Unifying Presampling via Concentration Bounds

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

[1] New York University Shanghai, Shanghai, China
siyao.guo@nyu.edu
[2] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
liqian@ict.ac.cn
[3] Simons Institute for the Theory of Computing, Berkeley, CA, USA
qipengliu0@gmail.com
[4] University of Southern California, CA, USA
jiapengz@usc.edu

Abstract. Auxiliary-input (AI) idealized models, such as auxiliary-input random oracle model (AI-ROM) and auxiliary-input random permutation model (AI-PRM), play a critical role in assessing *non-uniform security* of symmetric key and hash function constructions. However, obtaining security bounds in these models is often much more challenging. The presampling technique, initially introduced by Unruh (CRYPTO' 07) for AI-ROM and later exported to several other models by Coretti et al. (EUROCRYPT' 18). It generically reduces security proofs in AI models to much simpler bit-fixing (BF) models, making it much easier to obtain concrete bounds in AI models. As a result, the presampling technique has leads to simpler proofs for many known bounds (e.g. one-way functions), and has been applied to many settings where the compression technique appears intractable (e.g., Merkle-Damgård hashing).

We study the possibility of 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 to 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 in the classical setting, including AI-ROM and AI-PRM. Our theorem matches the best-known security loss and unifies previous presampling techniques by Coretti et al. (EUROCRYPT' 18) and Coretti et al. (CRYPTO' 18).
- Finally, we leverage our new classical presampling techniques to a novel "quantum bit-fixing" version of presampling. It matches the optimal security loss of the classical presampling. Using our techniques, we give the *first* post-quantum non-uniform security for salted Merkle-Damgård hash functions and reprove the tight non-uniform security for function inversion by Chung et al. (FOCS' 20).

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), ideal-cipher model (ICM). Since most constructions of block ciphers and hash functions lack solid theoretical foundations, security bounds in idealized models provide an essential (heuristic) justification and guidelines for their security in the standard model.

However, traditional idealized models fail to capture preprocessing attacks. The obtained bounds in idealized models are inaccurate or not applicable at all once preprocessing is allowed. For example, Hellman [**?**] showed a preprocessing attack that takes S bits of advice and makes T queries to a permutation over $[N]$, can invert a random element with probability roughly ST/N. [5] Hence, a permutation cannot be one-way against attacks beyond $S = T = N^{1/2}$. However, it is easy to derive in RPM that an image of 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 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 are 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) [**?**,**?**,**?**,**?**]. In AI models, an attacker can obtain arbitrary S bits of leakage about the idealized primitive before attacking the system, then make additional T queries to the primitive. Similar to 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 ways to prove security bounds. For example, it is not straightforward how we should analyze inverting a random permutation over $[N]$ 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. Specifically for permutation inversion, an optimal ST/N bound was first proved [**?**] via the "compression paradigm", as introduced by Yao [**?**], Gennaro and Trevisan [**?**] (and later adopted by [**?**]). The main idea is to argue that if an attacker succeeds with "high probability" in inverting a random permutation, 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. The compression paradigm is a general technique that can be applied to different problems in auxiliary-input models.

[5] For simplicity, we ignore big O or $\tilde{O}$ notations in the introduction.

The compression paradigm has been successfully applied to AI-ROM by Dodis et al. [?], and auxiliary-input Generic Group Model (AI-GGM) by Corrigan-Gibbs and Kogan [?]. While compression-based proofs often lead to optimal bounds, they are usually quite laborious. For every cryptographic construction, we need to carefully examine the property of the construction together with its security definition to compress the idealized primitive.

The presampling technique. Coretti et al. [?] give a simple and intuitive proof for permutation inversion by adapting the "presampling" approach taken by Coretti et al. [?] (first introduced in [?]) in the ROM. The presampling technique can be viewed as a general reduction from AI models to a much simpler bit-fixing (BF) model. In the BF model, an oracle is arbitrarily fixed on at most P coordinates chosen by the attacker and the remaining coordinates are chosen at random and independently of the fixed coordinates. Notably, the online attacker only knows the fixed coordinates. The BF model is easy to work with, because most proof techniques for idealized models can be applied as long as we avoid these fixed coordinates.

Specifically, Coretti et al. [?] and Coretti et al. [?] show that any attack with S-bit advice and T oracle queries in AI-ROM/RPM/ICM/GGM will have similar advantages 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. [?]). For unpredictability applications (such as one-way functions), additive loss such as ST/P is not preferable. They show that one can set P to rough ST and achieve a multiplicative factor of 2 in the exact security.

