Quantum Random Oracle Model with Auxiliary Input

Minki Hhan*[1], Keita Xagawa[2], and Takashi Yamakawa[2]

[1] Seoul National University, Seoul, Republic of Korea
hhan_@snu.ac.kr
[2] NTT Secure Platform Laboratories
3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585 Japan
keita.xagawa.zv@hco.ntt.co.jp, takashi.yamakawa.ga@hco.ntt.co.jp

Abstract. The random oracle model (ROM) is an idealized model where hash functions are modeled as random functions that are only accessible as oracles. Although the ROM has been used for proving many cryptographic schemes, it has (at least) two problems. First, the ROM does not capture quantum adversaries. Second, it does not capture non-uniform adversaries that perform preprocessings. To deal with these problems, Boneh et al. (Asiacrypt'11) proposed using the quantum ROM (QROM) to argue post-quantum security, and Unruh (CRYPTO'07) proposed the ROM with auxiliary input (ROM-AI) to argue security against preprocessing attacks. However, to the best of our knowledge, no work has dealt with the above two problems simultaneously.

In this paper, we consider a model that we call the QROM with (classical) auxiliary input (QROM-AI) that deals with the above two problems simultaneously and study security of cryptographic primitives in the model. That is, we give security bounds for one-way functions, pseudorandom generators, (post-quantum) pseudorandom functions, and (post-quantum) message authentication codes in the QROM-AI.

We also study security bounds in the presence of quantum auxiliary inputs. In other words, we show a security bound for one-wayness of random permutations (instead of random functions) in the presence of quantum auxiliary inputs. This resolves an open problem posed by Nayebi et al. (QIC'15). In a context of complexity theory, this implies $\mathsf{NP} \cap \mathsf{coNP} \not\subseteq \mathsf{BQP}/\mathsf{qpoly}$ relative to a random permutation oracle, which also answers an open problem posed by Aaronson (ToC'05).

1 Introduction

1.1 Background

Random Oracle Model with Auxiliary Input. The random oracle model (ROM) introduced by Bellare and Rogaway [BR93] is a remarkably useful tool for analyzing security of practical cryptographic schemes. In the ROM, we model a hash function as a truly random function that is only accessible as an oracle and assume that an adversary has no a priori knowledge about the function. This means that the traditional definition of the ROM does not capture *non-uniform* adversaries who perform heavy offline preprocessings to generate auxiliary information (also called advice) of the random function. Indeed, a non-uniform attack is effective in some cases [Hel80, FN99, DTT10]. For example, Hellman [Hel80] showed that one can speed up an inversion of a permutation by using the power of preprocessing. Bernstein and Lange [BL13] pointed out that non-uniform attacks are a potential threat in the real world by exhibiting some examples of (unrealistic) non-uniform attacks. To deal with such non-uniform attacks, Unruh [Unr07] introduced the *random oracle model with auxiliary input* (ROM-AI) where an adversary can perform arbitrarily heavy preprocessing to generate auxiliary information of the random function. He gave a generic tool for analyzing security in the ROM-AI by introducing another model called the bit-fixing ROM and showed that a random oracle is one-way and that RSA-OAEP [BR95] remains secure in the ROM-AI. Subsequently, Dodis, Guo, and Katz [DGK17], and Coretti, Dodis, Guo, and Steinberger [CDGS18] further studied the ROM-AI to show (tighter) security bounds for several natural applications including one-way functions (OWFs), collision resistant hash functions (CRHFs), pseudorandom generators (PRGs), pseudorandom functions (PRFs), message authentication codes (MACs), and more.

* This work was done in part while the first author was conducting an internship program in NTT Secure Platform Laboratories, Japan.

Quantum Random Oracle Model. The ROM has been strengthened in another direction called the *quantum ROM* (QROM) [BDF+11], where an adversary can access the random oracle quantumly. This is a natural model when considering post-quantum security since a random oracle is an idealization of a hash function that can be quantumly evaluated by an adversary once quantum computers are available. Since many proof techniques in the ROM cannot be directly translated into ones in the QROM, many studies have given security proofs in the QROM for schemes that are originally proven secure in the ROM (e.g., [Zha12b, Unr15, ES15, TU16, HRS16, CBH+18, KLS18, SXY18, JZC+18, KYY18, AHU19, DFMS19, LZ19]).

Quantum Random Oracle Model with (Quantum) Auxiliary Input. Although both the ROM-AI and QROM have been studied thoroughly, to the best of our knowledge, no work has considered both these extensions simultaneously. In this work, we consider a mix of them and initiate the study of the *QROM with auxiliary input.* In particular, we consider both the QROM with *classical* auxiliary input (QROM-AI) and the QROM with *quantum* auxiliary input (QROM-QAI). Both these models reasonably extend the QROM to capture adversaries with preprocessing in some sense. The QROM-AI captures an adversary that performs a long classical preprocessing to prepare classical auxiliary information that will be used in the future when quantum computers become available. This model is reasonable in the current situation in which quantum computers are not available yet and in a future situation in which quantum computers are available, but are far less efficient than classical computers. On the other hand, the QROM-QAI would be more reasonable in the situation where a highly efficient quantum computer is available at the time of preprocessing. The motivation of this work is to study security of natural applications of random oracles in these models.

The work most relevant to the above problem is that of Nayebi, Aaronson, Belovs, and Trevisan [NABT15], which showed a lower bound for the number of queries to invert a random permutation with classical auxiliary input. However, their result is not sufficient for our purpose in several aspects. First, they only considered a random *permutation* whereas we consider a random *function.* Since a hash function in the real world is not a permutation, we need to consider a random function instead of a random permutation to derive implications in the real world. Second, they only considered a lower bound for one-wayness whereas we are also interested in other applications such as CRHFs, PRGs, PRFs, and MACs. Third, they did not consider the effect of *salting*, which is a technique to use a random string that is chosen after the preprocessing as a public parameter. Salting is widely deployed in the real world, and sufficiently long salt defeats non-uniform attacks in the ROM-AI [DGK17, CDGS18]. Finally, they only considered settings where auxiliary inputs are classical, and their result seems difficult to directly extend to the setting where auxiliary inputs are quantum. Indeed, they left it extending their result to the quantum auxiliary input setting as an open problem. Thus it remains unknown if we can obtain security bounds for the security of OWFs, CRHFs, PRGs, PRFs, and MACs and if salting is effective in the QROM-AI and QROM-QAI.

1.2 Our Results

In this work, we initiate the study of the QROM-AI and the QROM-QAI, and give security bounds for several cryptographic applications in the QROM-AI. However, we do not know if we can extend them to ones in the QROM-QAI. Nonetheless, we make a step toward the goal by proving that a random permutation (instead of a random function) is hard to invert even with a quantum auxiliary input. This answers the open problem raised by Nayebi et al. [NABT15]. We describe more details of our results below.

Security Bounds in QROM-AI. We prove security bounds for natural "salted" constructions of OWFs, PRGs, PRFs, and MACs in the QROM-AI. A caveat of our results for PRFs and MACs is that we only consider *classical queries* for PRF and MAC oracles whereas queries to the random oracle can be quantum. To clarify this limitation, we denote them as pqPRFs and pqMACs. [3] On the other hand, we denote quantum-accessible PRFs and MACs as qPRFs and qMACs. We note that the attack models of pqPRFs and pqMACs make sense as post-quantum security models a setting where honest parties are all classical and only adversaries are quantum.

Our results are summarized in Table 1. (An extended table that includes security bounds and attacks in the ROM-AI can be found in Table 2 in Appendix E.) The notations used in the table are

[3] "pq" stands for "post-quantum".

	Security bounds in QROM-AI (Ours)	Best known attacks in QROM-AI
OWFs	$\left(\frac{ST^2}{K\alpha}+\frac{T^2N}{\alpha^2}\right)^{1/2}$	$\min\left\{\frac{ST}{K\alpha},\left(\frac{S^2T}{K^2\alpha^2}\right)^{1/3}\right\}+\frac{T^2}{\alpha}$
PRGs	$\left(\frac{ST^4}{KN}+\frac{T^4}{N}\right)^{1/6}$	$\left(\frac{ST}{KN}\right)^{1/2}+\frac{T^2}{N}$
pqPRFs	$\left(\frac{ST^4}{KN}+\frac{T^4}{N}\right)^{1/4}+Q_{\mathsf{prf}}\left(\frac{ST^2}{KN}\right)^{1/6}$	$\left(\frac{ST}{KN}\right)^{1/2}+\frac{T^2}{N}$
pqMACs	$\left(\frac{ST^4}{KN}+\frac{T^4}{N}+\frac{1}{M}\right)^{1/3}$	$\min\left\{\frac{ST}{KN},\left(\frac{S^2T}{K^2N^2}\right)^{1/3}\right\}+\frac{T^2}{N}+\frac{1}{M}$

Table 1. Security bounds and best known attacks using an S-bit auxiliary input and T queries to the random oracle for "salted" constructions of primitives in the QROM-AI. The first two primitives (unkeyed primitives) are constructed from a random oracle $\mathcal{O}:[K]\times[N]\to[M]$ where $[K]$ is the domain of the salt, $[N]$ is the domain of the input (or the seed for PRGs), $[M]$ is the domain of the outputs, and we let $\alpha:=\min(N,M)$. The latter two primitives (keyed primitives) are constructed from a random oracle $\mathcal{O}:[K]\times[N]\times[L]\to[M]$ where $[K]$ is the domain of the salt, $[N]$ is the domain of the key, $[L]$ is the domain of the inputs, and $[M]$ is the domain of the outputs (or authenticators for MACs). Q_{prf} denotes the number of queries to the PRF oracle in the security bound for pqPRFs. We omit constant factors and logarithmic terms for simplicity.

the same as those used in [DGK17]. The "Security bounds in QROM-AI" column indicates upper bounds of advantages to break these primitives by an adversary that makes T quantum queries to the random oracle and is given a classical auxiliary input of size at most S bits. The "Best known attacks in QROM-AI" column indicates advantages that are achieved by the best known attacks. (Appendix E briefly explains how we filled this column.) Though our bounds in the QROM-AI are much less tight than those in the ROM-AI and far from matching the best known attacks, we can derive some meaningful implications from them. For example, our bounds imply the computational hardness of these primitives if the size of domain and ranges are sufficiently large[4]. Moreover, our bounds imply that if we use a large enough salt, these primitives remain secure even if an adversary prepares a very long auxiliary input. That is, if the size K of the domain of the salt is exponentially larger than the auxiliary input size S, then terms that depend on S are negligible. This extends similar results in the ROM-AI [DGK17, CDGS18] to the QROM-AI.

On Quantum Auxiliary Input. Unfortunately, we could not obtain any meaningful security bound in the QROM-QAI where quantum auxiliary inputs are available. Nonetheless, we give a security bound for a closely related problem: one-wayness of a random permutation (instead of a random function) with *quantum auxiliary input.* That is, we show that the probability of inverting a random function $\mathcal{O}:[K]\times[N]\to[N]$ such that $\mathcal{O}(a,\cdot)$ is a permutation over $[N]$ for all $a\in[K]$ with an S-qubit quantum auxiliary input and T quantum queries is $\widetilde{O}\left(\left(\frac{ST^2}{KN}+\frac{T^2}{N}\right)^{1/3}\right)$. This answers the open problem raised by Nayebi et al. [NABT15]. Before our work, such a result was known in the setting where an auxiliary input is classical and $K=1$ [NABT15], which gave a security bound $\widetilde{O}(\sqrt{ST^2/N})$. [5]

Our result also has an implication in complexity theory. Specifically, it implies an oracle separation of $\mathsf{NP}\cap\mathsf{coNP}$ and $\mathsf{BQP/qpoly}$ which is the class of problems solvable by a polynomial-size quantum algorithm with a polynomial-size quantum advice [NY04, Aar05]. That is, we have $\mathsf{NP}\cap\mathsf{coNP}\not\subseteq\mathsf{BQP/qpoly}$ relative to a random permutation oracle. This affirmatively answers the open problem left by Aaronson [Aar05], who showed the existence of an oracle relative to which $\mathsf{NP}\not\subseteq\mathsf{BQP/qpoly}$ and left it open to show the existence of an oracle relative to which $\mathsf{NP}\cap\mathsf{coNP}\not\subseteq\mathsf{BQP/qpoly}$.

[4] More precisely, if both S and T are polynomial in the security parameter and (appropriate parts of) domains and ranges of the random oracle are exponentially large then our bounds become negligibly small.

[5] They claim that their security bound is $\widetilde{O}(ST^2/N)$. However, their definition of one-wayness is weaker than ours, and if we use our definition, then the quadratic security loss naturally occurs. See Appendix D for more detailed discussion.

1.3 Technical Overview

Our main tool is the *compression technique* developed by Genarro, Gertner, Katz, and Trevisan [GT00, GGKT05]. The basic idea behind the technique is a very simple information theoretic argument: For sets $\mathcal{M}, \mathcal{C}$, if there exist an encoding algorithm $E : \mathcal{M} \to \mathcal{C}$ and a decoding algorithm $D : \mathcal{C} \to \mathcal{M}$ such that $D(E(m)) = m$ holds with high probability (over the uniformly random choice of m), then the cardinality of $\mathcal{C}$ cannot be much smaller than that of $\mathcal{M}$. More precisely, if the decoding succeeds with probability δ, then we must have $|\mathcal{C}| \geq \delta|\mathcal{M}|$. This holds even if the encoder and the decoder share a randomness of any length [DTT10]. We call this information theoretical bound the *compression lemma*. In the following, we explain how to apply this to derive security bounds in the QROM-AI. We omit salting for simplicity since similar methods still work with salting.

OWFs in QROM-AI. Here, we explain how to obtain a security bound for OWFs in the QROM-AI. First, we review the case of random permutations, which is shown by Nayebi et al. [NABT15] because this is much simpler. Suppose that we have a random permutation $f : [N] \to [N]$ and an adversary $\mathcal{A}$ that succeeds in inverting f with high probability, say $2/3$, for ε-fraction of $x \in [N]$ by using S-bit classical auxiliary information of f and T quantum queries to f. We want to give an upper bound for ε.

The idea is to construct an encoder that compresses the truth table of the random oracle by using the power of the adversary $\mathcal{A}$ and then invoke the compression lemma. Specifically, we choose a random subset $R \subset [N]$ by putting each element $x \in [N]$ into R with a certain probability, which will be used as the shared randomness between the encoder and the decoder. Then we define the set $G \subset R$ of good elements where we say that $x \in R$ is good if $\mathcal{A}$ succeeds in inverting $f(x)$ with high probability and $\mathcal{A}$'s total query magnitude on any $x' \in R \setminus \{x\}$ is "small" when it runs on the input $f(x)$. By appropriately setting parameters, we can show that G is "not too small" with high probability. Then the encoder generates an encoding that consists of a "partial truth table" of f on $[N] \setminus G$, the description of the set $f(G)$ and the auxiliary input that is used by $\mathcal{A}$. The decoder recovers the whole truth table of f by inverting f on each element of $f(G)$ by running $\mathcal{A}$. Here, we have to be careful about the fact that the decoder is not given the whole truth table of f and cannot correctly simulate the oracle f for $\mathcal{A}$. Thus, when the decoder tries to invert $y \in f(G)$ in f, it defines a function g_y by

$$g_y(x) := \begin{cases} f(x) & \text{if } x \notin R \\ y & \text{if } x \in R, \end{cases}$$

and uses g_y instead of f. Though f and g_y do not match on $R \setminus \{x\}$, by the definition of the good elements, $\mathcal{A}$'s query magnitude on $R \setminus \{x\}$ is "small," and thus $\mathcal{A}$ still succeeds in inverting y with high probability with the oracle access to g_y instead of f. Then the decoder can recover $x = f^{-1}(y)$ by computing the output distribution of $\mathcal{A}$ and taking the value that is output with the highest probability.[6] By repeating this for every $y \in f(G)$, the decoder can recover the whole truth table of f. On the other hand, the encoding is smaller than the original truth table of f since it "forgets" the truth table on the subset G that is "not too small." By setting parameters appropriately, we can derive the security bound.

For random functions instead of random permutations, the difference is that a preimage of y may not be unique, and we have to bound the probability that an adversary finds any of them. In that case, even if an adversary succeeds in inverting the random function with high probability, there may not be any particular value that is output with constant probability. Thus the decoder has to use a value that is output by the adversary with sub-constant probability for recovering the truth table. This only gives a somewhat bad bound related to this probability, even if we resolve other technical difficulties.

To deal with this problem, we include a randomness used in the measurement of the final state of $\mathcal{A}$ as a part of the shared randomness between the encoder and decoder. With a fixed randomness for the measurement, the decoder can *deterministically simulate* $\mathcal{A}$[7] and decide the value that is supposed to be used for recovering the table. With this idea (among others), we extend the above result to the case of random functions.

[6] Since the compression lemma works for unbounded-time encoders and decoders, we can assume that the decoder has an unbounded computational power to simulate quantum computations.

[7] Since the decoder has unbounded computational power, it can control the randomness for measurements in executions of the quantum algorithm $\mathcal{A}$.

PRGs in QROM-AI. For obtaining security bounds for PRGs, we first consider (an average case version of) Yao's box problem [Yao90] similarly to the classical case [DTT10, DGK17]. In Yao's box problem, we consider a random oracle $\mathcal{O} : [N] \to \{0,1\}$ and an adversary that tries to compute $\mathcal{O}(x)$ for uniform $x \in [N]$ by using an S-bit classical auxiliary input and T quantum queries to $\mathcal{O}$ *without querying x itself* (i.e., $\mathcal{A}$'s query magnitude on x is 0 in the quantum case). If we obtain a proper bound for Yao's box problem, then a bound for PRGs follows as discussed below. To construct PRGs, we consider a random oracle $\mathcal{O} : [N] \to [M]$ and want to bound the advantage of $\mathcal{A}$ to distinguish $\mathcal{O}(x)$ for $x \leftarrow [N]$ from a truly random string $y \leftarrow [M]$ by using an S-bit classical auxiliary input and T quantum queries to $\mathcal{O}$.

