n)$ and any $\varepsilon > 0$, there is a decision tree $g : \{-1,1\}^n \to [0,1]$ of depth $(T/\varepsilon)^C$ such that $\|g - f\|_2 \leq \varepsilon$.

The equivalence between Conjecture 1 and Conjecture 5 is based on the fact that $\|g-f\|_2^2 \leq \|g-f\|_1 \leq \|g-f\|_2$ as $\|g-f\|_\infty \leq 1$.

Conjecture 6 (restated of Conjecture 3). There is a constant $C > 0$. For the acceptance probability $f : \{-1,1\}^n \to [0,1]$ of any quantum algorithm that makes T queries to a Boolean string $x = (x_1, \cdots, x_n)$ and any $\delta > 0$, if f is $\big((T/\delta)^C, (\delta/T)^C\big)$-smooth, then $\mathbf{Var}[f] \leq \delta$.

The following claim will be used.

Claim 4. *Let $J \subset [n]$ be any set. Then*

$$\mathop{\mathbb{E}}_{x_J}\left[\mathop{\mathbb{E}}_{x_{\bar{J}}}\left[\mathrm{bs}(f|_{x_J}, x_{\bar{J}})\right]\right] \leq \mathop{\mathbb{E}}_{x}[\mathrm{bs}(f,x)] - \max_{x_J}\left\{|f(x_J, \mathbf{0}_{\bar{J}}) - f(\mathbf{0})|/4\right\}.$$

Proof. Let $r := \max_{x_J}\{|f(x_J, \mathbf{0}_{\bar{J}}) - f(\mathbf{0})|\}$ and let x_J^* be the input reaching it. Since $f(\mathbf{0}) = \mathbb{E}_{x_J}[f(x_J, \mathbf{0}_{\bar{J}})]$, there is a string $x'_J \in \{-1,1\}^J$ such that $|f(x_J^*, \mathbf{0}_{\bar{J}}) - f(x'_J, \mathbf{0}_{\bar{J}})| \geq r$. Hence

$$\begin{aligned} r \leq & \Big|\mathop{\mathbb{E}}_{x_{\bar{J}}}\left[f(x_J^*, x_{\bar{J}}) - f(x'_J, x_{\bar{J}})\right]\Big| \\ \leq & \mathop{\mathbb{E}}_{x_J, x_{\bar{J}}}\left[|f(x_J^*, x_{\bar{J}}) - f(x_J, x_{\bar{J}})| + |f(x'_J, x_{\bar{J}}) - f(x_J, x_{\bar{J}})|\right] \\ \leq & 2 \cdot \mathop{\mathbb{E}}_{x_J, x_{\bar{J}}}\left[\max_{z_J \in \{-1,1\}^J}\left\{|f(z_J, x_{\bar{J}}) - f(x_J, x_{\bar{J}})|\right\}\right]. \end{aligned}$$

On the other hand, according to the definition of block-sensitivity, it has

$$\mathrm{bs}(f|_{x_J}, x_{\bar{J}}) + \max_{z_J \in \{-1,1\}^J}\left\{|f(z_J, x_{\bar{J}}) - f(x_J, x_{\bar{J}})|/2\right\} \leq \mathrm{bs}(f, x).$$

Putting these together, we conclude that

$$\begin{aligned}\mathop{\mathbb{E}}_{x_J,x_{\bar{J}}}[\mathrm{bs}(f|_{x_J},x_{\bar{J}})] \leq & \mathop{\mathbb{E}}_{x}[\mathrm{bs}(f,x)] - \mathop{\mathbb{E}}_{x_J,x_{\bar{J}}}\left[\max_{z_J\in\{-1,1\}^J}\{|f(z_J,x_{\bar{J}})-f(x_J,x_{\bar{J}})|/2\}\right] \\ \leq & \mathop{\mathbb{E}}_{x}[\mathrm{bs}(f,x)] - r/4.\end{aligned}$$

Theorem 8. *Conjecture 6 implies Conjecture 5.*

Proof. Assuming Conjecture 6 holds, and let C_0 be the constant from this conjecture. For the acceptance probability $f : \{-1,1\}^n \to [0,1]$ of any quantum algorithm that makes T queries to a Boolean string $x = (x_1, \cdots, x_n)$ and any $\varepsilon > 0$, we construct a small depth decision tree g that ε-approximates f as follows: given any function f, we continually query non-smooth set, a set J of size $(T/\varepsilon)^{C_0}$ such that $|f(x_J, \mathbf{0}_{\bar{J}}) - f(\mathbf{0})| \geq (\varepsilon/T)^{C_0}$, until reaches a small variance function. We formalize the idea as follows:

1. Define $f_0 = f$ and $B := c_1 \cdot T^{C_0+2}/\varepsilon^{C_0+1}$, where c_1 is a large constant to be determined later.
2. Given a function $f_i : \{-1,1\}^{S_i} \to [0,1]$, if $\mathbf{Var}[f_i] \leq \varepsilon$ or $i \geq B$, then output $T(x) = \mathbb{E}[f_i]$;
3. Otherwise, since f_i is a restricted function of f, f_i is also the acceptance probability of some quantum algorithm that makes at most T queries. Then Conjecture 6 guarantees the existence of J of size at most $(T/\varepsilon)^{C_0}$ and $x_J^* \in \{-1,1\}^J$ s.t. $|f_i(x_J^*, \mathbf{0}_{S_i\setminus J}) - f_i(\mathbf{0})| \geq (\varepsilon/T)^{C_0}$. Deterministically choose such a J, query x_J and define $f_{i+1} : \{-1,1\}^{S_i\setminus J} \to [0,1]$ by $f_{i+1}(y) = f_i(x_J, y)$. Repeat Stage 2 with f_{i+1}.