These previous works result in a general way for proving security in AI models. For a cryptographic application in AI-model, we can 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 roughly $2 \cdot \varepsilon(S, T, ST)$, i.e., twice its security in the (ST)-BF model. As an example, in the (ST)-BF-RPM, it can be shown that a random image of a random permutation f over $[N]$ is invertible with probability at most $O(ST/N)$ [6] which immediately gives the optimal $O(ST/N)$ bound (matching Hellman's attack) in AI-RPM.

The presampling technique offers a more straightforward approach for proving security bounds in AI models than the compression technique. By presampling techniques, Coretti et al. [?] and Coretti et al. [?], reprove the AI-ROM/RPM/GGM security bounds obtained by the compression technique [?,?,?], and give the first non-uniform bounds for many practical applications (in which compression appears intractable).

We remark that *the optimal additive loss* and *multiplicative version* of presampling techniques in [?,?] are *crucial* for obtaining exact (tight) bounds. As

[6] If the challenge $f(x)$ does not come from the fixed coordinates, then a proof by standard techniques bounds the probability of finding $f(x)$ 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 $O(ST/N)$.

shown by Dodis et al. [?], the presampling technique by Unruh [?] with additive 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 suboptimal bounds for unpredictable applications, such as $\sqrt{ST/N}$ security 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 do 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 quantum random oracle model (QROM) [?], 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 cryptography and hash functions, quantum versions of AI models have been proposed and studied [?,?,?,?], in which the adversary is allowed to obtain S-(qu)bit precomputed advice about the idealized primitive.

By leveraging classical compression proofs, [?,?,?] obtain many non-uniform security bounds. However, they only manage to analyze 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 question we study in this paper:

Can we leverage presampling techniques to the quantum setting?

Specifically, we hope to reduce the AI quantum models to more straightforward "BF quantum models", then export similar proofs from quantum idealized models.

Recently, Chung et al. [?] gave a new technique for analyzing AI models with quantum adversaries. This technique reduces (Q)AI security[7] against attackers with (quantum) advice to analyzing multi-instance (MI) security against attackers *without* advice. They use this technique to prove the tight bound $ST/N + T^2/N$ for inverting random functions in the AI-QROM model. Although the new approach 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 game has an exponential decay in the number of instances. For practical symmetric-key and hash-function constructions, proving such statements may be challenging. By 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 presampling theorem of Coretti et al. [?] for AI-ROM. In this work, we first show

[7] Here, QAI allows quantum states as advice.

that such direct leveraging is difficult, which will resolve a major open problem in quantum computing [**?**]. In light of the barrier, we revisit the classical presampling techniques and give a simpler and unified proof for the classical presampling theorems. Finally, following the new classical proof, we give the first quantum presampling theorem and several non-uniform lower bounds as applications.

Barriers for leveraging presampling to the quantum setting. In Section 3, we show that such leveraging has a technical barrier: it will resolve a major open problem in quantum computing [**?**], which asserts that any quantum algorithm can be approximated on most inputs by an efficient classical algorithm[8]. This open problem, dating back to (according to [**?**]) 1999 or earlier, was included twice in Aaronson's list of "ten semi-grand challenges for quantum computing theory" [**?**,**?**].

In [**?**], Aaronson and Ambainis proposed an approach, which became well-known as the Aaronson-Ambainis conjecture, towards this open problem via Boolean function analysis. Specifically, 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 [**?**,**?**,**?**,**?**,**?**], this open problem and the closely related Aaronson-Ambainis conjecture seem still quite open. They are proven only for some class of functions [**?**,**?**,**?**]. The best-known bound for general functions is exponentially far from conjectured [**?**,**?**,**?**].

Remark 1. Note that the barrier does not contradict our quantum presampling theorem. Direct leveraging will give us a better presampling theorem than ours, which pre-fixes at most P coordinates classically. Whereas, our presampling theorem requires to pre-fix P coordinates "quantumly".

Ideally, we would like to show a statement similar to classical presampling: AI-QROM can be reduced to BF-QROM, where the random oracle is fixed *classically* on at most P coordinates. However, what we obtain in this work is (informally): AI-QROM can be reduced to BF-QROM, where the random oracle is fixed "*quantumly*" on at most P coordinates. We will show in the following paragraph why this ideal presampling statement is better than the presampling statement obtained in this work. Our first contribution points out a barrier to prove the above ideal version (with connections to AA conjecture). In light of the barrier, we present our quantum presampling theorem.