First, we argue that $\mathcal{A}$'s total query magnitude on x is "small." This holds because if it is "not small," then we can use $\mathcal{A}$ to invert $\mathcal{O}$ with "non-small" probability by measuring one of its queries, which contradicts the bound for the one-wayness of $\mathcal{O}$. Then we can convert $\mathcal{A}$ to an algorithm $\mathcal{A}'$ whose query magnitude on x is 0 while only slightly degrading its distinguishing advantage.[8] Now, $\mathcal{A}'$ distinguishes $\mathcal{O}(x)$ from a random string without querying x at all. By Yao's equivalence of distinguishability and predictability [Yao82], there exists an algorithm $\mathcal{B}$ such that for some $i \in [\log M]$, it predicts the i-th bit of $\mathcal{O}(x)$ given an advice $\mathsf{st}_{\mathcal{O}}$ of S-bit, x, and the first $i-1$ bits of $\mathcal{O}(x)$ making T quantum queries to $\mathcal{O}$ without querying x to $\mathcal{O}$. This is exactly an algorithm that solves Yao's box problem by also considering the first $i-1$ bits of $\mathcal{O}(x)$ as a part of the auxiliary input.[9] Therefore we can apply the bound for Yao's box problem to derive a security bound for PRGs in the QROM-AI.

What is left is how to derive a security bound for Yao's box problem.[10] Basically, we follow the classical counterpart that was shown by De et al. [DTT10], which is roughly described as follows. First, we choose a random subset $R \subset [N]$ by putting each element of $x \in [N]$ into R with a certain probability, which will be used as the shared randomness between the encoder and the decoder. Then we define the set G of good elements where we say that $x \in [N]$ is good if (A): $x \in R$, and (B): for any query x' made by $\mathcal{A}$ with input x, we have $x' \notin R$.[11] Then we partition G into two subsets G_0 that consists of all $x \in G$ such that $\mathcal{A}$ correctly guesses $\mathcal{O}(x)$ on input x, and $G_1 := G \setminus G_0$. By some analyses of probabilities, they showed that $|G|$ is "not too small" and $|G_0| - |G_1| = \Omega(\varepsilon|G|)$ with "non-small" probability where ε is $\mathcal{A}$'s advantage (i.e., $\mathcal{A}$ returns the correct answer with probability $1/2 + \varepsilon$). Then they construct an encoder that outputs the partial truth table of $\mathcal{O}$ on $[N] \setminus G$, the description of the set G_0, and the auxiliary input used by $\mathcal{A}$. The decoder can recover the whole truth table of $\mathcal{O}$ by running $\mathcal{A}$ on each $x \in G$ and negating it if $x \in G_1$.[12] We note that the decoder never gets stuck in simulating the oracle since all of $\mathcal{A}$'s queries are outside R where the decoder knows the value of $\mathcal{O}$. They showed that the encoding size is much smaller than the whole truth table when $|G_0| - |G_1|$ is "large". (Note that the needed number of bits to represent the set G_0 is smaller when $|G_0| - |G_1|$ is larger since the number of possible choices of G_0 and G_1 is smaller when $|G_0| - |G_1|$ is larger assuming $|G_0| > |G_1|$.) More specifically, they showed that we can obtain a meaningful bound when $|G|$ is "not too small" and we have $|G_0| - |G_1| = \Omega(\varepsilon|G|)$, which occurs with "non-small" probability.

When generalizing this strategy to the quantum setting, there are several obstacles.

First, the condition (B) is not well-defined in the quantum setting. This can be easily adapted by requiring that $\mathcal{A}$'s query magnitudes on elements of R are "small" instead of requiring $\mathcal{A}$ to not query any of them.

Second, the sets G_0 and G_1 are not well-defined in the quantum setting since we cannot assume $\mathcal{A}$ is deterministic in the quantum setting. This can be resolved by including the randomness for measurements in the shared randomness between the encoder and decoder similarly to the case of OWFs.

Third, in the classical setting, for proving that $|G|$ is "not too small" and we have $|G_0| - |G_1| = \Omega(\varepsilon|G|)$ with "non-small" probability, we use the fact that the probability that x is good (i.e., $\Pr[x \in G]$) is constant for all $x \in [N]$. In the classical setting, this can be assumed without loss of

[8] In the actual proof, we rely on the *semi-classical one-way to hiding theorem* recently given by Ambainis, Hamburg, and Unruh [AHU19].

[9] More precisely, since an auxiliary input cannot depend on x, we consider the partial truth table of $\mathcal{O}$ that gives the first $i-1$ bits of $\mathcal{O}(x)$ for all x as a part of the auxiliary input.

[10] Nayebi et al. [NABT15] also studied Yao's box problem. However, they only considered the worst case, so their result is not applicable for our purpose.

[11] Recall that this is a review of the classical case, and thus this condition is well-defined.

[12] Though the encoding does not contain the description of G, the decoder can recover it from R.

generality since we can force an adversary to not make the same queries twice. On the other hand, this cannot be assumed in the quantum setting, and $\Pr[x \in G]$ may depend on x. Fortunately, we can still show that if we choose parameters appropriately, then $\Pr[x \in G]$ are well-balanced, i.e., maximal and minimal values of $\Pr[x \in G]$ are very close. By using this, we can still prove that $|G|$ is "not too small" and we have $|G_0| - |G_1| = \Omega(\varepsilon|G|)$ with "non-small" probability though the proof becomes more involved.

With these ideas, we obtain a security bound for Yao's box problem in the quantum setting.

pqPRFs and pqMACs in QROM-AI. With ideas used for OWFs and PRGs as explained above, the results for pqPRFs and pqMACs in the ROM-AI in [DGK17] can be naturally translated into ones in the QROM-AI. Since the original bounds in [DGK17] only considered classical accesses to PRF/MAC oracles, our results inherit this. One thing we have to care about here is that classical PRF and MAC oracles are not unitary, and we cannot assume that measurements are deferred to the end of the computation by the adversary. Thus for applying our technique of deterministic simulation of quantum computations, we include randomness for all measurements that are possibly done in the middle of the computation by the adversary in the shared randomness between the encoder and decoder. We note that the size of shared randomness does not affect the limitation of a compression, and this does not make our bounds worse.

Bound for Inverting Permutations with Quantum Advice. Next, we move on to discussing quantum auxiliary inputs. Our strategy is to use the compression technique similarly to the case of the classical auxiliary inputs. However, if we consider quantum auxiliary inputs, we first have to extend the compression lemma to the setting where encodings are quantum. Fortunately, such an extension is already known [Nay99, NS06], and both papers showed that the bound is almost the same as the classical case.

Given this, one may think that security bounds in the QROM-AI are quite easy to extend to ones in the QROM-QAI. However, this is not the case. Recall that decoders in these proofs run an adversary $\mathcal{A}$ many times. On the other hand, we cannot reuse a quantum auxiliary input since it may be broken in each running of $\mathcal{A}$. Thus, an encoding has to contain as many copies of the auxiliary input as the number of executions of $\mathcal{A}$ by the decoder, in which case the encoding is no longer small. Indeed, Nayebi et al. [NABT15] mentioned that their result is difficult to extend to the quantum auxiliary input setting for this reason.

We overcome this issue by using a general principle of quantum information, often called the *gentle measurement lemma* [Win99, AR19], which states that if we can predict the outcome of a measurement with probability almost 1, then the measurement barely damages the quantum state. To apply the lemma, we amplify the success probability of an adversary $\mathcal{A}$ to almost 1 by running it many times.[13] Especially, if the correct solution of a problem in question is unique (as in the inversion problem of a permutation), then $\mathcal{A}$ outputs a certain value with probability almost 1. In this case, the quantum auxiliary input is not damaged much in each running of $\mathcal{A}$ due to the gentle measurement lemma and can be reused many times in the decoding procedure. We note that the decoder still needs a certain number of copies of the auxiliary input since it has to run the adversary many times to amplify the success probability. However, the number of copies needed is not too large since the adversary's error probability decreases exponentially in the number of repetitions. Thus, the encoding does not become too large, and we can obtain a meaningful bound. This is how we obtain a security bound for inverting a random permutation with quantum advice.

We note that the above method crucially relies on the solution of the problem being unique. Otherwise, even if an adversary's success probability is almost 1, its output may still have high entropy, in which case the gentle measurement lemma is not applicable. This is why we limit our attention to random *permutations* instead of random *functions*.

1.4 Limitations and Open Problems

Though we made progress in understanding the power of non-uniform attacks in the quantum setting, our results contain many limitations.

1. We do not have any result for CRHFs in the QROM-AI/QROM-QAI.

[13] A similar idea was used by Aaronson [Aar05] to show limitations of quantum one-way communication and algorithms with quantum advice.

2. Our results on PRFs and MACs in the QROM-AI are limited to pqMACs and pqPRFs where oracles (except for the random oracle) are classical.
3. All security bounds shown in this paper are much less tight than the counterparts in the classical setting, and far from matching the best known attacks. We note that known security bounds of many primitives including OWFs, PRGs, PRFs, and MACs in the ROM-AI do not match the best known attacks even in the classical setting [DGK17, CDGS18].
4. Our techniques cannot be used for analyzing schemes on the basis of computational assumptions since it would be difficult to capture these assumptions with the compression technique. We note that this limitation is overcome by using another technique called the *pre-sampling technique* instead of a compression technique in the classical setting [Unr07, CDGS18].
5. We have no security bound in the QROM-QAI. A possible approach toward that is to extend our result on one-wayness of a random permutation with quantum auxiliary input.

We leave the above limitations as open problems to be overcome.

Also, we are not aware of any non-trivial attack in the QROM-AI or QROM-QAI that outperforms ones in the ROM-AI except for attacks that just ignore auxiliary inputs (e.g., Grover's algorithm [Gro96] and BHT [BHT97] algorithm). We leave it as an interesting open problem to give a non-trivial attack that utilizes auxiliary inputs against any primitive in the QROM-AI or QROM-QAI.

1.5 Related Work

Security Bounds against Non-Uniform Attacks in Other Models. Corrigan-Gibbs and Kogan [CK18] studied non-uniform attacks in the generic group model (GGM), showed security bounds for several problems including the discrete logarithm problem that matches the best known attack. Their results are based on the compression technique. Coretti, Dodis, and Guo [CDG18] studied non-uniform attacks in the random permutation model (RPM), ideal-cipher model (IPM), and GGM, and showed security bounds for many applications in these models by developing a general tool to analyze them. Their results are based on the pre-sampling technique. We note that both above works only consider classical attacks.

Quantum-Accessible PRFs and MACs. Zhandry [Zha12a] gave the first constructions of qPRFs from OWFs or learning with errors (LWE) assumption in the standard model as well as a separation between pqPRFs and qPRFs.

Boneh and Zhandry [BZ13] formally defined qMACs and showed that qPRFs are sufficient to construct them. A stronger and the best current security notion for qMACs was proposed by Garg, Yuen, and Zhandry [GYZ17].

We note that these works focus on constructions in the standard model, whereas this work focuses on hash-based constructions in the QROM-AI or QROM-QAI that are much more efficient.

Compression Technique in Quantum Setting. Besides Nayebi et al. [NABT15], Hosoyamada and Yamakawa [HY18] also used the compression technique in the quantum setting to show a black-box separation of CRHFs from one-way permutations. Their technique is incomparable with ours as they showed bounds for inverting random permutations in the presence of a specific quantum oracle that finds collisions whereas we show bounds for several applications of a random oracle in the presence of any bounded-length auxiliary inputs.

2 Preliminaries

Notations. We say a function $\varepsilon(n)$ is negligible if $\varepsilon(n) < 1/|p(n)|$ for any polynomial p for sufficiently large n. For a positive integer n, we write $[n] = \{1, \ldots, n\}$ to denote the set of positive integers less than or equal to n. In tilde notations $\widetilde{O}(f(A, B, \cdots))$ or $\widetilde{\Omega}(f(A, B, \cdots))$, we ignore non-negative degree polylogarithmic factors with respect to all capital variables which appear in the context.For example, we write $(T^2/N)\cdot\log M = \widetilde{O}(T^2/N)$. To denote the event that a probabilistic or quantum algorithm $\mathcal{A}$ with input z outputs x, we write $\mathcal{A}(z) \to x$.

Quantum algorithms have intrinsic randomness when they perform measurements. The probability that a quantum algorithm $\mathcal{A}$ outputs x on an input z is denoted by $\Pr_{\mathcal{A}}[\mathcal{A}(z) \to x]$. To denote quantum objects such as quantum states or a quantum-accessible oracle, we use the ket notation $|\cdot\rangle$. For example, $|\phi\rangle$ denotes a quantum state, while x is a classical string. For basics of quantum computing, we refer readers to [NC00].

2.1 Oracle-Aided Quantum Algorithm

An oracle-aided quantum algorithm is a quantum algorithm that can perform quantum computations and can access oracles. In this paper, we consider three types of oracles: quantum-accessible oracle, classical-accessible oracle, and semi-classical oracle [AHU19], which is explained in the next subsection. A quantum-accessible oracle that computes a function $f : X \to Y$ applies a unitary that transforms a query $|x, y\rangle$ to $|x, y \oplus f(x)\rangle$, and returns the resulting state. A classical-accessible oracle that computes a function $f : X \to Y$, given a query $|x, y\rangle$, first measures the input register $|x\rangle$, and then returns $|x, y \oplus f(x)\rangle$. Note that a classical-accessible oracle is not unitary. We often use $\mathcal{A}^{|f\rangle}$ to mean that $\mathcal{A}$ accesses a quantum-accessible oracle that computes f and $\mathcal{A}^f$ to mean that $\mathcal{A}$ accesses classical-accessible oracle that computes f. We allow an oracle-aided quantum algorithm to make queries in parallel. Its query depth d is defined to be the number of oracle calls counting parallel queries as one query.

2.2 Semi-Classical Oracle

In this section, we review *semi-classical oracles* introduced in [AHU19]. Here, we only define a semi-classical oracle for the indicator function of a set S since we only need it in this paper. A semi-classical oracle $\mathcal{O}_S^{SC}$ for a set $S \subseteq X$ is queried with two registers, an input register Q with $\mathbb{C}^X$ and an output register R with space $\mathbb{C}^2$. When queried with a value $|x\rangle$ in Q, the oracle returns whether $x \in S$ in the output register R. More formally, it performs a measurement with projectors M_0 and M_1, where $M_0 := \sum_{x \in X \setminus S} |x\rangle \langle x|$ and $M_1 := \sum_{x \in S} |x\rangle \langle x|$, and initializes R to $|0\rangle$ or $|1\rangle$ corresponding to the measurement result.

In the execution of a quantum algorithm $\mathcal{A}^{\mathcal{O}_S^{SC}}$, Find denotes the event that $\mathcal{O}_S^{SC}$ returns $|1\rangle$. This event is well-defined, since $\mathcal{O}_S^{SC}$ *measures* its outputs.

Punctured oracle. If H is an oracle with domain X and codomain Y, we define $|H\rangle \setminus S$ as an oracle which, on input x, first queries $\mathcal{O}_S^{SC}(x)$ and then queries $H(x)$. The following lemma ([AHU19, Lemma 1]) states that the outcome of $\mathcal{A}^{|H\rangle \setminus S}$ is independent of $H(x)$ for all $x \in S$ when Find does not occur.

Lemma 1 (Punctured Oracle [AHU19, Lemma 1]). *Let $S \subseteq X$ be random. Let $G, H : X \to Y$ be random functions satisfying $G(x) = H(x)\ \forall x \notin S$. Let z be a random bit string. (S, G, H, z may have an arbitrary joint distribution.)*

Let $\mathcal{A}$ be an oracle-aided quantum algorithm of query depth d (not necessarily unitary). Let E be an arbitrary (classical) event. Then we have

$$\Pr[E \wedge \neg\mathsf{Find} : x \leftarrow \mathcal{A}^{|H\rangle \setminus S}(z)] = \Pr[E \wedge \neg\mathsf{Find} : x \leftarrow \mathcal{A}^{|G\rangle \setminus S}(z)].$$

We review the semi-classical oneway-to-hiding lemma (the SC-O2H lemma in short):

Lemma 2 (The SC-O2H lemma [AHU19, Theorem 1]). *Let $S \subseteq X$ be random. Let $G, H : X \to Y$ be random functions satisfying $\forall x \notin S\ [G(x) = H(x)]$. Let z be a random bit string. (S, G, H, z may have an arbitrary joint distribution.)*

Let $\mathcal{A}$ be an oracle-aided quantum algorithm of query depth d (not necessarily unitary). Let

$$\begin{aligned} P_{\text{left}} &:= \Pr[b = 1 : b \leftarrow \mathcal{A}^{|H\rangle}(z)], \\ P_{\text{right}} &:= \Pr[b = 1 : b \leftarrow \mathcal{A}^{|G\rangle}(z)], \\ P_{\text{find}} &:= \Pr[\mathsf{Find} : A^{|G\rangle \setminus S}(z)] = \Pr[\mathsf{Find} : A^{|H\rangle \setminus S}(z)]. \end{aligned}$$

Then we have

$$|P_{\text{left}} - P_{\text{right}}| \leq 2\sqrt{(d+1) \cdot P_{\text{find}}} \text{ and } |\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}| \leq 2\sqrt{(d+1) \cdot P_{\text{find}}}.$$

The lemma also holds with bound $\sqrt{(d+1) \cdot P_{\text{find}}}$ for the following alternative definition of P_{right}:

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

We often denote the above probability by $\Pr[\neg\mathsf{Find} : A^{|G\rangle \setminus S}(z) \to 1]$ for notational simplicity.

Lemma 3 (Search in semi-classical oracle [AHU19, Theorem 2 and Corollary 1]). *Let $\mathcal{A}$ be any oracle-aided quantum algorithm making at most q queries (depth d) to a semi-classical oracle with domain X. Let $S \subseteq X$ and $z \in \{0,1\}^*$. (S, z may have an arbitrary joint distribution.)*

Let $\mathcal{B}$ be an algorithm that on input z chooses $i \leftarrow \{1,\dots,d\}$; runs $\mathcal{A}^{\mathcal{O}_{\emptyset}^{SC}}(z)$ until (just before) the i-th query; then measures all query input registers in the computational basis and outputs the set T of measurement outcomes.