Obviously the depth of g is at most $B \cdot (T/\varepsilon)^{C_0} = c_1 \cdot T^{2C_0+2}/\varepsilon^{2C_0+1}$. What remains is to show that g approximates f, i.e., $\|f-g\|_2^2 \leq 2\varepsilon$. Let I be a random variable counting the number of iterations on a random input x. Then

$$\begin{aligned}\|f-g\|_2^2 = & \mathbb{E}_x\left[(f_I(x) - \mathbb{E}(f_I))^2\right] = \mathbb{E}_x[\mathbf{Var}\left[f_I\right]] \\ = & \Pr\left[I < B\right] \cdot \mathbb{E}_x[\mathbf{Var}\left[f_I\right] | I < B] + \Pr[I = B] \cdot \mathbb{E}_x[\mathbf{Var}\left[f_I\right] | I = B] \\ \leq & \mathbb{E}_x[\mathbf{Var}\left[f_I\right] | I < B] + \Pr\left[I = B\right] \\ \leq & \varepsilon + \Pr\left[I = B\right],\end{aligned}$$

where the last inequality is due to that $\mathbf{Var}[f_I] \leq \varepsilon$ if $I < B$. In the following, we show that $\Pr(I = B) \leq \varepsilon$, and then finish the proof.

By contradiction, suppose $\Pr[I = B] > \varepsilon$. For convenience, we pretend that the algorithm iterates exactly B times on every input x, in the way that for any $I < i \leq B$, the algorithm queries nothing. Fix any $i < B$, let $J \subset [n]$ (and f_i resp.) be the random set queried (and the random resulted subfunction resp.) in

the i-th iteration on a random input x. Note that $f_{i+1} = f_i|_{x_J}$, then according to Claim 4, we have

$$\begin{aligned}
&\mathbb{E}_x[\mathrm{bs}(f_{i+1}, x)] \\
=&\mathbb{E}_{f_i}\mathbb{E}_x[\mathrm{bs}(f_{i+1}, x)|f_i] = \mathbb{E}_{f_i}\mathbb{E}_x[\mathrm{bs}(f_i|_{x_J}, x)|f_i] \\
\leq&\mathbb{E}_{f_i}\mathbb{E}_x[\mathrm{bs}(f_i, x)|f_i] - \frac{1}{4}\cdot\mathbb{E}_{f_i}\mathbb{E}_x\left[\max_{x_J\in\{-1,1\}^J}\left\{|f_i(x_J, \mathbf{0}_{S_i\setminus J}) - f(\mathbf{0}_{S_i})|\right\}\Big|f_i\right] \\
=&\mathbb{E}_x[\mathrm{bs}(f_i, x)] - \frac{1}{4}\cdot\mathbb{E}_x\left[\max_{x_J\in\{-1,1\}^J}\left\{|f_i(x_J, \mathbf{0}_{S_i\setminus J}) - f(\mathbf{0}_{S_i})|\right\}\right] \\
\leq&\mathbb{E}_x[\mathrm{bs}(f_i, x)] - \frac{1}{4}\cdot\Pr(I > i)\cdot\mathbb{E}_x\left[\max_{x_J\in\{-1,1\}^J}\left\{|f_i(x_J, \mathbf{0}_{S_i\setminus J}) - f(\mathbf{0}_{S_i})|\right\}\Big|I > i\right] \\
\leq&\mathbb{E}_x[\mathrm{bs}(f_i, x)] - \frac{1}{4}\cdot\varepsilon^{C_0+1}/T^{C_0}.
\end{aligned}$$

By induction, we have that $\mathbb{E}_x[\mathrm{bs}(f_i, x)] \leq \mathbb{E}_x[\mathrm{bs}(f_0, x)] - \frac{i}{4}\cdot\varepsilon^{C_0+1}/T^{C_0}$. Particularly,

$$\begin{aligned}
\mathbb{E}_x[\mathrm{bs}(f_B, x)] \leq&\mathbb{E}_x[\mathrm{bs}(f_0, x)] - \frac{B}{4}\cdot\varepsilon^{C_0+1}/T^{C_0} \\
\leq&\mathbb{E}_x[\mathrm{bs}(f_0, x)] - \frac{c_1}{4}\cdot T^2 < 0,
\end{aligned}$$

if c_1 is sufficiently large. A contradiction.

We then prove the other direction.