If the ideal presampling holds, we can get the lower bound of function inversion in the AI-QROM easily, without using any involved techniques. Because either the challenge image is in one of the fixed coordinates (with probability ST/N), or it is outside the fixed coordinates, in which we argue the success probability by simply using the existing lower bound of Grover's search. This will give an much easier proof for the lower bound of function inversion in the AI-QROM, which is $ST/N + T^2/N$, reproves the result by Chung et al.

[8] It will be only polynomially slower than the quantum algorithm in terms of query complexity

Unifying presampling via concentration bounds. Faced 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 [**?**] and RPM [**?**], using an equivalent characterization of P-BF-ROM/P-BF-RPM.

Instead of viewing P-BF-ROM as a random function with at most P prefixed inputs/outputs, we give an equivalent formulation with respect to a classical *randomized* algorithm f making at most P queries. The security game is then under the oracle access to the function H, where H is given by rejection sampling a fully random oracle H, but conditional on $f^H = 1$. This definition naturally extends to P-BF-RPM by rejection sampling a random permutation H.

We show a unified proof for the classical presampling theorems with the alternative definition and basic concentration bounds. The proof is much simpler than the original proof [**?**], as the original proof needs to first decompose a random oracle distribution with advice into dense distributions (a technique used in the area of communication complexity [**?**]), 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 [**?**]. Note that our proof achieves optimal bounds, as it matches the optimal bounds in [**?**].

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 our 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 $\varepsilon'(S,T)$-secure in the AI-QROM, where*

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

In particular, if G is $\varepsilon(T)$-secure in the P-BF-QROM for $P \geq (S + \log\gamma^{-1})T^{\mathsf{comb}}$, then it is $\varepsilon'(S,T)$-secure in the AI-QROM, where

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

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

Note that it is optimal in the sense that it matches the optimal classical presampling theorem by Coretti et al. [**?**].

Therefore, to obtain security in the AI-QROM, it is sufficient to obtain its security in the P-BF-QROM. We use Zhandry's compressed oracle [?] in the P-BF-QROM, and present the first non-trivial security analysis of (salted) Merkle-Damg̊ard Hash Functions (MDHF) in the AI-QROM.

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

Here, G_{MDHF} denotes the security game of MDHF (See Section 5.2).

In the classical setting, Coretti et al. [?] show an attack with advantage $\Omega(ST^2/M)$ (which is optimal), and Akshima et al. [?] show an attack for 2-block MDHF with advantage $\Omega((ST + T^2)/M)$. We observe that the attack by Akshima et al. [?] 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. [?] 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.

Finally, to show the simplicity and generality of our quantum presampling technique, we additionally reprove that function inversion has security $O((ST + T^2)/N)$ in the AI-QROM [?] (See Section 5.3).

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 [?], by boosting the success probability and applying Gentle Measurement Lemma [?]. However, we suspect that the resulting statement may not be optimal.

Bit-Fixing Security of Random 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 [?] 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 review the relevant literature on the quantum random oracle model.

2.1 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 [**?**]. This section is taken verbatim from Section 2.2 of [**?**].

An oracle-aided quantum algorithm can perform quantum computation as well as quantum oracle queries. 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 the 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.

2.2 Compressed Oracle

Here we briefly recall some backgrounds about compressed oracle techniques, which was first introduced in [**?**]. More details are provided in the full version.

Intuitively, compressed oracle is an analogy of the 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. Such an implementation of a random oracle is inefficient, and security games based on such an implementation are usually hard to analyze. Therefore, instead of recording all the information of H in the registers, Zhandry provides a solution to argue the amount of 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 $|\emptyset\rangle :=$

$|\bot, \bot, \cdots, \bot\rangle$. Let $|D|$ denote the number of entries in D that are not $\bot$. Define $D(x)$ to be the x-th entry. Intuitively, $D(x)$ can be seen as the output of the oracle on x if $D(x) \neq \bot$; otherwise, the oracle's output on x is still undetermined.

For any D and x such that $D(x) = \bot$, 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 $D(x) \neq \bot$, which writes the output of x defined in D to the u register. This operator will never be applied on an x, D where $D(x) = \bot$.
- $\mathsf{StdDecomp}(|x\rangle \otimes |D\rangle) := |x\rangle \otimes \mathsf{StdDecomp}_x \,|D\rangle$, 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 $|\bot\rangle$ on the x-th register. Formally,
  - If $D(x) = \bot$, $\mathsf{StdDecomp}_x$ maps $|\bot\rangle$ 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 $D(x) \neq \bot$, $\mathsf{StdDecomp}_x$ 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 $|\bot\rangle$.