Then we have

$$\Pr[\mathsf{Find} : \mathcal{A}^{\mathcal{O}_S^{SC}}(z)] \le 4d \cdot \Pr[S \cap T \neq \emptyset : T \leftarrow \mathcal{B}(z)].$$

In particular, if S and z are independent, $\mathcal{A}$ makes at most q queries, and we let $P_{\max} := \max_{x \in X} \Pr[x \in S]$, then we have

$$d \cdot \Pr[S \cap T \neq \emptyset : T \leftarrow \mathcal{B}(z)] \le q \cdot P_{\max}.$$

Remark 1. In the above lemmas, the input z is assumed to be a classical string. However, we can obtain exactly the same bound even if z is a quantum state. This is because any quantum state can be described by a classical string with an exponential blowup of the size, and the above lemmas are only about query-complexities and the size of z does not matter.

3 Quantum ROM with Classical AI

In this section, we show security bounds for primitives in the QROM-AI.

3.1 Preparations

First, we prepare some lemmas and notations that are used in our proofs.

Compression Lemma The following lemma states that there exists an information-theoretic lower bound for a compression algorithm.

Lemma 4 ([DTT10, Fact 8.1]). *Let M, C, R be sets. Let $E : M \times R \to C$ and $D : C \times R \to M$ be deterministic algorithms. For $\delta \in [0,1]$, if we have*

$$\Pr_{r \leftarrow R}[D(E(m,r),r) = m] \ge \delta$$

for all $m \in M$, then we have $|C| \ge \delta |M|$, which can be rephrased as $\log |C| \ge \log |M| - \log 1/\delta$.

We use the above lemma (which we call the *compression lemma*) to derive security bounds for various primitives in the QROM-AI by constructing a pair of encoding and decoding algorithms that compress the truth table of a random function by using the power of an adversary against the primitive. Note that we encode a function into a classical bit string while we use a quantum adversary.

Simulating Measurement Quantum algorithms are inherently randomized due to the intrinsic randomness of measurements. However, if we do not care about the running-time, we can fix the randomness in the measurement by classically simulating the execution of the algorithm.

More precisely, we can classically simulate an execution of any quantum algorithm $\mathcal{A}(z)$ with a randomness $r \in [0,1]$[14] by first computing the final state, which is known to be possible in classical exponential time, and then choosing a measurement result in accordance with the randomness r, where we assume that $\mathcal{A}$ performs only one measurement at the end of its execution without loss of generality. We denote this procedure by $\mathsf{Sim}_r(\mathcal{A}(z))$. If we consider many inputs $z \in Z$ and a corresponding random coin $R = \{r_z\} \in [0,1]^{|Z|}$, we just denote $\mathsf{Sim}_{r_z}(\mathcal{A}(z))$ by $\mathsf{Sim}_R(\mathcal{A}(z))$ for notational simplicity. We note that exactly the same procedure is possible for an oracle-aided quantum algorithm $\mathcal{A}^{|f\rangle}$ that accesses a quantum oracle $|f\rangle$ that computes a function f *if the simulator knows the whole truth table of f* since we can think of the combination of $\mathcal{A}$ and $|f\rangle$ as a single quantum algorithm. We also note that almost the same procedure is possible for an

[14] In an actual simulation, the randomness should be approximated by a rational number up to a sufficient precision. We just think of the randomness as a real number for simplicity.

oracle-aided quantum algorithm $\mathcal{A}^{|f\rangle,g}$ that accesses both a quantum oracle $|f\rangle$ and a classical oracle g if the simulator knows the whole truth table of f and g with the following modification. The difference from the case of a quantum oracle is that the oracle may not be unitary and we are no longer able to assume that the algorithm performs a measurement once, and it may perform a measurement in the middle of the computation. This can be dealt with by augmenting the amount of randomness used by the simulator so that fresh randomness is available in the simulation of each measurement.

Since the compression lemma (Lemma 4) holds even for an unbounded-time encoder and decoder that may share unbounded-size randomness, we can allow them to simulate a (oracle-aided) quantum algorithm classically in the above way.

Notations. In this section, we consider a random oracle with the domain $[K]\times[N]$ (or $[K]\times[N]\times[L]$ for the case of pqPRFs and pqMACs) and the codomain $[M]$. We omit to state a distribution of a random oracle $\mathcal{O}$ if that is uniformly chosen from the set of functions with the corresponding domain and codomain. We use a and x (or k for the case of pqPRFs and pqMACs) to represent elements of $[K]$ and $[N]$ respectively throughout the section, and often omit to state distributions when they are uniform. For example, we write $\Pr_{a,x}[f(a,x)=y]$ instead of $\Pr_{a\leftarrow[K],x\leftarrow[N]}[f(a,x)=y]$.

3.2 Function Inversion

The following theorem is the main result of this section.

Theorem 1. *Let $\mathcal{O}\in\mathsf{Func}([K]\times[N],[M])$ be a random oracle. Suppose that $\mathcal{A}$ is an oracle-aided quantum algorithm that takes an S-bit classical advice $\mathsf{st}_{\mathcal{O}}$ (that may depend on $\mathcal{O}$) as input, makes at most T oracle queries, and satisfies*

$$\Pr_{\mathcal{A},\mathcal{O},a,x}\left[\mathcal{O}(a,x)=\mathcal{O}(a,x') : \mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}},a,\mathcal{O}(a,x))\to x'\right]=\varepsilon.$$

Then it holds that

$$\varepsilon^2=\widetilde{O}\left(\frac{ST^2}{K\min(M,N)}+\frac{T^2N}{\min(M,N)^2}\right).$$

The main idea of the proof of this theorem is to compress the truth table of the random function into a smaller encoding by using an algorithm that inverts the function. Then by applying Lemma 4, we obtain a bound for the advantage to invert the function. Specifically, we encode a function into an encoding that consists of a partial truth table and information to recover the remaining part of the truth table similarly to [DGK17].

We also introduce another lemma, which can be seen as a variant of the above theorem. This lemma is used for proving lower bounds for other problems in the next sections. In this lemma, we give an upper bound for the probability that the event Find occurs when an adversary is given a punctured oracle on the correct answer. (See Section 2.2 for the definitions of Find and the punctured oracle.) This corresponds to [DGK17, Corollary 1], which gives a bound for the probability that an adversary *ever queries* the correct answer to the oracle in the classical case.

Lemma 5. *Let $\mathcal{O}\in\mathsf{Func}([K]\times[N],[M])$ be a random oracle. Suppose that $\mathcal{A}$ is an oracle-aided quantum algorithm that takes an S-bit classical advice $\mathsf{st}_{\mathcal{O}}$ (that may depend on $\mathcal{O}$) as input, and makes at most T oracle queries. Then it holds that*

$$\Pr_{\mathcal{A},\mathcal{O},a,x}\left[\mathsf{Find} : \mathcal{A}^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}},a,\mathcal{O}(a,x))\right]=O\left(\frac{ST^2}{KN}+\frac{T^2\log N}{N}\right).$$

Proof of Theorem 1. First, we consider an adversary $\mathcal{A}$ (which we call a *biased adversary*) that breaks the one-wayness in a slightly stronger sense. Namely, we assume that we have

$$\Pr_{\mathcal{O},a,x}\left[\Pr_{\mathcal{A}}[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}},a,\mathcal{O}(a,x))\to x'\wedge\mathcal{O}(a,x)=\mathcal{O}(a,x')]\geq c\right]\geq\varepsilon$$

for a fixed constant c. We will later show that we have

$$\varepsilon=\widetilde{O}\left(\frac{ST^2}{K\min(M,N)}+\frac{T^2N}{\min(M,N)^2}\right)$$

in this setting. For the time being, we assume that the above statement is true and prove the theorem. Suppose that there exists an algorithm $\mathcal{A}$ such that

$$\Pr_{\mathcal{A},\mathcal{O},a,x}\left[\mathcal{O}(a,x)=\mathcal{O}(a,x') : \mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}},a,\mathcal{O}(a,x))\to x'\right]=\varepsilon'.$$

By an averaging argument, at least an $(\varepsilon'/2)$-fraction of $(\mathcal{O},a,x)$ satisfies

$$\Pr_{\mathcal{A}}\left[\mathcal{O}(a,x)=\mathcal{O}(a,x') : \mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}},a,\mathcal{O}(a,x))\to x'\right]\geq\varepsilon'/2.$$

Applying the amplitude amplification [BHMT02] (see Lemma 10 in Appendix A), we obtain another algorithm $\mathcal{A}'$ that uses $\mathcal{A}$, $\mathcal{A}^{-1}$ and $\mathcal{O}$ as sub-routines $O(\varepsilon'^{-1/2})$ times and satisfies

$$\Pr_{\mathcal{A}'}\left[\mathcal{O}(a,x)=\mathcal{O}(a,x') : \mathcal{A}'^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}},a,\mathcal{O}(a,x))\to x'\right]=\Omega(1),$$

where we abuse the notation to use $\mathcal{A}$ and $\mathcal{A}^{-1}$ to mean the unitary part of $\mathcal{A}$ and its inverse, respectively. By the bound for the biased adversary, we have $\varepsilon'=\widetilde{O}\left(\frac{ST^2/\varepsilon'}{K\min(M,N)}+\frac{T^2N/\varepsilon'}{\min(M,N)^2}\right)$, which implies

$$\varepsilon'^2=\widetilde{O}\left(\frac{ST^2}{K\min(M,N)}+\frac{T^2N}{\min(M,N)^2}\right)$$

as desired.

Now it suffices to prove the bound for the biased adversary. For the sake of contradiction, we assume that we have

$$\varepsilon=\widetilde{\Omega}(ST^2/K\min(M,N)+T^2N/\min(M,N)^2). \tag{1}$$

Note that it particularly implies $CT^2\leq\varepsilon KN$ for a sufficiently large C since the tilde notation hides a non-negative degree polylogarithmic factor and $T^2/KN=O(ST^2/K\min(M,N))$ holds.[15] Here, to apply Lemma 2, we consider another adversary $\mathcal{B}$ that takes a list L of classical strings as an additional input and works as follows:

$\mathcal{B}^{|f\rangle}(\mathsf{st}_{\mathcal{O}},a,y,L)$: It runs $\mathcal{A}^{|f\rangle}(\mathsf{st}_{\mathcal{O}},a,y)$. Then $\mathcal{B}$ outputs 1 if the answer z of the algorithm $\mathcal{A}$ satisfies $(a,z)\in L$, and outputs 0 otherwise.

Note that the assumption on the biased adversary $\mathcal{A}$ can be rephrased as

$$\Pr_{\mathcal{O},a,x}\left[\Pr_{\mathcal{B}}[\mathcal{B}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}},a,\mathcal{O}(a,x),\mathcal{O}_a^{-1}(\mathcal{O}(a,x)))\to 1]\geq c\right]\geq\varepsilon$$

where $\mathcal{O}_a(x):=\mathcal{O}(a,x)$ and $\mathcal{O}_a^{-1}(y):=\{(a,x):\mathcal{O}(a,x)=y\}$. Here, we state a claim about the size of $\mathcal{O}_a^{-1}(y)$ whose proof can be found in Appendix B.1.

Claim 1. *Except for an $(\varepsilon/4)$-fraction of $\mathcal{O}\in\mathsf{Func}([K]\times[N],[M])$, we have*

$$|\mathcal{O}_a^{-1}(y)|=|\{x:\mathcal{O}_a(x)=y\}|=\widetilde{O}(N/\min(N,M))$$

for all $(a,y)\in[K]\times[M]$.

By an averaging argument, at least an $(\varepsilon/2)$-fraction of $f\in\mathsf{Func}([K]\times[N],[M])$ satisfies

$$\Pr_{a,x}\left[\Pr_{\mathcal{B}}[\mathcal{B}^{|f\rangle}(\mathsf{st}_f,a,f(a,x),f_a^{-1}(f(a,x)))\to 1]\geq c\right]\geq\varepsilon/2.$$

Combining this with Claim 1, at least an $(\varepsilon/4)$-fraction of $\mathsf{Func}([K]\times[N],[M])$, denoted by $\mathcal{F}$, simultaneously satisfies $\Pr_{\mathcal{B}}[\mathcal{B}^{|f\rangle}(\mathsf{st}_f,a,f(a,x),f_a^{-1}(f(a,x)))\to 1]\geq c$ and $|f_a^{-1}(y)|=\widetilde{O}(N/\min(N,M))$ for all $(a,y)\in[K]\times[M]$. We define $\beta=\widetilde{O}(N/\min(M,N))$ so that we have $|f_a^{-1}(y)|\leq\beta$ for all (a,y).

We fix an arbitrary function $f\in\mathcal{F}$ and write L to denote the set $f_a^{-1}(f(a,x))$. We will describe an encoder that compresses the truth table of f to generate an encoding that consists of a partial truth table of f and other information to recover the remaining part of the truth table by using the

[15] Looking ahead, this is used in the proof of Claim 2.

algorithm $\mathcal{A}$. What is non-trivial is that the decoder has to simulate the algorithm $\mathcal{A}$ that makes queries to f though it is given only a partial truth table of f as a part of the encoding. We will show that this is actually possible by using the SC-O2H lemma (Lemma 2) below.

A public randomness r shared by the encoder and decoder (in Lemma 4) specifies R_1 and R_2 as explained below. A set $R_1 \subset [K] \times [M]$ is chosen so that each $(a, y) \in [K] \times [M]$ is included in R_1 with probability $d/T(T+1)$ for a fixed constant $d \leq c^2/1280$. Let $R_{(a,x)} := R_1 \setminus \{(a, f(a, x))\}$. For a set $S \subset [K] \times [M]$, we define $S_a := \{y \in [M] : (a, y) \in S\}$ and $f^{-1}(S) := \cup_{a \in [K]} f_a^{-1}(S_a)$.

We say that $(a, x) \in I$ is good if both

$$\text{(A) } (a, f(a, x)) \in R_1,$$

$$\text{(B) } \Pr[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus f^{-1}(R_{(a,x)})}(\mathsf{st}_f, a, f(a, x), L)] \leq \frac{c^2}{16(T+1)}$$

hold. We denote the set of good elements by G. Note that if we have $f(a, x) = f(a, x')$, then we have $(a, x) \in G$ if and only if $(a, x') \in G$.

Here, we state a claim that states that G is "not too small" with high probability whose proof is given in Appendix B.1.

Claim 2. $\Pr_{R_1}[|G| \geq \delta \varepsilon KN/T^2] \geq 0.8$ *for some constant* $\delta > 0$.

For $y \in [M]$, we define a function $g_y : [K] \times [N] \to [M]$ by

$$g_y(z) = \begin{cases} f(z), & \text{if } z \in ([K] \times [N]) \setminus f^{-1}(R_1), \\ y, & \text{otherwise.} \end{cases}$$

By the SC-O2H lemma (Lemma 2), for any $(a, x) \in G$, it holds that

$$\left| \Pr_{\mathcal{B}}[\mathcal{B}^{|f\rangle}(\mathsf{st}_f, a, f(a, x), L) \to 1] - \Pr_{\mathcal{B}}[\mathcal{B}^{|g_{f(a,x)}\rangle}(\mathsf{st}_f, a, f(a, x), L) \to 1] \right|$$
$$\leq 2\sqrt{(T+1) \cdot \Pr[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus f^{-1}(R_{(a,x)})}(\mathsf{st}_f, a, f(a, x), L)]} \leq c/2,$$

where we used the condition (B) for deriving the last inequality. Since we have $\Pr_{\mathcal{B}}[\mathcal{B}^{|f\rangle}(\mathsf{st}_f, a, f(a, x), L) \to 1] \geq c$, we have

$$\Pr_{\mathcal{B}}[\mathcal{B}^{|g_{f(a,x)}\rangle}(\mathsf{st}_f, a, f(a, x), L) \to 1] \geq \frac{c}{2}$$

for any $(a, x) \in G$. It is easy to see that this can be rephrased as

$$\Pr_{\mathcal{A}}[\mathcal{A}^{|g_{f(a,x)}\rangle}(\mathsf{st}_f, a, f(a, x)) \to x' \wedge f(a, x) = f(a, x')] \geq c/2.$$

The randomness R_2, which is another random coin specified by r, is used for the simulation

$$\mathsf{Sim}_{R_2}\left(\mathcal{A}^{|g_{f(a,x)}\rangle}(\mathsf{st}_f, a, f(a, x))\right)$$

of $\mathcal{A}^{|g_{f(a,x)}\rangle}(\mathsf{st}_f, a, f(a, x))$.[16] It outputs x' such that $f(a, x) = f(a, x')$ with probability at least $c/2$ over the choice of R_2. Then for at least a $(c/4)$-fraction of R_2, the simulation of $\mathcal{A}$ with oracle access to $|g_{f(a,x)}\rangle$ instead of $|f\rangle$ outputs a correct preimage for at least a $(c/4)$-fraction of (a, x). More precisely, for at least a $(c/4)$-fraction of R_2, the following condition is satisfied:

($*$) There exists at least a $(c/4)$-fraction of good elements (a, x), which we denote by X, such that we have

$$\mathsf{Sim}_{R_2}\left(\mathcal{A}^{|g_{f(a,x)}\rangle}(\mathsf{st}_f, a, f(a, x))\right) \to x' \text{ such that } f(a, x) = f(a, x')$$

for all $(a, x) \in X$.

[16] Specifically, R_2 consists of independent random coins $r_2(a, y)$ for each $(a, y) \in [K] \times [M]$ to simulate $\mathcal{A}^{|g_y\rangle}(\mathsf{st}_f, a, y)$.

We again remark that $(a, x) \in X$ and $(a, x') \in X$ are equivalent if $f(a, x) = f(a, x')$. We say that (R_1, R_2) is good if the following three conditions all hold:

1) $|G| \geq \delta\varepsilon KN/T^2$, 2) the condition $(*)$, 3) $|R_1| = \Theta(\varepsilon KM/T^2)$.

By Claim 2, the first statement holds with probability at least 0.8 (over the choice of R_1), and the second holds with probability at least $c/4$ (over the choice of R_2 for any fixed R_1) as discussed above, and the last holds with probability $1 - o(1)$ by the Chernoff bound. Therefore, the probability that (R_1, R_2) is good is $\Omega(1)$. When (R_1, R_2) is good, we clearly have $|X| = \Omega(\varepsilon KN/T^2)$ by definition.