Theorem 9. *Conjecture 5 implies Conjecture 6.*

Proof. Assuming Conjecture 5 holds, and let C_0 be the constant. For the acceptance probability $f : \{-1,1\}^n \to [0,1]$ of any quantum algorithm that makes T queries to a Boolean string $x = (x_1, \cdots, x_n)$, we show there is a set $J \subset [n]$ of size $|J| \leq (T/\mathbf{Var}[f])^{C_0}$ and a string $x_J \in \{-1,1\}^J$ such that

$$|f(x_J, \mathbf{0}_{\bar{J}}) - f(\mathbf{0})| \geq \mathbf{Var}\left[f\right]/4.$$

Let $\varepsilon = \mathbf{Var}[f]/4$, then by Conjecture 5, there is a decision tree g of depth $(T/\varepsilon)^{C_0}$ such that $\|f - g\|_2 \leq \varepsilon$. For each $x \in \{-1,1\}^n$, let $Q_x \subset [n]$ denote the set of variables queried by g on input x. Notice that,

$$\mathbb{E}_x[|f(x) - f(x_{Q_x}, \mathbf{0}_{\bar{Q}_x})|] \leq \mathbb{E}_x[|f(x) - g(x)|] = \|f - g\|_1 \leq \|f - g\|_2 \leq \mathbf{Var}\left[f\right]/4.$$

Since $|f(x) - f(\mathbf{0})| \leq 1$,

$$\mathbb{E}_x\left[|f(x) - f(\mathbf{0})|\right] \geq \mathbb{E}_x\left[|f(x) - f(\mathbf{0})|^2\right] = \mathbf{Var}\left[f\right].$$

By combining them, we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_x\left[|f(x_{Q_x}, \mathbf{0}_{\bar{Q}_x}) - f(\mathbf{0})|\right] \geq&\mathop{\mathbb{E}}_x\left[|f(x) - f(\mathbf{0})|\right] - \mathop{\mathbb{E}}_x\left[|f(x_{Q_x}, \mathbf{0}_{\bar{Q}_x}) - f(x)|\right] \\
\geq&3\,\mathbf{Var}\left[f\right]/4.
\end{aligned}$$

Since the set Q_x has size at most $(T/\varepsilon)^{C_0}$ for every input x. There is a pair (x, Q_x) that certifies Conjecture 6.

C Applications to AI-QRPM

In this section, we define P-BF-QRPM (quantum random permutation model) and present the presampling theorem for quantum random permutations. Similar to P-BF-QROM, P-BF-QRPM is defined by a P-query quantum algorithm f.

Definition 14 (P-BF-QRPM). *A security game in the P-BF-QRPM consists the following two procedures:*

- *Before the challenging phase, the offline adversary $\mathcal{A}_1$ prepares a quantum algorithm f, and then interacts with a challenger:*
  1. *The challenger samples a random permutation H;*
  2. *$\mathcal{A}_1$ computes f^H which makes at most P superposition queries to H.*
  3. *$\mathcal{A}_1$ gets a single bit output z of f^H. If $z \neq 1$, it restarts the whole procedure (including sampling a new random permutation H at the beginning).*
- *In the challenging phase, the security game is executed with an online algorithm $\mathcal{A}_2$ and oracle access to the function H.*

Note that the algorithm f can be inefficient, including running time of f and time for sampling a random H conditioned on $f^H = 1$, except the number of queries are bounded by P.

Equivalently, the definition says that the permutation distribution in the online phase is determined by a P-query bounded quantum algorithm in the pre-computation stage, conditioned on the output of the algorithm f^H being 1.

With the definition above, we obtain the following theorem.

Theorem 10. *For any $P \in \mathbb{N}$ and every $\gamma > 0$, if a security game G is $\varepsilon(T)$-secure in the P-BF-QRPM, then it is (S, T, ε')-secure in the AI-QRPM, for some ε' such that,*

$$\varepsilon'(S,T) \leq \varepsilon(T) + \frac{(S + \log \gamma^{-1})T^{\mathsf{comb}}}{P} + \gamma.$$

If G is $\varepsilon(T)$-secure in the P-BF-QRPM for $P \geq (S + \log \gamma^{-1})T^{\mathsf{comb}}$, then it is (S, T, ε')-secure in the AI-QRPM, for some ε' such that,

$$\varepsilon'(S,T) \leq 2 \cdot \varepsilon(T) + \gamma.$$

$T^{\mathsf{comb}} = T + T_{\mathsf{Ver}}$ is the combined query complexity and T_{Ver} is the query complexity for the challenger .

One major question raised by [CGLQ20] is the optimal security of OWF in the AI-QRPM. They prove an almost optimal bound for OWF in the AI-QROM (reproved in Section 5.3 of this work), but their tools can not handle AI-QRPM. We give a sufficient condition for achieving the optimal security bound of OWF in the AI-QRPM, which is the conjecture below.

Conjecture 7. G_{OWF} is $O((P + T^2)/N)$-secure in the P-BF-QRPM.

If we can prove the conjecture, an optimal security bound of OWF in the AI-QRPM can be achieved, following from Theorem 10.

Theorem 11. *G_{OWF} is $\varepsilon(S,T) = \tilde{O}((ST + T^2)/N)$-secure in the AI-QRPM.*