    More formally, for a D' such that $D'(x) = \bot$,
    $$\mathsf{StdDecomp}_x \frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |D' \cup (x, y)\rangle = \frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |D' \cup (x, y)\rangle \text{ for any } u \neq 0,$$
    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 $|\bot\rangle$ 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 [**?**,**?**].

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]). *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 $(x_1, \cdots, x_k, y_1, \cdots, y_k, z)$. Suppose 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, $D(x_i) \neq \bot$) 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.

2.3 Security Game with Classical Advice

In this paper, we 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 [**?**].

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 tell if $\mathcal{A}_1$ is classical or quantum because it is unbounded. We say $\mathcal{A}$ is quantum if $\mathcal{A}_2$ makes quantum queries to $\tilde{H}$ 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 2.4.

Below, we will use the words “adversary” and “algorithm” interchangeably.

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*[9].
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 an adversary is supposed to produce a binary* $\mathsf{ans} \in \{0, 1\}$.

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

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.*

2.4 Presampling Techniques for Random Oracles

We recall classical presampling techniques for random oracles from [**?**].

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

Since any oracle $\mathcal{O} : [N] \to [M]$ can be represented by a string in $[M]^N$, we also 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.*

Coretti et al. [**?**] then defined security in the P-BF-ROM.

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

- *Before the challenge 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 challenge phase, the security game (see Definition 2) 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 pre-fixed inputs/outputs are $\mathcal{L}$.*

Remark 2. Note that $\mathcal{A}_2$ knows the strategy of $\mathcal{A}_1$. In [**?**], 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 [**?**]. 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,*

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

and

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

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 [**?**].

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 $\varepsilon'(S, T)$-secure in the AI-ROM, where*

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

In particular, if G is $\varepsilon(T)$-secure in the P-BF-ROM for $P \geq (S + \log\gamma^{-1})T^{\mathsf{comb}}$, then it is $\varepsilon'(S,T)$-secure in the AI-ROM, where

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

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

Built upon the above theorems, [**?**] proved 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.

2.5 Aaronson-Ambainis Conjecture

A major open problem in quantum computing is whether polynomial quantum speedups need the input to be "structured"–that is, the domain includes only inputs that satisfy a stringent promise. This question is formalized as the following conjecture.

Conjecture 1 (folklore, see [?]). 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 [**?**,**?**]. In the paper [**?**], Aaronson and Ambainis proposed a new conjecture (a.k.a Aaronson-Ambainis conjecture) which is sufficient to affirm Conjecture 1. Specifically, they conjectured that any low-degree function $f : \{-1,1\}^n \to [0,1]$ has an influential variable.

Conjecture 2 ([?]). 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 an $i \in [n]$ such that

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

Despite much effort [**?**,**?**,**?**,**?**,**?**], both Conjecture 1 and Conjecture 2 are still quite open , and they are proven only for some class of functions [**?**,**?**,**?**]. The best known bound for general functions is still exponentially far from conjectured [**?**,**?**,**?**].

The paper [**?**] implicitly provided an equivalent form of Conjecture 1, which seems easier to prove and will be used in this paper. 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 $\{0,1\}^N$.

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 full version. The nontrivial direction is to show how Conjecture 3 implies Conjecture 1. It follows the general strategy of the argument of Midrijanis [**?**] 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.

2.6 Concentration Bounds

The following claim and lemmas of concentration bounds will be used in our proof. We prove them in this section. The following proof uses the same idea as Theorem 3.1 in [**?**].

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*

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

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 2.4, 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"[10]. 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,

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

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

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 2.4 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.

Theorem 4. *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

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

So $\Pr_X\left[\left|\mathcal{A}^X - \mathbb{E}[\mathcal{A}]\right| \geq \sigma/2\right] \geq 3\sigma^2/4$. 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 [**?**] and AI-RPM (random permutation model) [**?**]. 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'.

To overcome the barrier in the quantum setting, we first realize that the classical definition P-BF-ROM can be defined by a bounded query algorithm. We find this equivalent definition is much easier to work with and is helpful for generalizing to the quantum setting.

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

- *Before the challenge 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 challenge 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. Later the security game will be executed under oracle access to H. 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 6. *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, which is a uniform distribution over all oracles that are compatible with $\mathcal{L}$ (also taken the randomness of $\mathcal{L}$).

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.