Now we are ready to explicitly describe the encoder and decoder for f. Note that the decoder will correctly recover f as long as (R_1, R_2) is good. The encoder induces R_1, R_2 from the given public randomness. The encoder computes $X_a := \{x : (a, x) \in X\}$, $Y_a := \{y : y = f(a, x) \text{ for } x \in X_a\}$, $Y := \cup_{a\in[K]}\{(a, y) : y \in Y_a\}$, and $R_a = R_1 \cap (\{a\} \times [M])$ for all $a \in [K]$. Then, $|Y| \geq |X|/\beta$ holds by the definition of β.

For each $a \in [K]$, the encoder computes a set $Z_a \subset [N]$ as the set consisting of outputs of simulations $\mathsf{Sim}_{R_2}\left(\mathcal{A}^{|g_y\rangle}(\mathsf{st}_f, a, y)\right)$ for all $y \in Y_a$. We note that Z_a is well-defined since the simulation is deterministic once R_2 is fixed. Let $Z := \cup_{a\in[K]}\{(a, z) : z \in Z_a\}$. Clearly, we have $|Z_a| = |Y_a|$ and $|Z| = |Y|$. Now the function $f \in \mathcal{F}$ is encoded as follows, given the public randomness R_1, R_2.

- The advice string st_f: S bits.
- The description of Z_a with its size for each $a \in [K]$: $\log N + \log \binom{N}{|Z_a|}$ bits.
- The description of Y_a with its size for each $a \in [K]$: $\log M + \log \binom{|R_a|}{|Y_a|}$ bits.
- The values of f on $([K] \times [N]) \setminus Z$: $(KN - |Z|) \log M$ bits.

The values are encoded in the lexicographic order of their inputs. The size of the third component is derived by observing $Y_a \subset R_a$. Given this encoding and random sets R_1, R_2, the decoder fills the truth table of f as follows:

1. Reconstruct st_f, Y_a, Z_a, Y, and Z.
2. Fill the truth table of f on $([K] \times [N]) \setminus Z$.
3. Recover the set $f^{-1}(R_1) \subset [K] \times [N]$: this is done by 1) including all elements of Z (which are definitely in $f^{-1}(R_1)$ since they are good) and 2) including all $(a, x) \notin Z$ such that $f(a, x) \in R_1$, which can be checked by using the partial truth table on $([K] \times [N]) \setminus Z$.
4. Recover the function values on Z. This step is done by simulating the algorithm $\mathcal{A}$. More precisely, for each $(a, y) \in Y_a$, the decoder executes the simulation $\mathsf{Sim}_{R_2}\left(\mathcal{A}^{|g_y\rangle}(\mathsf{st}_f, a, y)\right)$ to obtain an output z and set the value of f on (a, z) to be y. By the definition of Z, this simulation correctly recovers the function values if the randomness (R_1, R_2) is good. Note that since the decoder has already recovered $f^{-1}(R_1)$, the decoder can simulate the function g_y.

The decoder successfully recovers f as long as (R_1, R_2) is good, which happens with probability $\Omega(1)$. The overall encoding size is

$$\begin{aligned} S + K\log N + K\log M + \sum_{a\in[K]}\left(\log\binom{N}{|Z_a|} + \log\binom{|R_a|}{|Y_a|}\right) + (KN - |Z|)\log M \\ \geq \log(\varepsilon M^{KN}) + O(1) = KN\log M + \log\varepsilon + O(1), \end{aligned} \tag{2}$$

by the compression lemma (Lemma 4). Since we have $\log\binom{a}{b} \leq b\log(ea/b)$, $|Z_a| = |Y_a|$, and $|Z| = |Y|$, we obtain

$$\begin{aligned} &\sum_{a\in[K]}\log\binom{N}{|Y_a|} + \sum_{a\in[K]}\log\binom{|R_a|}{|Y_a|} - |Y|\log M \\ &\leq \sum_{a\in[K]}|Y_a|\log\left(\frac{eN}{|Y_a|}\right) + \sum_{a\in[K]}|Y_a|\log\left(\frac{e|R_a|}{|Y_a|}\right) - |Y|\log M \\ &\leq |Y|\log\left(\frac{eKN}{|Y|}\right) + |Y|\log\left(\frac{e|R_1|}{|Y|}\right) - |Y|\log M \\ &= |Y|\log\left(\frac{e^2KN|R_1|}{M|Y|^2}\right), \end{aligned}$$

where the second inequality is obtained by using log-concavity (or Jensen's inequality for log with weights $|Y_a|$ and $|R_a|$.) Combining this bound with the inequality (2), we obtain

$$S + K\log(MN) \geq |Y|\log\left(\frac{M|Y|^2}{e^2KN|R_1|}\right) + \widetilde{O}(1), \tag{3}$$

where we used (1) to remove the $\log\varepsilon$ term in the right-hand side. Using $|X| = \Omega(\varepsilon KN/T^2)$, $|Y| \geq |X|/\beta$, and $|R_1| = \Theta(\varepsilon KM/T^2)$, we obtain $|Y|^2/|R_1| = \Omega(\varepsilon KN^2/MT^2\beta^2)$. This implies $|Y|^2/|R_1| \geq D\varepsilon KN^2/MT^2\beta^2$ for some constant D. If $D\varepsilon N/T^2\beta^2 \leq e^3$ holds, then we have $\varepsilon \leq (e^3T^2N/D)\cdot(\beta/N)^2 = \widetilde{O}(T^2N/\min(M,N)^2)$ since $\beta/N = \widetilde{O}(1/\min(M,N))$. Otherwise, we have $\frac{M|Y|^2}{e^2KN|R_1|} \geq \frac{M}{e^2KN}\cdot\frac{D\varepsilon KN^2}{MT^2\beta^2} \geq e$. Putting this bound and the bound $|Y| \geq |X|/\beta = \Omega(\varepsilon KN/T^2\beta)$ into (3), we obtain

$$O\left(S + K\log\max(M,N)\right) \geq |Y| + \tilde{O}(1) = \Omega\left(\frac{\varepsilon KN}{\beta T^2}\right),$$

which implies $\varepsilon = \widetilde{O}\left(\frac{ST^2}{K\min(M,N)} + \frac{T^2}{\min(M,N)}\right)$. Combining the two cases, we obtain

$$\varepsilon = \widetilde{O}\left(\frac{ST^2}{K\min(M,N)} + \frac{T^2N}{\min(M,N)^2}\right).$$

□

Proof sketch of Lemma 5. The proof is very similar to the proof of Theorem 1 except some parts. The main differences are

1. the algorithm does not output an element in $[N]$, and
2. we cannot apply the amplitude amplification since it uses a semi-classical oracle that is not unitary.

The first problem is resolved by considering another algorithm $\mathcal{B}$ that outputs the query register of the semi-classical oracle whenever Find occurs, and the second problem is circumvented by amplifying the success probability just by a parallel repetition. We note that there are two technical differences that make the proof easier: we choose the random coin R as a subset of $[K]\times[N]$ instead of $[K]\times[M]$ and need not consider a counterpart of Claim 1. The detailed proof can be found in Appendix B.2. □

3.3 Pseudorandom Generators

In this section, we prove that a random function is a secure PRG even if we allow an adversary to make quantum queries to the function and to obtain a classical advice string. Our result is stated as follows.

Theorem 2. *Let $\mathcal{O} \in \mathsf{Func}([K]\times[N],[M])$ be a random oracle. Suppose that $\mathcal{A}$ is an oracle-aided quantum algorithm that takes an S-bit classical advice $\mathsf{st}_{\mathcal{O}}$ (that may depend on $\mathcal{O}$) as input, and makes at most T oracle queries. Then it holds that*

$$\left|\Pr_{\mathcal{A},\mathcal{O},a,x}\left[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}},a,\mathcal{O}(a,x)) \to 1\right]\right| - \left|\Pr_{\mathcal{A},\mathcal{O},a,y}\left[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}},a,y) \to 1\right]\right|$$
$$= \widetilde{O}\left(\sqrt[6]{\frac{ST^4}{KN} + \frac{T^4}{N}}\right),$$

where y is uniform in $[M]$.

For proving Theorem 2, we need the following lemma, which can be seen as a security bound for a quantum average case version of Yao's box problem [Yao90]. We note that the classical average case version was proven in [DTT10, Lemma 8.4] and quantum worst-case version was proven in [NABT15, Theorem 1], neither of which suffices for our purpose.

Lemma 6. *Let $\mathcal{F} \subset \mathsf{Func}([N], \{0,1\})$ be a set of functions. Suppose that $\mathcal{A}$ is an oracle-aided quantum algorithm that takes an S-bit classical advice st_f (that may depend on $f \in \mathcal{F}$) as input, makes at most T oracle queries, has query magnitudes 0 on its second input (i.e. x) for all queries, and satisfies*

$$\Pr_{\mathcal{A},x}[\mathcal{A}^{|f\rangle}(\mathsf{st}_f, x) \to f(x)] \geq \frac{1}{2} + \varepsilon$$

for all $f \in \mathcal{F}$. Then there is a pair of an encoder and decoder for the truth tables of functions in $\mathcal{F}$ with recovery probability $\Omega(\varepsilon^5/T^2)$ and encoding length at most $S + N - \Omega(\varepsilon^6 N/T^2)$. In particular, this implies $\varepsilon^6 = O(ST^2/N)$ for $\mathcal{F} = \mathsf{Func}([N], \{0,1\})$.

This lemma can be proven similarly to its classical counterpart in [DTT10, Lemma 8.4] except for some technical issues as discussed in Section 1.3. The proof of this lemma can be found in Appendix B.3. Now, we are ready to prove Theorem 2.

Proof of Theorem 2. We first sketch the outline of the proof by the following diagram:

$$\begin{aligned}
& & p_0 &:= \Pr_{\mathcal{A},\mathcal{O},a,x}[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x)) \to 1] \\
&\overset{\text{O2H+Lemma 5}}{\approx} & p_1 &:= \Pr_{\mathcal{A},\mathcal{O},a,x}[\neg\mathsf{Find} : \mathcal{A}^{|\mathcal{O}\rangle \setminus \{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x)) \to 1] \\
&\overset{\text{Lemma 6}}{\approx} & p_2 &:= \Pr_{\mathcal{A},\mathcal{O},a,x}[\neg\mathsf{Find} : \mathcal{A}^{|\mathcal{O}\rangle \setminus \{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, y) \to 1] \\
&\overset{\text{O2H+Lemma 5}}{\approx} & p_3 &:= \Pr_{\mathcal{A},\mathcal{O},a,x}[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, a, y) \to 1].
\end{aligned}$$

We assume that M is a power of 2 for simplicity.

Step 1. $|p_0 - p_1| = \widetilde{O}\left(\sqrt[4]{\frac{ST^4}{KN} + \frac{T^4}{N}}\right)$

This is simply proven by using the SC-O2H lemma. More precisely, by Lemma 2,

$$|p_0 - p_1| \leq \sqrt{(T+1) \Pr_{\mathcal{A},\mathcal{O},a,x}\left[\mathsf{Find} : \mathcal{A}^{|\mathcal{O}\rangle \setminus \{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x))\right]}$$

holds, which is bounded by $\widetilde{O}\left(\sqrt[4]{\frac{ST^4}{KN} + \frac{T^4}{N}}\right)$ by Lemma 5.

Step 2. $|p_2 - p_3| = \widetilde{O}\left(\sqrt[4]{\frac{ST^4}{KN} + \frac{T^4}{N}}\right)$

This is exactly the same as Step 1.

Step 3. $|p_1 - p_2| = \widetilde{O}\left(\sqrt[6]{\frac{ST^2}{KN}}\right)$

First, we consider an oracle-aided quantum algorithm $\mathcal{B}$ that uses $\mathcal{A}$ as a sub-routine as follows.

$\mathcal{B}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, a, x, y)$**:** It runs $\mathcal{A}^{|\mathcal{O}\rangle \setminus \{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, y)$. If the event Find occurs w.r.t. the running of $\mathcal{A}$, $\mathcal{B}$ immediately halts and returns 0. Otherwise, $\mathcal{B}$ returns what $\mathcal{A}$ outputs.

We note that $\mathcal{B}$ can simulate the oracle $|\mathcal{O}\rangle \setminus \{(a,x)\}$ for $\mathcal{A}$ since it knows the punctured point (a,x). Moreover, $\mathcal{B}$'s query magnitude on (a,x) is 0 since before making a query to $\mathcal{O}$, it performs a partial measurement to check if the query is equal to (a,x) and immediately aborts if so by the definition of the punctured oracle. By the construction of $\mathcal{B}$, it is easy to see that

$$\begin{aligned}
p_1 &= \Pr_{\mathcal{B},\mathcal{O},a,x}[\mathcal{B}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, a, x, \mathcal{O}(a,x)) \to 1], \\
p_2 &= \Pr_{\mathcal{B},\mathcal{O},a,x}[\mathcal{B}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, a, x, y) \to 1].
\end{aligned}$$

Let $|p_1 - p_2| = \varepsilon$. By Yao's equivalence of pseudorandomness to unpredictability [Yao82], there exists an $i \in [\log M]$, an oracle-aided quantum algorithm $\mathcal{C}$ whose query magnitude at (a,x) is 0, and an advice string $\widetilde{\mathsf{st}}_{\mathcal{O}}$ that have at most $S+1$ bits such that

$$\Pr_{\mathcal{C},\mathcal{O},a,x}[\mathcal{C}^{|\mathcal{O}\rangle}(\widetilde{\mathsf{st}}_{\mathcal{O}}, a, x, \mathcal{O}_1(a,x), \cdots, \mathcal{O}_{i-1}(a,x)) \to \mathcal{O}_i(a,x)] \geq \frac{1}{2} + \frac{\varepsilon}{\log M},$$

where $\mathcal{O}_i(a,x)$ denotes the i-th bit of $\mathcal{O}(a,x)$.

If we define $T_{\mathcal{O}}$ as a partial truth table of $\mathcal{O}$ that specifies the first $i-1$ bits of $\mathcal{O}(a,x)$ for all $(a,x) \in [K] \times [N]$, then there is another algorithm $\mathcal{D}$ (that just runs $\mathcal{C}$ once) whose query magnitude on (a,x) is 0 that satisfies

$$\Pr_{\mathcal{D},\mathcal{O},a,x}[\mathcal{D}^{|\mathcal{O}\rangle}(\widetilde{\mathsf{st}}_{\mathcal{O}}, T_{\mathcal{O}}, a, x) \to \mathcal{O}_i(a,x)] \geq \frac{1}{2} + \frac{\varepsilon}{\log M}.$$

Then at least an $(\varepsilon/\log M)$-fraction of $\mathcal{O}$ satisfies

$$\Pr_{\mathcal{D},a,x}[\mathcal{D}^{|\mathcal{O}\rangle}(\widetilde{\mathsf{st}}_{\mathcal{O}}, T_{\mathcal{O}}, a, x) \to \mathcal{O}_i(a,x)] \geq \frac{1}{2} + \frac{\varepsilon}{2\log M}.$$

By Lemma 6, there exists a pair of an encoder and decoder for this fraction of functions with the success probability $\Omega(\varepsilon^5/T^2\log^5 M)$ and encoding size

$$KN + KN \cdot (\log M - 1) + S + O(1) - \Omega\left(\frac{\varepsilon^5 KN}{T^2 \log^6 M}\right).$$

By Lemma 4, it holds that

$$KN\log M + S + O(1) - \Omega\left(\frac{\varepsilon^6 KN}{T^2\log^6 M}\right) \geq \log\left(\frac{\varepsilon M^{KN}}{\log M}\right) + \log(\varepsilon^5/T^2\log^5 M)$$

or $O\left(S + \log\left(\frac{T^2\log^6 M}{\varepsilon^6}\right)\right) \geq \Omega\left(\frac{\varepsilon^6 KN}{T^2\log^6 M}\right)$, which implies $\varepsilon = \widetilde{O}\left(\sqrt[6]{\frac{ST^2}{KN}}\right)$ as desired.[17]

Overall, we obtain $|p_0 - p_3| = \widetilde{O}\left(\sqrt[6]{\frac{ST^4}{KN} + \frac{T^4}{N}}\right)$. □

3.4 Post-Quantum Pseudorandom Functions

The main theorem of this subsection is that random oracles are secure pqPRFs in the QROM-AI, which is formally stated as follows.

Theorem 3. *Let $\mathcal{O} \in \mathsf{Func}([K] \times [N] \times [L], \{0,1\})$ be a random oracle. Suppose that $\mathcal{A}$ is an oracle-aided quantum algorithm that takes an S-bit classical advice $\mathsf{st}_{\mathcal{O}}$ (that may depend on $\mathcal{O}$) as input, and makes at most T (quantum) queries to the oracle $\mathcal{O}$ and at most Q classical queries to the other oracle. Then it holds that*

$$\left|\Pr_{\mathcal{A},\mathcal{O},a,k}\left[\mathcal{A}^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}}, a) \to 1\right] - \Pr_{\mathcal{A},\mathcal{O},a,F}\left[\mathcal{A}^{|\mathcal{O}\rangle,F}(\mathsf{st}_{\mathcal{O}}, a) \to 1\right]\right| = \widetilde{O}\left(\sqrt[4]{\frac{ST^4}{KN} + \frac{T^4}{N}} + Q\sqrt[6]{\frac{ST^2}{KN}}\right),$$

where F is uniform in $\mathsf{Func}([L], \{0,1\})$.