For a transcript τ and an oracle H, we say they are compatible if for every $(x, y) \in \tau$, $H(x) = y$. Fix any transcript τ, let X_τ be the oracle distribution that is a *uniform distribution* over all oracles that are compatible with τ. Thus, conditioned on f producing transcript τ, the oracle will have distribution X_τ.

Note that every pair of transcripts τ, τ' (produced by f) is 'disjoint'. Namely, for any τ, τ', there always exists an input x and $y \neq y'$ such that $(x, y) \in \tau$ and $(x, y') \in \tau'$. Then X_τ and $X_{\tau'}$ have disjoint support. We further notice that the support of X_τ for all τ 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 the support of $\{X_\tau\}_\tau$ 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 $\mathcal{L} = \tau$.

In other words, the distribution generated by g is a bit-fixing source 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 [?], 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 [?]), 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 $\varepsilon'(S,T)$-secure in the AI-ROM, where*

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

In particular, if G is $\varepsilon(T)$-secure in the P-BF-ROM for $P \geq (S + \log \gamma^{-1})T^{\mathsf{comb}}$, then it is $\varepsilon'(S,T)$-secure in the AI-ROM, where

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

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

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 for the j-th instance, with a uniformly chosen challenge.

We also define $Y_{<j}^\alpha$ in a similar way in Section 2.6: it is 1 if and only if all $Y_1^\alpha = \cdots = Y_{j-1}^\alpha = 1$. $Y_{<j}^\alpha$ is the random variable indicating if an algorithm $\mathcal{A}(\alpha)$ wins all games in the first $(j-1)$ instances, with uniformly chosen challenges for each instance.

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. *For all $j \leq g := (S + \log \gamma^{-1})$,*

$$\Pr\left[Y_j^\alpha = 1 \,|\, Y_{<j}^\alpha = 1\right] \leq \varepsilon.$$

Proof. Fixing a $j \leq g$. Let f be an algorithm that computes $Y_{<j}^\alpha$. We know that $Y_{<j}^\alpha = 1$ if and only if $Y_1^\alpha = \cdots = Y_{j-1}^\alpha = 1$. To compute each Y_k^α 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 $Y_{<j}^\alpha$ (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,

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

It holds for all $j \leq g$. □

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

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

Applying union bound, we have

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

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_i^α, Y_j^α and $Y_{<j}^\alpha$ as above.

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

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

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 theorem 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 of the following two procedures:*

- *Before the challenge 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 challenge 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 precomputation 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 $\varepsilon'(S,T)$-secure in the AI-QROM, where*

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

In particular, if G is $\varepsilon(T)$-secure in the P-BF-QROM for $P \geq (S + \log \gamma^{-1}) T^{\mathsf{comb}}$, then it is $\varepsilon'(S,T)$-secure in the AI-QROM, where

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

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

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 the full version.

5.2 Post-quantum Non-uniform Security of Merkle-Damg̊ard 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 $a \overset{\$}{\leftarrow} [K]$, 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-Damg̊ard 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.

Definition 12 (G_{MDHF}). *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 $H_{\mathsf{MD}}(a, x) = H_{\mathsf{MD}}(a, x')$.*

Recall the definition of a security game is defined in Section 2.3. In other words, an algorithm gets access to the random oracle H in the pre-processing stage; in the online phase, it has the advice computed in the pre-processing 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 [**?**]. Later Dodis *et al.* [**?**] proved a tight bound $\tilde{O}(ST^2/M)$ for the general MDHF case. More recently, [**?**] 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 7. *G_{MDHF} is $O((PT^2+T^3)/M)$-secure in the P-BF-QROM.*

Proof. To prove this lemma, we assume a random oracle is implemented as a compressed standard oracle, which is identical to a truly random oracle from the adversary's view.

In the P-BF-QROM, the oracle distribution in the challenge phase is a uniform random oracle distribution conditioned on a P-query quantum algorithm f outputting 1. As stated in Lemma 1, the overall state of the algorithm f and the oracle conditioned on the measurement of the first P queries:

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

where Z register is the state of the algorithm f and D register is the state for compressed standard oracle.

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 $\mathbb{E}_a[p_a] = \frac{1}{M}\sum_a p_a \le \frac{P}{M}$.

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 $\sqrt{q_a} \le \sqrt{q_a'} + \sqrt{\frac{B+B'}{M}}$. Without loss of generality we can assume $B + B' \le T$, therefore $\sqrt{q_a} \le \sqrt{q_a'} + \sqrt{\frac{T}{M}}$.

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 $|R_a\,|\psi_0\rangle\,| \le |Q_a\,|\psi_0\rangle\,|$, 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 8. $|R_a \left|\psi_i'\right\rangle| = |R_a \left|\psi_{i-1}\right\rangle|$ *for all* $i \in [T]$.

Proof. By the definition of $\left|\psi_i'\right\rangle$, we have $|R_a \left|\psi_i'\right\rangle| = |R_a(U_{i-1} \otimes I) \left|\psi_{i-1}\right\rangle|$. 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, $|R_a \left|\psi_i'\right\rangle| = |R_a \left|\psi_{i-1}\right\rangle|$. □

Lemma 9. $|R_a \left|\psi_i\right\rangle| \le |R_a \left|\psi_i'\right\rangle| + 3\sqrt{2} \cdot \sqrt{\frac{P+i-1}{M}}$ *for all* $i \in [T]$.

Proof. We first give the following claim. Let D be a database that does not contain an MDHF collision and $x = (\tilde{a}||\tilde{x}) \in [M] \times [N]$ be a query not in D. Define $G_{x,D}$ be the set of images $y \in [M]$ such that $D \cup \{(x, y)\}$ contains an MDHF collision.

Claim 3. *For any database D that does not contain an MDHF collision and a query $x = (\tilde{a}||\tilde{x})$, $|G_{x,D}| \le |D|$.*

We give the proof for the above claim. By making the query, 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'). Thus, the necessary condition to form an MDHF collision is that the image $H(\tilde{a}, \tilde{x})$ is already in the database. We conclude that $|G_{x,D}| \le |D|$.

We prove our main lemma for compressed phase oracle CPhO. The same argument holds for compressed standard oracle CStO since they are equivalent. The proof follows the same structure of the proof for Theorem 1 in [?].

First recall that $\mathsf{CPhO} = \mathsf{StdDecomp} \circ \mathsf{CPhO}' \circ \mathsf{StdDecomp}$. CPhO' is defined as follows: $\mathsf{CPhO}' \left|x, y, z, D\right\rangle = \omega_M^{yD(x)} \left|x, y, z, D\right\rangle$. Here D has range $[M]$, and $y \cdot \perp$ is defined as 0.

We define R_a as the projection on databases that contain an MDHF collision starting with salt a. By definitions of $\left|\psi_i'\right\rangle$ and $\left|\psi_i\right\rangle$ (they are states before or after making the i-th queries to a random oracle), we have:

$$|R_a \left|\psi_i\right\rangle| = |R_a \mathsf{CPhO} \left|\psi_i'\right\rangle| .$$

Without loss of generality, we assume the state $\left|\psi_i'\right\rangle$ is the following:

$$\left|\psi_i'\right\rangle = \sum_{x,y,z,D:|D| \le P+i-1} \alpha_{x,y,z,D} \left|x, y, z\right\rangle \otimes \left|D\right\rangle .$$

Here x is the input registers, y is the output registers, z is the algorithm's private registers, and D is the registers for compressed phase oracle. Moreover, it only has non-zero weight over D such that $|D| \le P + i - 1$. This is because $\left|\phi_0\right\rangle$ has support over D whose cardinality is at most P and each query to CPhO only increases the size by at most 1.

We then define three more projections on all registers:

- W_a: it projects to the following space: (1) D *does NOT* contain an MDHF collision starting with salt a; (2) $y \neq 0$; (3) $D(x) = \perp$.
- W_a': it projects to the following space: (1) (2) in W_a, but $D(x) \neq \perp$.
- W_a'': (1) in W_a, but $y = 0$.

It is easy to see that $R_a + W_a + W_a' + W_a'' = I$ (in the proof for Theorem 1 of [**?**], they used the notations P, Q, R, S respectively; since we have already used some of these letters, we choose a set of different notations).

Therefore, we have:

$$\begin{aligned}|R_a \mathsf{CPhO}\,|\psi_i'\rangle\,| =&|R_a \mathsf{CPhO}\,(R_a + W_a + W_a' + W_a'')\,|\psi_i'\rangle\,| \\ \leq&|R_a \mathsf{CPhO}\,R_a\,|\psi_i'\rangle\,| + |R_a \mathsf{CPhO}\,W_a\,|\psi_i'\rangle\,| \\ &+ |R_a \mathsf{CPhO}\,W_a'\,|\psi_i'\rangle\,| + |R_a \mathsf{CPhO}\,W_a''\,|\psi_i'\rangle\,|.\end{aligned}$$

By triangle inequality, we can bound them separately:

Part 1. $|R_a \mathsf{CPhO}\,R_a\,|\psi_i'\rangle| \leq |\mathsf{CPhO}\,R_a\,|\psi_i'\rangle| = |R_a\,|\psi_i'\rangle|$.