Although Theorem 3 is similar to Theorem 2, we require the following lemma, which is a function variant of Lemma 6. The classical counterpart of this lemma is implicitly proven in [DGK17, Theorem 7] for a similar purpose. To show this lemma, we should note that the simulations for the decoder are deterministic, and thus the encoder knows all required queries to the second oracle for the decoder's simulation in advance. Except this, the proof is essentially the same as that of Lemma 6. We defer the proof of this lemma to Appendix B.4

Lemma 7. *Let $\mathcal{O} \in \mathsf{Func}([K]\times[N]\times[L], \{0,1\})$ be a random oracle. For any oracle-aided quantum algorithm $\mathcal{A}$ with a set of S-bit classical advice $\{\mathsf{st}_{\mathcal{O}}\}_{\mathcal{O}}$ that makes at most T oracle queries to the oracle $\mathcal{O}$ satisfying*

$$\Pr_{\mathcal{A},\mathcal{O},a,k}[\mathcal{A}^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}}, a, k) \to (m,t) \wedge t = \mathcal{O}(a,k,m)] \geq \frac{1}{2} + \varepsilon,$$

where $\mathcal{A}$ has the query magnitude 0 for $\{(a,k,\cdot)\}$ to its first oracle and never queries m to its second oracle, we have

$$\varepsilon^6 = O(ST^2/KN).$$

[17] More concretely, $\varepsilon^6 > CST^2\log^6 M(1+\log KN)/KN$ for sufficiently large C implies contradiction.

Proof of Theorem 3. The outline of the proof is as follows.

$$
\begin{aligned}
& & p_0 &:= \Pr_{\mathcal{A},\mathcal{O},a,k}\left[\mathcal{A}^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a)\to 1\right]\\
&\overset{\text{O2H+Lemma 5}}{\approx} & p_1 &:= \Pr_{\mathcal{A},\mathcal{O},a,k}\left[\neg\mathsf{Find}:\mathcal{A}^{|\mathcal{O}\rangle\setminus\{(a,k,\cdot)\},\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a)\to 1\right]\\
&\overset{\text{Lemma 7}}{\approx} & p_2 &:= \Pr_{\mathcal{A},\mathcal{O},a,F}\left[\neg\mathsf{Find}:\mathcal{A}^{|\mathcal{O}\rangle\setminus\{(a,k,\cdot)\},F}(\mathsf{st}_{\mathcal{O}},a)\to 1\right]\\
&\overset{\text{O2H+Lemma 5}}{\approx} & p_3 &:= \Pr_{\mathcal{A},\mathcal{O},a,F}\left[\mathcal{A}^{|\mathcal{O}\rangle,F}(\mathsf{st}_{\mathcal{O}},a)\to 1\right]
\end{aligned}
$$

where $\{(a,k,\cdot)\}$ denotes the set $\{(a,k,m)\}_{m\in[L]}$.

Step 1. $|p_0-p_1|=\widetilde{O}\left(\sqrt[4]{\frac{ST^4}{KN}+\frac{T^4}{N}}\right)$

This is proved by using the semi-classical O2H lemma.[18] More precisely, by Lemma 2,

$$|p_0-p_1|\le\sqrt{(T+1)\Pr_{\mathcal{A},\mathcal{O},a,k}\left[\mathsf{Find}:\mathcal{A}^{|\mathcal{O}\rangle\setminus\{(a,k,\cdot),\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a)\right]}$$

holds. If we define $\mathcal{O}':[K]\times[N]\to[M]^L$ by $\mathcal{O}'(a,k):=\{\mathcal{O}(a,k,m)\}_{m\in[L]}$ and apply Lemma 5 for the oracle $\mathcal{O}'$, then we obtain

$$\Pr_{\mathcal{A},\mathcal{O},a,k}\left[\mathsf{Find}:\mathcal{A}^{|\mathcal{O}\rangle\setminus\{(a,k,\cdot)\},\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a)\right]\le O\left(\sqrt[2]{\frac{ST^4}{KN}+\frac{T^4\log N}{N}}\right).$$

This implies $|p_0-p_1|=\widetilde{O}\left(\sqrt[4]{\frac{ST^4}{KN}+\frac{T^4}{N}}\right)$.

Step 2. $|p_2-p_3|=\widetilde{O}\left(\sqrt[4]{\frac{ST^4}{KN}+\frac{T^4}{N}}\right)$

This step is essentially the same as Step 1.

Step 3. $|p_1-p_2|=O\left(Q\sqrt[6]{\frac{ST^2}{KN}}\right)$

We consider an oracle-aided quantum algorithm $\mathcal{B}$ that utilizes $\mathcal{A}$ as follows.

$\mathcal{B}^{|\mathcal{O}\rangle,G}(\mathsf{st}_{\mathcal{O}},a,k)$**:** It runs $\mathcal{A}^{|\mathcal{O}\rangle\setminus\{(a,k,\cdot)\},G}(\mathsf{st}_{\mathcal{O}},a)$. If the event Find occurs w.r.t. the running of $\mathcal{A}$, $\mathcal{B}$ immediately halts and returns 0. Otherwise, $\mathcal{B}$ returns what $\mathcal{A}$ outputs.

Since k is given as input to $\mathcal{B}$, $\mathcal{B}$ can simulate the oracle $|\mathcal{O}\rangle\setminus\{(a,k,\cdot)\}$ for $\mathcal{A}$. Note that $\mathcal{B}$'s query magnitude on $(a,k,\cdot)$ is 0 since before making a query to $\mathcal{O}$, it performs a partial measurement to check if the query is included in $(a,k,\cdot)$, and immediately aborts if so. We can easily see that

$$
\begin{aligned}
p_1 &= \Pr_{\mathcal{B},\mathcal{O},a,k}[\mathcal{B}^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a,k)\to 1],\\
p_2 &= \Pr_{\mathcal{B},\mathcal{O},a,F}[\mathcal{B}^{|\mathcal{O}\rangle,F}(\mathsf{st}_{\mathcal{O}},a,k)\to 1].
\end{aligned}
$$

Let $|p_1-p_2|=\varepsilon$. We will directly give a bound for ε. By Yao's equivalence of pseudorandomness to unpredictability [Yao82], there exist $i\in[Q]$, an oracle-aided quantum algorithm $\mathcal{C}$ whose query magnitude at $(a,k,\cdot)$ is 0, and advice strings $\{\widetilde{\mathsf{st}}_{\mathcal{O}}\}$ that have at most $S+1$ bits such that

$$\Pr_{\mathcal{C},\mathcal{O},a,k}[\mathcal{C}^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\widetilde{\mathsf{st}}_{\mathcal{O}},a,k)\to(m,t)\wedge t=\mathcal{O}(a,k,m)]\ge\frac{1}{2}+\frac{\varepsilon}{Q},$$

where $\mathcal{C}$ does not query m to its second oracle. By Lemma 7, we obtain $\varepsilon/Q=O(\sqrt[6]{ST^2/KN})$ as desired. ∎

[18] Though Lemma 2 does not consider an additional oracle, we can apply it simply by considering a slightly modified algorithm that works similarly to $\mathcal{A}$ except that it is given the truth table of $\mathcal{O}(a,k,\cdot)$ as input instead of as an oracle.

3.5 Post-Quantum MACs

The main theorem of this subsection is that random oracles are secure pqMACs in the QROM-AI, which is formally stated as follows.

Theorem 4. *Let $\mathcal{O} \in \mathsf{Func}([K] \times [N] \times [L], [M])$ be a random oracle. Suppose that $\mathcal{A}$ is an oracle-aided quantum algorithm that takes an S-bit classical advice $\mathsf{st}_{\mathcal{O}}$ (that may depend on $\mathcal{O}$) as input, and makes at most T oracle queries to the oracle $\mathcal{O}$. Then it holds that*

$$\Pr_{\mathcal{A},\mathcal{O},a,k}\left[\mathcal{A}^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a) \to (m,t) \wedge \mathcal{O}(a,k,m)=t\right] = \widetilde{O}\left(\sqrt[3]{\frac{ST^4}{KN}+\frac{T^4}{N}+\frac{1}{M}}\right)$$

where $\mathcal{A}$ never queries m to its second oracle.

Proof of Theorem 4. We first consider another algorithm $\mathcal{A}'$ such that

$\mathcal{A}'^{|f\rangle,g}(\mathsf{st}_{\mathcal{O}},a)$ It runs $\mathcal{A}^{|f\rangle,g}(\mathsf{st}_{\mathcal{O}},a)$. For the output $z=(m,t)$ of $\mathcal{A}$, it queries m to the second oracle g. If $t=g(m)$ then outputs 1, and 0 otherwise.

Then it holds that

$$\Pr\left[\mathcal{A}^{|f\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a) \to (m,t) \wedge \mathcal{O}(a,k,m)=t\right] = \Pr\left[\mathcal{A}'^{|f\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a) \to 1\right]$$

for any function f. Applying Lemma 2, we obtain

$$\begin{aligned}\sqrt{\Pr_{\mathcal{A}',\mathcal{O},a,k}\left[\mathcal{A}'^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a)\to 1\right]} &\\ \leq \sqrt{\Pr_{\mathcal{A}',\mathcal{O},a,k}\left[\neg\mathsf{Find}:\mathcal{A}'^{|\mathcal{O}\rangle\setminus\{(a,k,\cdot)\},\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a)\to 1\right]}&\\ +\sqrt{(T+1)\Pr_{\mathcal{A}',\mathcal{O},a,k}\left[\mathsf{Find}:\mathcal{A}'^{|\mathcal{O}\rangle\setminus\{(a,k,\cdot)\},\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a)\right]}&\end{aligned}$$

where $\{(a,k,\cdot)\}$ denotes the set $\{(a,k,m)\}_{m\in[L]}$. We will bound two terms of the right-hand side in order. First, we define $\mathcal{O}' : [K]\times[N] \to [M]^L$ by $\mathcal{O}'(a,k) := \{\mathcal{O}(a,k,m)\}_{m\in[L]}$ and apply Lemma 5 for the oracle $\mathcal{O}'$ to obtain

$$\Pr_{\mathcal{A}',\mathcal{O},a,k}\left[\mathsf{Find}:\mathcal{A}'^{|\mathcal{O}\rangle\setminus\{(a,k,\cdot)\},\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a)\right] \leq O\left(\sqrt[2]{\frac{ST^2}{KN}+\frac{T^2\log N}{N}}\right).$$

This can be used to bound the second term. Next, we consider another algorithm $\mathcal{B}$ that proceeds as follows.

$\mathcal{B}^{|f\rangle,g}(\mathsf{st},a,k)$**:** It runs $\mathcal{A}'^{|f\rangle\setminus\{(a,k,\cdot)\},g}(\mathsf{st}_{\mathcal{O}},a)$. If the event Find occurs w.r.t. the running of $\mathcal{A}'$, $\mathcal{B}$ immediately halts and returns 0. Otherwise, $\mathcal{B}$ returns what $\mathcal{A}'$ outputs.

Note that the query magnitude on $\{(a,k,\cdot)\}$ is 0 for algorithm $\mathcal{B}$. Then the first term is equal to

$$\Pr_{\mathcal{B},\mathcal{O},a,k}\left[\mathcal{B}^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a,k)\to 1\right] =: \varepsilon.$$

Moreover, the total query magnitude of $\mathcal{B}$ on (a,k,m) is 0 for all $m\in[L]$. By applying averaging argument twice, there are a set of functions $\mathcal{F}\subset \mathsf{Func}([K]\times[N]\times[L],[M])$ with size at least $\varepsilon/2\cdot M^{KNL}$ and a set I of $(a,k)\in[K]\times[N]$ with size $|I|\geq \varepsilon/4\cdot KN$ such that for all $f\in\mathcal{F}$ and $(a,k)\in I$ it holds that

$$\Pr_{\mathcal{B}}\left[\mathcal{B}^{|f\rangle,f(a,k,\cdot)}(\mathsf{st}_f,a,k)\to 1\right] \geq \varepsilon/4.$$

Fix $f\in\mathcal{F}$. We choose a random set $R_1\subset[K]\times[N]$ such that each (a,k) is included in R_1 with probability $c\varepsilon/T(T+1)$ for a constant $c\leq 1/5120$. We say $(a,k)\in I$ is good if

$$(A)\ (a,k)\in R_1 \text{ and } (B)\ \Pr[\mathsf{Find}:\mathcal{B}^{|f\rangle\setminus(R_1\setminus\{(a,k)\})\times[L],f(a,k,\cdot)}(\mathsf{st}_f,a,k)] \leq \frac{160c\varepsilon}{T+1}$$

hold simultaneously. We denote a set of good elements by G.

Claim 3. $\Pr_{R_1}[|G| \geq \delta\varepsilon^2 KN/T^2] \geq 0.8$ *for some constant* $\delta > 0$.

The proof is deferred to Appendix B.5. Then, for good (a,k), it holds that

$$\Pr_{\mathcal{B}}\left[\mathcal{B}^{|g\rangle\setminus R_1, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1\right] \geq \varepsilon/16$$

by Lemma 2, where $g(a,k,m)$ is defined by $g(a,k,m) := f(a,k,m)$ for $(a,k,m) \notin R_1 \times [L]$, and $g(a,k,m) := 0$ otherwise. Note that this is equal to

$$\begin{aligned}&\Pr\left[\mathcal{A}'^{|g\rangle\setminus R_1, \mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}}, a) \to 1\right] \\ &= \Pr\left[\mathcal{A}^{|g\rangle\setminus R_1, \mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}}, a) \to (m,t) \wedge \mathcal{O}(a,k,m) = t\right]\end{aligned}$$

since $(|g\rangle \setminus R_1) \setminus \{(a,k,\cdot)\}$ is equivalent to $|g\rangle \setminus R_1$.

Now fix the simulation coin R_2 for

$$\mathsf{Sim}_{R_2}\left(\mathcal{A}^{|g\rangle\setminus R_1, \mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}}, a)\right).$$

Note that this randomness includes coins for intermediate measurements as well. Then, with the probability at least $\varepsilon/32$ over the choice of R_2 it holds that at least $(\varepsilon/32)$-fraction of the simulations for good elements outputs the correct answer. We say a random coin is good if $|R_1| = \Theta(\varepsilon KN/T^2)$ and the above argument holds simultaneously, and we say (a,k) is very good (for a good random coin) if the simulation of $\mathcal{B}$ with input (a,k) outputs a correct answer. For a good coin and the set of very good elements V, it holds that $|V| = \Omega(\varepsilon^3 KN/T^2)$. At least $(\varepsilon/100)$-fraction of random coins is good.

We encode f using the algorithm $\mathcal{A}$. As in Theorem 3, the simulation of a quantum algorithm with random coin is deterministic, and the encoder knows the required queries for the simulation a priori. An encoding consists of the following components:

- Advice with size S.
- Values of f on $([K] \times [N] \setminus R_1) \times [L]$.
- Position of V in R_1: $\log\binom{|R_1|}{|V|}$.
- For all $(a,k) \in V$, the required query answers for the simulation.
- All other information of f on $R_1 \times [L]$, except the queried values and the answer of algorithm.

The decoding procedure can be done as in Theorem 3, by noting that the condition (B) can be checked by computing

$$\Pr[\mathsf{Find} : \mathcal{B}^{|g\rangle\setminus R_1\times[L], f(a,k,\cdot)}(\mathsf{st}_f, a, k)]$$

by using g instead of f, by Lemma 1 and the fact that $\mathcal{B}$ has the query magnitude 0 on $\{(a,k,\cdot)\}$. The overall encoding size is $S + KNL\log M - |V|\log M + \log\binom{|R_1|}{|V|}$ since the values not encoded are only the answers of simulation. By Lemma 4 and $\log(\binom{a}{b}) \leq b\log(ea/b)$, we obtain

$$O(S) \geq |V|\log\left(\frac{M|V|}{e|R_1|}\right),$$

or

$$O(S) \geq \frac{\varepsilon^3 KN}{T^2}\log\left(\varepsilon^2 M\right),$$

which implies

$$\varepsilon^3 = O\left(\frac{ST^2}{KN} + \frac{1}{M}\right).$$

Combining all, we obtain the desired results. □

4 Random Permutation with Quantum AI

In this section, we give a security bound for inverting random permutations with *quantum auxiliary input*.

4.1 Preparations

First, we prepare some lemmas that are needed for proving our results.

Quantum Compression Lemma Nayak [Nay99] generalized the seminal result of Holevo [Hol73] to relate the number of qubits that is needed to transmit n-bit classical information and the success probability of it.

Theorem 5. [Nay99, NS06, adapted] *Suppose that Alice holds an n-bit string x and wants to convey it to Bob via a (noiseless) quantum channel. If, for any x, the probability that Bob successfully recovers x is $p \in (0,1]$, then the number of qubits m transmitted by Alice is at least $n - \log 1/p$.*

Note that the above statement is very similar to the compression argument in the classical setting. Using this Theorem 5, we can obtain the following quantum compression lemma.

Lemma 8 (Quantum compression lemma). *Let M, R be a set. Let E be a procedure that takes $(x,r) \in M \times R$ and outputs a m-qubit quantum state and D a procedure that takes a quantum state along with string $r \in R$. If we have*

$$\Pr_r[D(E(x,r),r) = x] \geq p$$

for all $x \in M$, then it holds that $m \geq \log|M| - 2\log 1/p + 1$.

Proof. By the standard averaging argument, there exist an $r_0 \in R$ and set $M' \subset M$ with $|M'| \geq p|M|$ such that $\Pr[D(E(x,r_0),r_0) = x] \geq p$ for all $x \in M'$. We then apply Theorem 5 on $D'(\cdot) = D(\cdot, r_0)$ and $E'(\cdot) = E(\cdot, r_0)$ and any set $M'' \subset M'$ with power-of-two size and $|M''| \geq |M|/2$, we obtain the desired result as follows:

$$m \geq \log|M''| - \log 1/p \geq \log(p/2 \cdot |M|) - \log 1/p = \log|M| - 2\log 1/p + 1$$

□

Rewinding Quantum Advice Here, we describe a way to reuse a quantum advice for quantum algorithms when the outputs of the algorithms are fixed values with very high probability. We note that a similar idea has been used in several works [Aar05, AR19].

Specifically, Aaronson [Aar05] implicitly proved the following lemma by using the *gentle measurement lemma* [Win99], whose proof can be found in Appendix C.1

Lemma 9 (Implicit in [Aar05]). *Let ρ be any (mixed) quantum state, n be any positive integer, and for $i \in [n]$, let $\mathcal{A}_i$ be a unitary quantum algorithm (i.e., $\mathcal{A}_i$ is unitary except for the final measurement) such that $\Pr[\mathcal{A}_i(\rho) = x_i] > 1 - \frac{1}{9n^4}$ for some classical string x_i. Then there exists an algorithm $\mathcal{B}$ such that $\Pr[\mathcal{B}(\rho) = \{x_i\}_{i\in[n]}] > 2/3$.*

4.2 Bound for Inverting Random Permutations

Theorem 6. *Let $\mathcal{O} \in \mathsf{Func}([K] \times [N], [N])$ be a random permutation with salt (i.e., $\mathcal{O}(a,\cdot)$ is a random permutation). Suppose that $\mathcal{A}$ is an oracle-aided quantum algorithm that takes an S-bit quantum advice $|\mathsf{st}_\mathcal{O}\rangle$ (that may depend on $\mathcal{O}$) as input, makes at most T oracle queries, and satisfies*

$$\Pr_{\mathcal{A},\mathcal{O},a,x}\left[\mathcal{A}^{|\mathcal{O}\rangle}(|\mathsf{st}_\mathcal{O}\rangle, a, \mathcal{O}(a,x)) \to x\right] = \varepsilon,$$

Then it holds that $\varepsilon^3 = \widetilde{O}\left(\frac{ST^2}{KN} + \frac{T^2}{N}\right)$.

Remark 2. In the above, we assumed the advice $|\mathsf{st}_\mathcal{O}\rangle$ is a pure state. This does not lose generality since any S-qubit mixed state can be realized as half of a $2S$-qubit pure state by purification.

Proof of Theorem 6. By an averaging argument, there exists a set of functions $\mathcal{F}$ that is an $\varepsilon/2$-fraction of random oracles such that

$$\Pr_{\mathcal{A},a,x}[\mathcal{A}^{|f\rangle}(|\mathsf{st}_f\rangle, a, f(a,x)) \to x] \geq \varepsilon/2$$

for all $f \in \mathcal{F}$. Fix $f \in \mathcal{F}$. Again, by an averaging argument, there are at least $\varepsilon/4 \cdot KN$ elements (a,x) satisfying

$$\Pr_{\mathcal{A}}[\mathcal{A}^{|f\rangle}(|\mathsf{st}_f\rangle, a, f(a,x)) \to x] \geq \varepsilon/4.$$

We denote the set of such (a,x) by I and call it *semi-good*.

Now we consider an algorithm $\mathcal{B}$ that is an "amplified version" of $\mathcal{A}$ that satisfies

$$\Pr_{\mathcal{B}}[\mathcal{B}^{|f\rangle}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x)) \to x] \geq 3/4$$

for all $(a,x) \in I$. More precisely, $\mathcal{B}$ runs $\Theta(1/\varepsilon)$ copies of $\mathcal{A}$ in parallel except measurements, checks the correctness of outputs of $\mathcal{A}$ (before measurements) by querying them to f, and then outputs x if any of them is the correct answer x and $\perp$ otherwise. The number and depth of queries of $\mathcal{B}$ are $T' = \Theta(T/\varepsilon)$ and $D' = T+1$, respectively, and the quantum advice $|\widetilde{\mathsf{st}}_f\rangle$ is $\Theta(S/\varepsilon)$-qubit.

Then a random set $R \subset [K] \times [N]$ is chosen that will serve as a random public coin for encoding, so that $(a,x) \in R$ with probability $p = d/T'(T+2)$ (independently for each (a,x)) for some constant d ($d < 1/46080$ suffices). Here, we may assume that $p|I| \geq C$ for a sufficiently large constant C ($C \geq 16 \ln 10$ suffices) since otherwise we have $\varepsilon^2 KN/T^2 = O(1)$ in which case the statement of Theorem 6 trivially holds.[19]

We say that $(a,x) \in I$ is good if both

$$(A)\ (a,x) \in R, \qquad (B)\ \Pr_{\mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \backslash (R \backslash \{(a,x)\})}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x))] \leq \frac{1}{576(T+2)}$$

hold. A set of good elements is denoted by G.

Then the following claim can be proven similarly to Claim 2. The proof can be found in Appendix C.2.

Claim 4. $\Pr_R[|G| \geq \delta\varepsilon^2 KN/T^2] > 0.8$ *for some constant* δ*.*

We say that R is good if $|G| \geq \delta\varepsilon^2 KN/T^2$. We now fix a good R. For $y \in [N]$, we define a function $g_y : [K] \times [N] \to [N]$ by

$$g_y(a,z) = \begin{cases} f(a,z) & \text{if } (a,z) \notin R, \\ y & \text{otherwise.} \end{cases}$$

We note that g_y agrees with f on $R \backslash \{(a,x)\}$ where (a,x) is any preimage of y in f (i.e., $f(a,x) = y$). Here, we consider an algorithm $\mathcal{C}$ that works similarly to $\mathcal{B}$ except that it takes x as an additional input and returns 1 if $\mathcal{B}$'s output is x and 0 otherwise. By Lemma 2 and Remark 1, for any $(a,x) \in G$, we have

$$\begin{aligned} &\left|\Pr_{\mathcal{C}}[\mathcal{C}^{|g_{f(a,x)}\rangle}(|\widetilde{\mathsf{st}}_f\rangle, a, x, f(a,x)) \to 1] - \Pr_{\mathcal{C}}[\mathcal{C}^{|f\rangle}(|\widetilde{\mathsf{st}}_f\rangle, a, x, f(a,x)) \to 1]\right| \\ &\quad \leq 2\sqrt{(T+2)\Pr_{\mathcal{C}}[\mathsf{Find} : \mathcal{C}^{|f\rangle \backslash (R \backslash \{(a,x)\})}(|\widetilde{\mathsf{st}}_f\rangle, a, x, f(a,x))]} \end{aligned}$$

which is clearly equivalent to

$$\begin{aligned} &\left|\Pr_{\mathcal{B}}[\mathcal{B}^{|g_{f(a,x)}\rangle}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x)) \to x] - \Pr_{\mathcal{B}}[\mathcal{B}^{|f\rangle}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x)) \to x]\right| \\ &\quad \leq 2\sqrt{(T+2)\Pr_{\mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \backslash (R \backslash \{(a,x)\})}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x))]} \leq \frac{1}{12}. \end{aligned}$$

Thus we have

$$\Pr_{\mathcal{B}}[\mathcal{B}^{|g_{f(a,x)}\rangle}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x)) \to x] \geq \frac{3}{4} - \frac{1}{12} = \frac{2}{3}.$$

Note that the algorithm $\mathcal{B}$ outputs one particular answer x or $\perp$, so we can amplify the success probability by running $O(\log(KN))$ copies of $\mathcal{B}$ in parallel and taking an output of any execution of $\mathcal{B}$ that is not $\perp$ as its final output if any (before the measurement). We call this algorithm $\widetilde{\mathcal{B}}$, which satisfies

$$\Pr_{\widetilde{\mathcal{B}}}[\widetilde{\mathcal{B}}^{|g_{f(a,x)}\rangle}(|\overline{\mathsf{st}}_f\rangle, a, f(a,x)) \to x] \geq 1 - \frac{1}{9(KN)^4},$$

where $|\overline{\mathsf{st}}_f\rangle$ is $O(S\log(KN)/\varepsilon)$ qubits.

Now we are ready to encode the function f for good R. Let $R_a := R \cap (\{a\} \times [N])$ and $G_a = G \cap (\{a\} \times [N])$. The encoding of f includes the following information:

[19] Looking ahead, this is used in the proof of Claim 4.

- The advice string $|\mathsf{st}_f\rangle$: $O(S\log(KN)/\varepsilon)$ qubits.
- The set $f(R_a)$ for each $a \in [K]$: $\sum_a \log\binom{N}{|R_a|}$ bits.
- The values of f on $(\{a\} \times [N]) \setminus R_a$ for each $a \in [K]$: $\sum_a \log(N - |R_a|)!$ bits.
- The cardinality of G_a for each $a \in [K]$: $K \log N$ bits.
- The set $f(G_a)$ for each $a \in [K]$: $\sum_a \log\binom{|R_a|}{|G_a|}$ bits.
- The values of f on $R_a \setminus G_a$: $\sum_a \log(|R_a| - |G_a|)!$ bits.

The decoding procedure initializes an empty table to store the values of f and then fills the table as follows:

1. Recover $|\mathsf{st}_f\rangle$, G_a, and G.
2. Fill the values of f on inputs in $([K] \times [N]) \setminus R$. This can be done since the decoder knows R as a shared random string.
3. Fill the table of f for G by the following procedures. For each $(a, y) \in f(G_a)$, let $x \in [N]$ be the inversion of y at a, i.e., $y = f(a, x)$ (which is unknown to the decoder so far). Note that the function g_y can be evaluated by the decoder since it only needs values of f on $([K] \times [N]) \setminus R$ which is already recovered. As discussed above, we have

$$\Pr_{\widetilde{\mathcal{B}}}[\widetilde{\mathcal{B}}^{|g_{f(a,x)}\rangle}(|\mathsf{st}_f\rangle, a, f(a,x)) \to x] \geq 1 - \frac{1}{9(KN)^4}.$$

   Then the decoder uses the procedure in Lemma 9 to recover x for all $(a, y) \in f(G)$. Noting that $|f(G)| \leq KN$, by Lemma 9, the decoder succeeds in correctly recovering x for all $(a, y) \in f(G)$ with probability at least $2/3$. We note that the set G is also recovered at this point.
4. The decoder fills the values of f on inputs in $R \setminus G$ by using the partial truth table and the description of G that is recovered in the previous step.

The decoding procedure succeeds with a constant probability (over the choice of R and the randomness of measurements) since a constant fraction of R is good and the decoding succeeds with a constant probability for good R.

The overall encoding size except the size of advice string and the size of G_a is

$$\begin{aligned}
&\sum_{a\in[K]} \left(\log\binom{N}{|R_a|} + \log(N - |R_a|)! + \log\binom{|R_a|}{|G_a|} + \log(|R_a| - |G_a|)!\right) \\
&= \sum_{a\in[K]} \log\left(\frac{N!}{(N-|R_a|)!|R_a|!} \cdot (N - |R_a|)! \cdot \frac{|R_a|!}{(|R_a|-|G_a|)!|G_a|!} \cdot (|R_a| - |G_a|)!\right) \\
&= K\log N! - \sum_{a\in[K]} \log|G_a|! \\
&\leq K\log N! - \sum_{a\in[K]} |G_a|\log(|G_a|/e) \leq K\log N! - |G|\log\left(\frac{|G|}{eK}\right),
\end{aligned}$$

where we used the fact that $n! \geq (n/e)^n$ and $x\log x$ is convex in the last two inequalities. Then by Lemma 8, we obtain the inequality

$$O\left(\frac{S\log(KN)}{\varepsilon} + K\log N\right) \geq |G|\log\left(\frac{|G|}{eK}\right) + \Theta(1).$$

Then we have either $|G|/eK < 2$, which implies $\varepsilon^2 = O(T^2/N)$, or

$$O\left(\frac{S\log(KN)}{\varepsilon} + K\log N\right) \geq |G| \geq \delta\varepsilon^2 KN/T^2.$$

Combining them, we obtain $\varepsilon^3 = \widetilde{O}\left(\frac{ST^2}{KN} + \frac{T^2}{N}\right)$.

□

4.3 Implication in Complexity Theory

Here, we discuss an implication of the result of the previous section in complexity theory. We denote by $\mathsf{BQP/qpoly}$ the class of languages that can be decided in quantum polynomial time with a polynomial-size quantum advice.[20] Then the following theorem follows from Theorem 6.

Theorem 7. $\mathsf{NP} \cap \mathsf{coNP} \not\subseteq \mathsf{BQP/qpoly}$ *relative to a random permutation oracle with probability* 1.

Proof of Theorem 7. We follow the proof strategy of Bennett, Bernstein, Brassard, and Vazirani [BBBV97] who showed $\mathsf{NP} \cap \mathsf{coNP} \not\subseteq \mathsf{BQP}$ relative to a random permutation oracle with probability 1. Let $\mathcal{O} : \{0,1\}^* \to \{0,1\}^*$ be a function that defines a permutation over $\{0,1\}^n$ when restricted to $\{0,1\}^n$ for any n. (We denote the set of such functions by Perm.) We denote the restriction of $\mathcal{O}$ to $\{0,1\}^n$ by $\mathcal{O}_n$, and for any fixed choice of $\mathcal{O}^{-n} := \{\mathcal{O}_{n'}\}_{n' \in \mathbb{N} \setminus \{n\}}$, we denote the subset of Perm that consists of functions whose values on $\{0,1\}^{n'}$ match $\mathcal{O}_{n'}$ for all $n' \in \mathbb{N} \setminus \{n\}$ by $\mathsf{Perm}[\mathcal{O}^{-n}]$.

Relative to $\mathcal{O}$, we consider a language $\mathcal{L}^{\mathcal{O}} = \{(y,z) : \exists x \text{ s.t. } \mathcal{O}(x) = y \wedge x \leq z\}$ where $\leq$ means the inequality in lexicographical order, and we denote $\mathcal{L}^{\mathcal{O}} \cap (\{0,1\}^n)^2$ by $\mathcal{L}^{\mathcal{O}}_n$. Then it is clear that we have $\mathcal{L}^{\mathcal{O}} \in \mathsf{NP}^{\mathcal{O}} \cap \mathsf{coNP}^{\mathcal{O}}$ for any $\mathcal{O} \in \mathsf{Perm}$ since x can be used as a witness for both YES and NO instances. What is left is to prove $\mathcal{L}^{\mathcal{O}} \notin \mathsf{BQP}^{\mathcal{O}}/\mathsf{qpoly}$ with probability 1 over the choice of $\mathcal{O} \leftarrow \mathsf{Perm}$.

Let M be an oracle-aided quantum polynomial-time machine that takes $\mathrm{poly}(n)$-qubit quantum advice when its input length is $2n$ bits. Then we first show that for all sufficiently large n and any fixed $\mathcal{O}^{-n} = \{\mathcal{O}_{n'}\}_{n' \in \mathbb{N} \setminus \{n\}}$, we have

$$(*) \quad \Pr_{\mathcal{O} \leftarrow \mathsf{Perm}[\mathcal{O}^{-n}]}[\exists \left|\mathsf{st}\right\rangle \in \mathcal{H}^{\otimes \mathrm{poly}(n)},\ M^{|\mathcal{O}\rangle}(\left|\mathsf{st}\right\rangle, \cdot) \text{ decides } \mathcal{L}^{\mathcal{O}}_n] < 1/2$$

where $\mathcal{H}^{\otimes \mathrm{poly}(n)}$ denotes the set of all $\mathrm{poly}(n)$-qubit quantum states and we say that $M^{|\mathcal{O}\rangle}(\left|\mathsf{st}\right\rangle, \cdot)$ decides $\mathcal{L}^{\mathcal{O}}_n$ if $\Pr_M[M^{|\mathcal{O}\rangle}(\left|\mathsf{st}\right\rangle, (y,z)) = \mathcal{L}^{\mathcal{O}}(y,z)] > 2/3$ for all $(y,z) \in (\{0,1\}^n)^2$ where we define

$$\mathcal{L}^{\mathcal{O}}(y,z) = \begin{cases} 1 & \text{if } (y,z) \in \mathcal{L}^{\mathcal{O}}, \\ 0 & \text{otherwise.} \end{cases}$$

For the sake of contradiction, suppose that the above claim is false. Without loss of generality, we can assume that there exists a fixed choice of $\mathcal{O}^{-n}$ and $\mathrm{poly}(n)$-qubit quantum state $\left|\mathsf{st}_{\mathcal{O}}\right\rangle$ (that may depend on $\mathcal{O}$) such that $M^{\mathcal{O}}(\left|\mathsf{st}_{\mathcal{O}}\right\rangle, \cdot)$ decides $\mathcal{L}^{\mathcal{O}}_n$ with error probability $\exp(-\Omega(n))$ for at least $(1/2)$-fraction of $\mathcal{O} \in \mathsf{Perm}[\mathcal{O}^{-n}]$ by considering an $O(n)$ number of repetition (and giving $O(n)$ copies of $\left|\mathsf{st}_{\mathcal{O}}\right\rangle$ to M as input). Then by using a binary search, we can construct an algorithm $\mathcal{B}^{\mathcal{O}}$ that makes $\mathrm{poly}(n)$ queries such that $\Pr_{x,\mathcal{B}}[\mathcal{B}^{\mathcal{O}}(\left|\mathsf{st}_{\mathcal{O}}\right\rangle, \mathcal{O}(x)) = x] = 1 - \mathrm{poly}(n) \cdot \exp(-\Omega(n))$ for at least $(1/2)$-fraction of $\mathcal{O} \in \mathsf{Perm}[\mathcal{O}^{-n}]$. This clearly contradicts Theorem 6 with $T = \mathrm{poly}(n)$, $S = \mathrm{poly}(n)$, $K = 1$ and $N = 2^n$.

Since values of a random permutation on $\{0,1\}^n$ and those on $\{0,1\}^{n'}$ are independent for $n \neq n'$, by using the inequality $(*)$ for each $n \in \mathbb{N}$, we can conclude that for any polynomial-time quantum machine M, we have

$$\Pr_{\mathcal{O} \leftarrow \mathsf{Perm}}[\forall n \in \mathbb{N}, \exists \left|\mathsf{st}_n\right\rangle \in \mathcal{H}^{\otimes \mathrm{poly}(n)},\ M^{|\mathcal{O}\rangle}(\left|\mathsf{st}_n\right\rangle, \cdot) \text{ decides } \mathcal{L}^{\mathcal{O}}_n] = 0.$$

Finally, since the number of all machines is countable and the union of countable number of probability 0 events has probability 0, we have

$$\Pr_{\mathcal{O} \leftarrow \mathsf{Perm}}[\exists M, \forall n \in \mathbb{N}, \exists \left|\mathsf{st}_n\right\rangle \in \mathcal{H}^{\otimes \mathrm{poly}(n)},\ M^{|\mathcal{O}\rangle}(\left|\mathsf{st}_n\right\rangle, \cdot) \text{ decides } \mathcal{L}^{\mathcal{O}}_n] = 0.$$

This means that $\mathcal{L}^{\mathcal{O}} \notin \mathsf{BQP}^{\mathcal{O}}/\mathsf{qpoly}$ with probability 1. □

Acknowledgment

We thank anonymous reviewers of Asiacrypt 2019 and Andreas Hülsing for their helpful comments. Minki Hhan was partially supported by the Institute for Information & Communications Technology Promotion (IITP) Grant through the Korean Government (MSIT), (Development of lattice-based post-quantum public-key cryptographic schemes), under Grant 2017-0-00616 and by the Samsung Research Funding Center of Samsung Electronics under Project SRFC-TB1403-52.