It is an easy case because removing the first projection R_a does not increase the norm. The second equality is simply because unitary CPhO does not change the norm.

Part 2. $|R_a \mathsf{CPhO}\,W_a\,|\psi_i'\rangle\,| \leq \sqrt{\frac{P+i-1}{M}}|W_a\,|\psi_i'\rangle\,|$.

Since it is in the space defined by W_a, the database does not contain an MDHF collision for salt a, and the queried point x is not in the database. We have $W_a\,|\psi_i'\rangle$ is the following:

$$W_a\,|\psi_i'\rangle = \sum_{\substack{D \in \overline{\mathcal{D}_{\mathsf{MDHF}}} : |D| \leq P+i-1 \\ x \notin D, y \neq 0, z}} \alpha_{x,y,z,D}\,|x,y,z\rangle \otimes |D\rangle\,.$$

Here $\overline{\mathcal{D}_{\mathsf{MDHF}}}$ is the set of databases that do not contain an MDHF collision. By making the next query, we have,

$$\mathsf{CPhO}\,W_a\,|\psi_i'\rangle = \sum_{\substack{D \in \overline{\mathcal{D}_{\mathsf{MDHF}}} : |D| \leq P+i-1 \\ x \notin D, y \neq 0, z}} \alpha_{x,y,z,D}\,|x,y,z\rangle \otimes \left(\frac{1}{\sqrt{M}} \sum_{w \in [M]} \omega_M^{wy}\,|D \cup (x,w)\rangle\right).$$

Intuitively, a uniformly random output of x will be sampled.

Since D does not contain an MDHF collision, applying R_a to the state $\mathsf{CPhO}\,W_a\,|\psi_i'\rangle$ will force $w \in G_{x,D}$ (see the definition of $G_{x,D}$ at the beginning of this proof). Formally,

$$\begin{aligned}R_a \mathsf{CPhO}\,W_a\,|\psi_i'\rangle = &\sum_{\substack{D \in \overline{\mathcal{D}_{\mathsf{MDHF}}} : |D| \leq P+i-1 \\ x \notin D, y \neq 0, z}} \alpha_{x,y,z,D}\,|x,y,z\rangle \\ &\otimes \left(\frac{1}{\sqrt{M}} \sum_{w \in G_{x,D}} \omega_M^{wy}\,|D \cup (x,w)\rangle\right).\end{aligned}$$

Notice that the images of the different basis states are orthogonal, and $|G_{x,D}| \le |D| \le (P+i-1)$. We conclude that $|R_a \mathsf{CPhO}\, W_a \left|\psi_i'\right\rangle| \le \sqrt{\frac{P+i-1}{M}} |W_a \left|\psi_i'\right\rangle|$.

Part 3. $|R_a \mathsf{CPhO}\, W_a' \left|\psi_i'\right\rangle| \le 3\sqrt{2} \cdot \sqrt{\frac{P+i-1}{M}} |W_a' \left|\psi_i'\right\rangle|$.

For basis states $|x,y,z\rangle \otimes |D\rangle$ that is in the space defined by W_a', we know that D does not contain an MDHF collision for salt a, y is not 0 and $D(x)$ is not $\perp$. Let w be $D(x)$. Let D' be the database with x removed. Then by some algebraic manipulations (the same tricks in the proof of Theorem 1 in [**?**]), we have $\mathsf{CPhO}\, |x,y,z\rangle \otimes |D \cup (x,w)\rangle$ is:

$$|x,y,z\rangle \otimes \Bigg(\omega_M^{wy} \left(|D' \cup (x,w)\rangle + \frac{1}{\sqrt{M}} |D'\rangle \right) \\ + \frac{1}{M} \sum_{y'} \left(1 - \omega_M^{wy} - \omega_M^{y'y} \right) |D' \cup (x,y')\rangle \Bigg).$$

First, both $D' \cup (x,w)$ and D' do not contain an MDHF collision. This is by the assumption that the basis states are in the space defined by W_a'. We write $W_a' \left|\psi_i'\right\rangle$ as:

$$W_a' \left|\psi_i'\right\rangle = \sum_{x,y,z,D',w} \beta_{x,y,z,D',w} \, |x,y,z\rangle \otimes |D' \cup (x,w)\rangle \,.$$