[20] This class was originally introduced by Nishimura and Yamakami [NY04] with the name $\mathsf{BQP}/^*\mathsf{Qpoly}$, and renamed to $\mathsf{BQP/qpoly}$ by Aaronson [Aar05]. See these papers for the detailed definition.

References

Aar05. S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1(1):1–28, 2005.

AHU19. A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO 2019, Part II*, pages 269–295. 2019.

AR19. S. Aaronson and G. Rothblum. Gentle measurement of quantum states and differential privacy. In *STOC 2019*, pages 322–333. 2019.

BBBV97. C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

BDF+11. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, pages 41–69. 2011.

BHMT02. G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *Quantum Computation and Quantum Information*, 305:53–74, 2002.

BHT97. G. Brassard, P. Høyer, and A. Tapp. Quantum cryptanalysis of hash and claw-free functions. *SIGACT News*, 28(2):14–19, 1997.

BL13. D. J. Bernstein and T. Lange. Non-uniform cracks in the concrete: The power of free precomputation. In *ASIACRYPT 2013, Part II*, pages 321–340, 2013.

BR93. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS '93*, pages 62–73. 1993.

BR95. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *EUROCRYPT '94*. 1995.

BZ13. D. Boneh and M. Zhandry. Quantum-secure message authentication codes. In *EUROCRYPT 2013*, pages 592–608. 2013.

CBH+18. Jan Czajkowski, Leon Groot Bruinderink, A. Hülsing, Christian Schaffner, and D. Unruh. Post-quantum security of the sponge construction. In *PQCrypto 2018*, pages 185–204. 2018.

CDG18. S. Coretti, Y. Dodis, and S. Guo. Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In*CRYPTO 2018, Part I*, pages 693–721. 2018.

CDGS18. S. Coretti, Y. Dodis, S. Guo, and J. P. Steinberger. Random oracles and non-uniformity. In *EUROCRYPT 2018, Part I*, pages 227–258. 2018.

CK18. H. Corrigan-Gibbs and D. Kogan. The discrete-logarithm problem with preprocessing. In *EUROCRYPT 2018, Part II*, pages 415–447. 2018.

DFMS19. J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In *CRYPTO 2019, Part II*, pp 356-383. 2019.

DGK17. Y. Dodis, S. Guo, and Jonathan Katz. Fixing cracks in the concrete: Random oracles with auxiliary input, revisited. In *EUROCRYPT 2017, Part II*, pages 473–495. 2017.

DTT10. A. De, L. Trevisan, and M. Tulsiani. Time space tradeoffs for attacks against one-way functions and PRGs. In *CRYPTO 2010*, pages 649–665. 2010.

ES15. E. Eaton and F. Song. Making existential-unforgeable signatures strongly unforgeable in the quantum random-oracle model. In *TQC 2015*, pages 147–162, 2015. See also https://eprint.iacr.org/2015/878.

FN99. A. Fiat and M. Naor. Rigorous time/space trade-offs for inverting functions. *SIAM J. Comput.*, 29(3):790–803, 1999.

GGKT05. R. Gennaro, Y. Gertner, J. Katz, and L. Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005.

Gro96. L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC '96*, pages 212–219. 1996.

GT00. R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *FOCS 2000*, pages 305–313. 2000.

GYZ17. S. Garg, H. Yuen, and M. Zhandry. New security notions and feasibility results for authentication of quantum data. In *CRYPTO 2017, Part II*, pages 342–371. 2017.

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

Hol73. A. S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973.

HY18. A. Hosoyamada and T. Yamakawa. Finding collisions in a quantum world: quantum black-box separation of collision-resistance and one-wayness. Cryptology ePrint Archive, Report 2018/1066, 2018.

HRS16. A. Hülsing, J. Rijneveld, and F. Song. Mitigating multi-target attacks in hash-based signatures. In *PKC 2016, Part I*, pages 387–416. 2016.

JZC+18. H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *CRYPTO 2018, Part III*, pages 96–125. 2018.

KLS18. E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *EUROCRYPT 2018, Part III*, pages 552–586. 2018.

KYY18. S. Katsumata, S. Yamada, and T. Yamakawa. Tighter security proofs for GPV-IBE in the quantum random oracle model. In *ASIACRYPT 2018, Part II*, pages 253–282. 2018.
LZ19. Q. Liu and M. Zhandry. Revisiting post-quantum Fiat-Shamir. In *CRYPTO 2019, Part II*, pages 326-355. 2019.
NABT15. A. Nayebi, S. Aaronson, A. Belovs, and L. Trevisan. Quantum lower bound for inverting a permutation with advice. *Quantum Information & Computation*, 15(11-12):901–913, 2015.
Nay99. A. Nayak. Optimal lower bounds for quantum automata and random access codes. In *FOCS '99*, pages 369–376. 1999.
NC00. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Number 2. Cambridge University Press, 2000.
NS06. A. Nayak and J. Salzman. Limits on the ability of quantum states to convey classical messages. *Journal of the ACM*, 53(1):184–206, 2006.
NY04. H. Nishimura and T. Yamakami. Polynomial time quantum computation with advice. *Information Processing Letters*, 90(4):195–204, 2004.
SXY18. T. Saito, K. Xagawa, and T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *EUROCRYPT 2018, Part III*, pages 520–551. 2018.
TU16. E. E. Targhi and D. Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In *TCC 2016-B, Part II*, pages 192–216. 2016.
Unr07. D. Unruh. Random oracles and auxiliary input. In *CRYPTO 2007*, pages 205–223. 2007.
Unr15. D. Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT 2015, Part II*, pages 755–784. 2015.
Win99. A. Winter. Coding theorem and strong converse for quantum channels. *IEEE Trans. Information Theory*, 45(7):2481–2485, 1999.
Yao82. A. C.-C. Yao. Theory and applications of trapdoor functions. In *FOCS '82*, pages 80–91. 1982.
Yao90. A. C.-C. Yao. Coherent functions and program checkers. In *STOC '90*, pages 84–94. 1990.
Zha12a. M. Zhandry. How to construct quantum random functions. In *FOCS 2012*, pages 679–687. 2012.
Zha12b. M. Zhandry. Secure identity-based encryption in the quantum random oracle model. In *CRYPTO 2012*, pages 758–775. 2012.

A Amplitude Amplification

Here, we review a lemma called *amplitude amplification* [BHMT02] that is used in some parts of our proofs.

Lemma 10. *Let $f : X \to \{0,1\}$ be an arbitrary function. Let A be a unitary quantum algorithm (i.e., $\mathcal{A}$ is unitary except for the final measurement) that returns $x \in X$ such that $f(x) = 1$ with probability ε. Then there exists a quantum algorithm $\mathcal{B}$ that uses $\mathcal{A}$, $\mathcal{A}^{-1}$, and f as sub-routines $O(\varepsilon^{-1/2})$ times and returns $x \in X$ such that $f(x) = 1$ with probability $\Omega(1)$ where we abuse the notation to use $\mathcal{A}$ to mean the unitary corresponding to the algorithm $\mathcal{A}$ and $\mathcal{A}^{-1}$ to mean its inverse.*

B Omitted Proofs in Section 3

B.1 Proofs for OWFs in QROM-AI

Here, we give proofs of claims that are omitted in the proof of the security bound for OWFs in the QROM-AI (Theorem 1).

Proof of Claim 1

Proof of Claim 1. Since the expectation value of $|\{x : \mathcal{O}_a(x) = y\}|$ is N/M, by the Chernoff bound, we have

$$\Pr_{\mathcal{O}}[|\{x : \mathcal{O}_a(x) = y\}| \geq (1+\delta)N/M] \leq \exp(-\delta^2 N/(2+\delta)M) \leq \exp(-\delta N/2M)$$

for any (a, y) and for any $\delta \geq 2$. Now we choose $\delta = \max(2, 2M \log(4KM/\varepsilon)/N)$ so that

$$\exp(-\delta N/2M) \leq \exp(-\log(4KM/\varepsilon)) = \frac{\varepsilon}{4KM}.$$

Thus, the probability that there exists (a, y) such that the desired bound does not hold is at most $\varepsilon/4$ by union bound, over the choice of $\mathcal{O}$. In this case, the bound of $|\{x : \mathcal{O}_a(x) = y\}|$ is $O(N/M + \log(4KM/\varepsilon)) = \widetilde{O}(N/M + 1) = \widetilde{O}(N/\min(N, M))$.[21]

□

[21] Note that we assumed a lower bound of ε.

Proof of Claim 2

Proof of Claim 2. Let H be a subset of I that consists of all elements satisfying (A). Since for each $(a, x) \in I$ the probability that $(a, f(a, x)) \in R_1$ is $p = d/T(T+1)$, $\mathbb{E}[|H|] = p|I|$ holds. By the Chernoff bound

$$\Pr_{R_1}[|H| \geq p|I|/2] \geq 1 - \exp(-p|I|/8) \geq 1 - \exp(-C/16) \geq 0.9$$

holds for a sufficiently large C ($C \geq 16 \log 10$ suffices). Let

$$p_{\mathsf{Find}} := \Pr_{R_1, \mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus f^{-1}(R_{a,x})}(\mathsf{st}_f, a, f(a, x), L)].$$

By Lemma 3 we have

$$p_{\mathsf{Find}} \leq 4D \cdot \Pr[f^{-1}(R_{a,x}) \cap M \neq \emptyset : M \leftarrow \mathcal{C}(z)],$$

where D is the query depth of $\mathcal{B}$ and $\mathcal{C}$ is an algorithm that works as follows:

$\mathcal{C}(z)$ Chooses $i \leftarrow \{1, \cdots, D\}$; runs $\mathcal{B}^{\mathcal{O}_\emptyset^{SC}}(z)$ until (just before) the i-the query; then measures all query input registers in the computational basis and outputs the set M of measurement outcomes.

Then we have

$$4D \cdot \Pr[f^{-1}(R_1) \cap M \neq \emptyset : M \leftarrow \mathcal{C}(z)] \leq 4T \cdot p = \frac{4d}{T+1}$$

by the latter part of Lemma 3. Since we have $R_{(a,x)} \subseteq R_1$, we clearly have

$$\Pr[f^{-1}(R_{a,x}) \cap M \neq \emptyset : M \leftarrow \mathcal{C}(z)] \leq \Pr[f^{-1}(R_1) \cap M \neq \emptyset : M \leftarrow \mathcal{C}(z)],$$

which implies

$$p_{\mathsf{Find}} \leq \frac{4d}{T+1}.$$

Then the Markov inequality implies

$$\Pr_{R_1}\left[p_{\mathsf{Find}} \geq \frac{c^2}{16(T+1)}\right] \leq \frac{64d}{c^2}.$$

Let J be a subset of I that consists of all elements satisfying (A) but not (B). Note that two events (A) and (B) are independent since (A) only depends on if $f(a, x) \in R_1$ and (B) only depends on if the other points (i.e. that are in $[K] \times [M] \setminus \{f(a, x)\}$) are in R_1. Therefore, for any $(a, x) \in I$, we have

$$\Pr_{R_1}[(a, x) \in J] \leq 64d/c^2 \cdot p = 64d^2/c^2T(T+1).$$

Thus, by the Markov inequality,

$$\Pr_{R_1}\left[|J| \leq \frac{640d^2|I|}{c^2T(T+1)}\right] \geq 0.9$$

holds. Overall, with probability at least 0.8, it holds that

$$|G| = |H| - |J| \geq \frac{d|I|}{2T(T+1)} - \frac{640d^2|I|}{c^2T(T+1)} = \Omega\left(\frac{\varepsilon KN}{T^2}\right)$$

for appropriate choice of d and C as desired. □

B.2 Proof of Lemma 5

Here, we give a proof of Lemma 5, which is a variant of Theorem 1 and is used in the proof of security bounds for PRGs, pqPRFs, and pqMACs in the QROM-AI (Theorem 2, Theorem 3, and Theorem 4).

Proof of Lemma 5. Consider an algorithm $\mathcal{B}$ that works as follows:

$\mathcal{B}^{|f\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x))$**:** It runs $\mathcal{A}^{|f\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x))$. If the event Find occurs w.r.t. the running of $\mathcal{A}$, then this means that the query register of $\mathcal{A}$ collapsed to (a,x). In this case, $\mathcal{B}$ immediately halts and returns x. If Find never occurs, $\mathcal{B}$ returns $\perp$.

It is clear that $\mathcal{B}$ outputs x if and only if the event Find occurs. Thus, it suffices to bound

$$\Pr_{\mathcal{B},\mathcal{O},a,x}\left[\mathcal{B}^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x)) \to x\right].$$

To do so, we first consider an adversary $\mathcal{B}$ (which we call a *biased adversary*) $\mathcal{B}$ that satisfies

$$\Pr_{\mathcal{O},a,x}\left[\Pr_{\mathcal{B}}\left[\mathcal{B}^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x)) \to x\right] \geq 2/3\right] \geq \varepsilon,$$

and give a bound for such an adversary. More precisely, for a query depth D of the biased algorithm $\mathcal{B}$, we will show that

$$\varepsilon = O\left(\frac{STD}{KN} + \frac{TD\log N}{N}\right)$$

holds. For the time being, we assume that the above bound is correct, and show the statement of the lemma.

Suppose that there is an adversary $\mathcal{B}'$ satisfying

$$\Pr_{\mathcal{B}',\mathcal{O},a,x}\left[\mathcal{B}'^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x)) \to x\right] \geq \varepsilon'.$$

Then by applying averaging argument, there exists at least $(\varepsilon'/2)$-fraction of $(\mathcal{O}, a, x)$ that satisfies

$$\Pr_{\mathcal{B}'}\left[\mathcal{B}'^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x)) \to x\right] \geq \varepsilon'/2.$$

We then consider an algorithm $\widetilde{B}$ that works as follows:

$\widetilde{B}^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x))$ It runs $\mathcal{B}'^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x))$ r times independently. For each output z of $\mathcal{B}'$, $\widetilde{B}$ queries z to its semi-classical oracle $\mathcal{O}\setminus\{(a,x)\}$ and checks whether the event Find occurs. If Find ever occurs, then it can find the solution x, and outputs x. Otherwise it outputs $\perp$.

Note that the probability that $\widetilde{B}$ outputs x is at least $(1-\varepsilon'/2)^r \geq 2/3$ for an appropriately chosen $r = \Theta(1/\varepsilon')$. In particular, the number and depth of queries by the algorithm $\widetilde{B}$ are $\Theta(T/\varepsilon')$ and $T+1$, respectively, and we have

$$\Pr_{\mathcal{O},a,x}\left[\Pr_{\widetilde{B}}\left[\widetilde{B}^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}_{\mathcal{O}}, a, \mathcal{O}(a,x)) \to x\right] \geq 2/3\right] \geq \varepsilon'/2.$$

By the assumed bound for the biased adversary, we have $\varepsilon'/2 = O\left(\frac{STD}{\varepsilon' KN} + \frac{TD\log N}{\varepsilon' N}\right)$, or equivalently

$$\varepsilon'^2 = O\left(\frac{STD}{KN} + \frac{TD\log N}{N}\right) = O\left(\frac{ST^2}{KN} + \frac{T^2\log N}{N}\right)$$

as desired.

We then turn to prove the bound for the biased adversary. Suppose that $\varepsilon \geq C(STD/KN + TD\log N/N)$ for a sufficiently large constant C for the sake of contradiction, which particularly implies $CTD \leq \varepsilon KN$.[22] To apply the semi-classical oracle O2H, we consider another algorithm $\mathcal{C}$ that works as follows:

$\mathcal{C}^{|f\rangle}(\mathsf{st}, a, x, y)$: It runs $\mathcal{B}^{|f\rangle\setminus\{(a,x)\}}(\mathsf{st}, a, y)$. $\mathcal{C}$ outputs 1 if $\mathcal{B}$ outputs x, and outputs 0 otherwise.

Then it holds that

$$\Pr_{\mathcal{B}}\left[\mathcal{B}^{|\mathcal{O}\rangle\setminus\{(a,x)\}}(\mathsf{st}, a, \mathcal{O}(a,x)) \to x\right] = \Pr_{\mathcal{C}}\left[\mathcal{C}^{|\mathcal{O}\rangle}(\mathsf{st}, a, x, \mathcal{O}(a,x)) \to 1\right].$$

[22] This will be used in the proof of Claim 5.

By applying the averaging argument, there exists a set $\mathcal{F} \subset \mathsf{Func}([K] \times [N], [M])$ with size at least $\varepsilon/2 \cdot M^{KN}$ that satisfies

$$\Pr_{a,x}\left[\Pr_{\mathcal{C}}\left[\mathcal{C}^{|f\rangle}(\mathsf{st}_f, a, x, f(a,x)) \to 1\right] \geq 2/3\right] \geq \varepsilon/2$$

for all $f \in \mathcal{F}$. Fix $f \in \mathcal{F}$. Then there exists a set $I \subset [K] \times [N]$ with size $|I| \geq \varepsilon/2 \cdot KN$ such that for all $(a,x) \in I$

$$\Pr_{\mathcal{C}}\left[\mathcal{C}^{|f\rangle}(\mathsf{st}_f, a, x, f(a,x)) \to 1\right] \geq 2/3$$

holds. Now we choose a random set $R \subset [K] \times [N]$, which will serve as a random public coin for encoding, such that $(a,x) \in R$ with probability $p = b/T(D+1)$ for some constant b to be specified later. We say that $(a,x) \in I$ is *good* (for R) if the following two conditions simultaneously hold for some constant c ($80b < c < 1/36$ suffices).

$$(A)\ (a,x) \in R \qquad\qquad (B)\ \Pr_{\mathcal{C}}[\mathsf{Find} : \mathcal{C}^{|f\rangle\setminus(R\setminus\{(a,x)\})}(\mathsf{st}_f, a, x, f(a,x))] \leq \frac{c}{D+1}$$

We denote the set of all good elements by $G = G(R)$.