Thus, $|R_a \mathsf{CPhO}\, W_a' \left|\psi_i'\right\rangle|^2$ can be bounded as:

$$\begin{aligned} |R_a \mathsf{CPhO}\, W_a' \left|\psi_i'\right\rangle|^2 &= \frac{1}{M^2} \sum_{\substack{x,y,z,D' \\ y' \in G_{x,D}}} \left| \sum_w \beta_{x,y,z,D',w} (1 - \omega_M^{wy} - \omega_M^{y'y}) \right|^2 \\ &\le \frac{9}{M} \sum_{\substack{x,y,z,D' \\ y' \in G_{x,D}}} |\beta_{x,y,z,D',w}|^2 \\ &= \frac{9(P+i-1)}{M} |W_a' \left|\psi_i'\right\rangle|^2. \end{aligned}$$

The inequality on the second line follows by Cauchy-Schwarz inequality. The conclusion of Part 3 follows by taking square root on both sides of the above inequality.

Part 4. $|R_a \mathsf{CPhO}\, W_a'' \left|\psi_i'\right\rangle| = 0$.

It is also an easy case. Because for all basis states $|x,y,z\rangle \otimes |D\rangle$ in the space defined by W_a'', $y = 0$ and D does not contain an MDHF collision. When $y = 0$, D will not get updated after the next oracle query. Thus the conclusion follows.

Finally, we combine all the statements above:

$$\begin{aligned}|R_a \mathsf{CPhO} |\psi'_i\rangle| \leq& |R_a \mathsf{CPhO}\, R_a |\psi'_i\rangle| + |R_a \mathsf{CPhO}\, W_a |\psi'_i\rangle| \\ &+ |R_a \mathsf{CPhO}\, W'_a |\psi'_i\rangle| + |R_a \mathsf{CPhO}\, W''_a |\psi'_i\rangle| \\ \leq& |R_a |\psi'_i\rangle| + 3\sqrt{\frac{P+i-1}{M}}\left(|W_a |\psi'_i\rangle| + |W'_a |\psi'_i\rangle|\right) \\ \leq& |R_a |\psi'_i\rangle| + 3\sqrt{2}\cdot\sqrt{\frac{P+i-1}{M}}.\end{aligned}$$

The last equality follows by $|W_a |\psi'_i\rangle|^2 + |W'_a |\psi'_i\rangle|^2 \leq 1$ and Cauchy-Schwarz inequality. This concludes the proof of the lemma. □

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

$$|R_a |\psi_T\rangle| \leq 3\sqrt{2}\cdot\sum_{i=1}^{T}\sqrt{\frac{P+i-1}{M}} + |R_a |\psi_0\rangle|.$$

By Lemma 2, we have,

$$\begin{aligned}\sqrt{q_a} &\leq |R_a |\psi_T\rangle| + \sqrt{T/M} \\ &\leq 3\sqrt{2}\cdot\sum_{i=1}^{T}\sqrt{\frac{P+i-1}{M}} + |R_a |\psi_0\rangle| + \sqrt{T/M} \\ &\leq 3\sqrt{2}\cdot\sum_{i=1}^{T}\sqrt{\frac{P+i-1}{M}} + |Q_a |\psi_0\rangle| + \sqrt{T/M}.\end{aligned}$$

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 [**?**]. We only prove one of the main results in [**?**], 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.

Definition 13 (G_{OWF}). *The security game $G_{\mathsf{OWF}} = (C_{\mathsf{OWF}})$ is defined as the following, where the challenger C_{MDHF} is specified by these 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 8. 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 10. G_{OWF} *is* $O((P + T^2)/\min\{N, M\})$*-secure in the P-BF-QROM.*

Proof. To prove it, we first recall Lemma 1.5 in [**?**]. 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 [**?**].

Lemma 11 (Lemma 1.5, [?]). *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. □

Acknowledgments We thank the anonymous reviewers for useful suggestions and comments. Siyao Guo is supported by Shanghai Eastern Young Scholar Program SMEC-0920000169, and the NYU Shanghai Boost Fund. Qian Li's work is supported in part by the National Natural Science Foundation of China Grants No. 61832003, 61872334, 61801459, 62002229, and the Strategic Priority Research Program of Chinese Academy of Sciences Grant No. XDB28000000. Qipeng Liu is supported by the NSF and the Simons Institute for the Theory of Computing, through a Quantum Postdoctoral Fellowship.