Claim 5. $\Pr_R[|G(R)| \geq \delta\varepsilon KN/TD] \geq 0.8$ *for some constant* $\delta > 0$.

The proof can be found in the end of this section. For fixed R and $(a,x) \in G$, let $y = f(a,x)$ and we consider another function g_y that is defined by

$$g_y(z) = \begin{cases} f(z), & \text{if } z \in ([K] \times [N]) \setminus R, \\ y, & \text{if } z \in R. \end{cases}$$

Then by Lemma 2 and the condition (B), it holds that

$$\begin{aligned}&\left|\Pr_{\mathcal{C}}\left[\mathcal{C}^{|g_y\rangle\setminus R}(\mathsf{st}_f, a, x, f(a,x)) \to 1\right] - \Pr_{\mathcal{C}}\left[\mathcal{C}^{|f\rangle}(\mathsf{st}_f, a, x, f(a,x)) \to 1\right]\right| \\ &\leq \sqrt{(D+1)\Pr_{\mathcal{C}}[\mathsf{Find} : \mathcal{C}^{|f\rangle\setminus(R\setminus\{(a,x)\})}(\mathsf{st}_f, a, x, f(a,x))]} \leq \sqrt{c}.\end{aligned}$$

This gives

$$\Pr_{\mathcal{C}}\left[\mathcal{C}^{|g_y\rangle\setminus R}(\mathsf{st}_f, a, x, f(a,x)) \to 1\right] > 1/2$$

for any $(a,x) \in G$, which implies

$$\Pr_{\mathcal{B}}\left[\mathcal{B}^{|g_y\rangle\setminus R}(\mathsf{st}_f, a, f(a,x)) \to x\right] > 1/2,$$

since the twice-punctured semi-classical oracle $(|g_y\rangle \setminus R) \setminus \{(a,x)\}$ is essentially the same as $|g_y\rangle \setminus R$ since $(a,x) \in R$.

Let $V_a := \{(a, f(a,x)) : (a,x) \in G \cap \{a\} \times [N]\}$ and $V = \cup_{a\in[K]} V_a$. Note that $|V| = |G|$ since there is no good (a,x) and (a,x') such that $f(a,x) = f(a,x')$ due to the definition of I. Also note that $|V_a| \leq N$.

Now we are ready to encode the function f. We encode the function f only if $|V| = \Omega(\varepsilon KN/T^2)$, which holds with probability at least 0.8 over the choice of R, and declare a failure otherwise. The encoding of f includes the following information:

- The advice string st_f: S bits.
- The description of V_a with its size for each $a \in [K]$: $\log N + \log\binom{M}{|V_a|}$ bits.
- The values of $f(a,x)$ for $(a,x) \in ([K] \times [N]) \setminus R$ in lexicographical order of (a,x): $(KN - |R|)\log M$ bits.
- The values of $f(a,x)$ for $(a,x) \in R \setminus G$: $(|R| - |G|)\log M$ bits.

The decoding procedure initializes an empty table to store the values of f and then fills the table as follows:

1. Recover st_f, V_a and V.
2. Fill the values of f on $([K] \times [N]) \setminus R$. This can be done since the decoder knows $([K] \times [N]) \setminus R$ and $f(([K] \times [N]) \setminus R)$.

3. Fill the table of f for G by the following procedure. For each $(a, y) \in V_a$, let $x \in [N]$ be the inversion of y at a, i.e. $y = f(a, x)$ (which is unknown to the decoder so far). Note that the function g_y that is defined by

$$g_y(z) = \begin{cases} f(z), & \text{if } z \in ([K] \times [N]) \setminus R, \\ y, & \text{if } z \in R. \end{cases}$$

can be evaluated by the decoder. The output of

$$\mathcal{B}^{|g_y\rangle \setminus R}(\mathsf{st}_f, a, y)$$

is x with probability larger than $1/2$. Thus, the decoder can determine x by deterministically simulating this algorithm, i.e., by computing the output distribution. Then the decoder fills the table as $f(a, x) := y$.
4. The decoder fills the values of f on $R \setminus G$ using the partial truth table included in the encoding.

Note that the decoding procedure always succeeds whenever the encoding procedure does not declare a failure. Since $|G| = |V| = \sum_{a \in [K]} |V_a|$, the length of the encoding is

$$\begin{aligned} & S + K \log N + \sum_{a \in [K]} \log \binom{M}{|V_a|} + (KN - |G|) \log M \\ & \leq S + K \log N + KN \log M - \sum_{a} |V_a| \log \left(\frac{|V_a|}{e} \right) \\ & \leq S + K \log N + KN \log M - |V| \log \left(\frac{|V|}{eK} \right), \end{aligned}$$

where the first inequality is proven using $\binom{A}{B} \leq \left(\frac{eA}{B}\right)^B$ and the last inequality is due to the concavity of the function $f(y) = -\log(y/e)$. Then by Lemma 4, we obtain

$$S + K \log N + KN \log M - |V| \log \left(\frac{|V|}{eK} \right) \geq \log 0.4\varepsilon + KN \log M$$

or, since $|V| = \Omega(\varepsilon KN/TD)$,

$$S + K \log N \geq E\varepsilon KN/TD \log \left(\frac{|V|}{eK} \right) + \widetilde{O}(1)$$

for some constant E. This inequality implies either $|V|/2eK \leq 1$ or $\varepsilon = O\left(\frac{STD}{KN} + \frac{TD \log N}{N}\right)$. Combining them, we obtain

$$\varepsilon = O\left(\frac{STD}{KN} + \frac{TD \log N}{N}\right),$$

which concludes the proof. $\square$

Proof of Claim 5

Proof of Claim 5. The proof is essentially the same as the proof of Claim 2. Let H be the intersection of I and R. Then, since we have $\mathbb{E}[|H|] = p|I|$, by the Chernoff bound,

$$\Pr_R [|H| \geq p|I|/2] \geq 1 - \exp(-p|I|/8) \geq 1 - \exp(-C/16) \geq 0.9$$

holds for a sufficiently large C. We can show

$$\Pr_{R, \mathcal{C}}[\mathsf{Find} : \mathcal{C}^{|f\rangle \setminus (R \setminus \{(a,x)\})}(\mathsf{st}_f, a, x, f(a, x))] \leq 4b/(D+1)$$

by Lemma 3 and a similar argument to the proof of Claim 2. Then Markov's inequality implies

$$\Pr_R \left[\Pr_{\mathcal{C}}[\mathsf{Find} : \mathcal{C}^{|f\rangle \setminus (R \setminus \{(a,x)\})}(\mathsf{st}_f, a, x, f(a, x))] \geq \frac{c}{D+1} \right] \leq \frac{4b}{c}.$$

Let J be a subset of I that consists of all elements satisfying (A) but not (B). Since two events are independent and (a, x) not satisfying (B) implies

$$\Pr_{\mathcal{C}}[\mathsf{Find} : \mathcal{C}^{|f\rangle \setminus (R \setminus \{(a,x)\})}(\mathsf{st}_f, a, f(a,x))] \geq \frac{c}{D+1},$$

the probability that $(a, x) \in J$ is at most $4b/c \cdot p = 4b^2/cT(D+1)$ for any $(a, x) \in I$. Thus, by the Markov inequality

$$\Pr_R \left[|J| \leq \frac{40b^2|I|}{cT(D+1)} \right] \geq 0.9$$

holds. Overall, with probability at least 0.8, it holds that

$$|G| = |H| - |J| \geq \frac{b|I|}{2T(D+1)} - \frac{40b^2|I|}{cT(D+1)} = \Omega\left(\frac{\varepsilon KN}{TD}\right)$$

for the appropriate choice of b, c and C as desired. $\square$

B.3 Proofs for Yao's Box Problem

Here, we give a proof of the security bound for Yao's box problem (Lemma 6) that is used in the proof of the security bound for PRGs in the QROM-AI (Theorem 2).

Proof of Lemma 6. Fix $f \in \mathcal{F}$. Choose a random set $R_1 \subset [N]$ where $x \in [N]$ is included in R_1 with probability $\delta\varepsilon^3/T(T+1)$ for some fixed constant $\delta \leq 1/65536$. Now we say that an element $x \in [N]$ is good if the following two conditions hold:

(A) $x \in R_1$,

(B) $\Pr_{\mathcal{A}}[\mathsf{Find} : \mathcal{A}^{|f\rangle \setminus (R_1 \setminus \{x\})}(\mathsf{st}_{\mathcal{O}}, x)] = \Pr_{\mathcal{A}}[\mathsf{Find} : \mathcal{A}^{|f\rangle \setminus R_1}(\mathsf{st}_{\mathcal{O}}, x)] \leq \frac{16\delta\varepsilon^2}{T+1}$.

The equality in (B) holds because $\mathcal{A}$ has the query magnitude 0 on x.

Claim 6. *$\Pr_{R_1}[x$ is good$] \geq \delta\varepsilon^3/T(T+1) \cdot (1 - \varepsilon/4)$ for all $x \in [N]$.*

We defer the proof to the end of this section. We denote a set of good elements by $G = G(R_1)$. We say that R_1 is good if

(C) $|G(R_1)| / \mathbb{E}_{R_1}[|G|] \geq \varepsilon/8$ and (D) $\Pr_{\mathcal{A}, x \in G}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)] \geq \frac{1}{2} + \varepsilon/16$

hold simultaneously.

Claim 7. *$\Omega(\varepsilon^4/T^2)$-fraction of R_1 is good.*

We defer the proof of this claim since it is quite technical and relies a slightly complex analyses of probabilities. Fix a good R_1. Let g be a function defined by

$$g(z) = \begin{cases} f(z) & \text{for } z \in [N] \setminus R_1 \\ 0 & \text{otherwise} \end{cases}$$

which agrees with f on $[N] \setminus R_1$. Then, by the semi-classical O2H lemma (Lemma 2),

$$\left| \Pr_{\mathcal{A}, x \in G}[\mathcal{A}^{|f\rangle}(\mathsf{st}_f, x) \to f(x)] - \Pr_{\mathcal{A}, x \in G}[\mathcal{A}^{|g\rangle}(\mathsf{st}_f, x) \to f(x)] \right|$$

is bounded by

$$2\sqrt{T \cdot \Pr_{\mathcal{A}, x \in G}[\mathsf{Find} : \mathcal{A}^{|f\rangle \setminus R_1}(\mathsf{st}_f, x)]} \leq 2\varepsilon\sqrt{16\delta} \leq \frac{\varepsilon}{32}$$

because x is good and thus satisfies the condition (B). Overall, it holds that

$$\Pr_{\mathcal{A}, x \in G}[\mathcal{A}^{|g\rangle}(\mathsf{st}_f, x) \to f(x)] \geq \frac{1}{2} + \frac{\varepsilon}{32}.$$

Now we consider the simulation of this algorithm with a random coin $R_2 = (r_x)_{x\in[N]}$ as the measurement randomness. Then, for at least $(\varepsilon/32)$-fraction of R_2, it holds that

$$\Pr_{x\in G}[\mathsf{Sim}_{R_2}(\mathcal{A}^{|g\rangle}(\mathsf{st}_f, x)) \to f(x)] \geq \frac{1}{2} + \frac{\varepsilon}{64}.$$

We say that (R_1, R_2) is good if R_1 is good and (R_1, R_2) satisfies the above inequality. Note that the simulation is a deterministic algorithm for a fixed R_2. Let G_0 be the set of good elements (a, x) such that the simulation with coin R_2 outputs the correct answer, and $G_1 = G \setminus G_0$. For good $R = (R_1, R_2)$, which is at least $\Omega(\varepsilon^4/T^2)\cdot(\varepsilon/32) = \Omega(\varepsilon^5/T^2)$-fraction, it holds that

$$(C') \quad |G(R_1)| \geq \Omega(\varepsilon^4 N/T^2) \qquad \text{and} \qquad (D') \quad |G_0| - |G_1| = \Omega(\varepsilon|G|).$$

Now we are ready to describe our encoding and decoding procedures, which only work well for good R. Our encoding consists of the following components:

- Advice string st_f with size S,
- Values of f on $[N] \setminus R_1$ with size $N - |R_1|$, in lexicographic order,
- Values of f on $R_1 \setminus G$ with size $|R_1| - |G|$, in lexicographic order, and
- The position of G_0 in G, taking $|G| \cdot \mathbb{H}(1/2 + \Omega(\varepsilon))$ bits (by the condition (D')), which is at most $|G|(1 - O(\varepsilon^2))$ bits.[23]

The overall encoding size is at most $S + N - \Omega(\varepsilon^2|G|) = S + N - \Omega(\varepsilon^6 N/T^2)$. Then, our decoder reconstructs the table for f by the following procedure.

1. Recover advice string st_f.
2. Fill the values of f on $[N] \setminus R_1$ by using the partial truth table of f on $[N] \setminus R_1$ included in the encoding and the public randomness R_1.
3. Recover the set G and fill the values of f on G as follows. Note that the decoder can evaluate the oracle g defined by

$$g(z) = \begin{cases} f(z) & \text{for } z \in [N] \setminus R_1 \\ 0 & \text{otherwise} \end{cases}$$

by using information that is already recovered. For each $x \in R_1$, the decoder checks if

$$(B) \quad \Pr_{\mathcal{A}}[\mathsf{Find} : \mathcal{A}^{|g\rangle\setminus R_1}(\mathsf{st}_{\mathcal{O}}, x)] \overset{\text{Lemma 1}}{=} \Pr_{\mathcal{A}}[\mathsf{Find} : \mathcal{A}^{|f\rangle\setminus R_1}(\mathsf{st}_{\mathcal{O}}, x)] \leq \frac{16\delta\varepsilon^2}{T+1}$$

holds for each $x \in R_1$. If this condition holds, then the decoder decides $x \in G$. Note that the condition (B) can be checked by using g instead of f since they agree on $[N] \setminus R_1$. Then the decoder recovers sets G_0 and G_1 by using the position of G_0 included in the encoding and simulates

$$\mathsf{Sim}_{R_2}(\mathcal{A}^{|g\rangle}(\mathsf{st}_f, x))$$

for every $x \in G$. Then, for $x \in G_0$, the decoder fills $f(x)$ as the result of the simulation, and for $x \in G_1$, the decoder fills $f(x)$ by flipping the result.
4. Fill the values of f on $R_1 \setminus G$ in lexicographic order.

The success probability of recovering f is $\Omega(\varepsilon^5/T^2)$, which is the probability that R is good. This concludes the first part of the proof of the lemma. For the last statement, by applying Lemma 4, we obtain

$$S + N - \Omega(\varepsilon^6 N/T^2) \geq N + \log(CT^2/\varepsilon^5)$$

for some constant C. If $\varepsilon^6 > \max(1, C)\cdot ST^2/N$ holds, then we have $CT^2/\varepsilon^5 < N$ and $\varepsilon^6 N/T^2 > S$, which contradicts the above inequality. Thus, $\varepsilon^6 = O(ST^2/N)$ holds as we desired. ☐

[23] Here $\mathbb{H}$ stands for the information entropy, $\mathbb{H}(p) = -p\log p - (1-p)\log(1-p)$.

Proof of Claim 6

Proof of Claim 6. Fix $x \in [N]$. Note that two events (A) and (B) are independent since (A) only depends on whether $x \in R_1$ and (B) only depends on if the other points are in R_1. The probability that (A) holds is exactly $\delta\varepsilon^3/T(T+1)$. For (B), by Lemma 3,

$$\mathop{\mathbb{E}}_{R_1}\left[\Pr_{\mathcal{A}}[\mathsf{Find} : \mathcal{A}^{f\setminus R_1}(\mathsf{st}_{\mathcal{O}}, x)]\right] \leq \frac{4\delta\varepsilon^3}{T+1}$$

holds. By the Markov's inequality, we have

$$\Pr_{R_1}\left[\Pr_{\mathcal{A}}[\mathsf{Find} : \mathcal{A}^{f\setminus R_1}(\mathsf{st}_{\mathcal{O}}, x)] \geq \frac{16\delta\varepsilon^2}{T+1}\right] \leq \frac{4\delta\varepsilon}{16\delta} = \frac{\varepsilon}{4}.$$

Overall, the probability that x is good is at least $\delta\varepsilon^3/T(T+1)\cdot(1-\varepsilon/4)$ as desired. $\square$

Proof of Claim 7

Proof of Claim 7. Let $p := \delta\varepsilon^3/T(T+1)$ and recall that $\Pr_{R_1}[x \in G] \geq p(1-\varepsilon/4)$ holds for all x. Thus, we have $\mathbb{E}[|G|] \geq pN(1-\varepsilon/4)$ and $\mathbb{E}[|G|] \leq \mathbb{E}[|R_1|] = pN$. Let $\mathcal{R}$ be the probabilistic space of R_1 and μ be the uniform probabilistic measure on $\mathcal{R}$. Then we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{R_1}[|G| \Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)]] \\
&= \int_{\mathcal{R}} |G(R_1)| \cdot \Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)] d\mu \\
&= \int_{\mathcal{R}} \sum_{x\in G} \Pr_{\mathcal{A}}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)] d\mu \\
&= \int_{\mathcal{R}} \sum_{x\in[N]} \delta_G(x) \cdot \Pr_{\mathcal{A}}[\mathcal{A}^f(st, x) = f(x)] d\mu \\
&= \sum_{x\in[N]} \int_{\mathcal{R}} \delta_G(x) \cdot \Pr_{\mathcal{A}}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)] d\mu \\
&= \sum_{x\in[N]} \Pr_{R_1}[x \in G] \cdot \Pr_{\mathcal{A}}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)] \\
&\geq \sum_{x\in[N]} p(1-\varepsilon/4) \cdot \Pr_{\mathcal{A}}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)] \\
&= pN(1-\varepsilon/4) \Pr_{\mathcal{A},x}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)]
\end{aligned}$$

where δ_G denotes the indicator function for G. From this, we have

$$\mathop{\mathbb{E}}_{R_1}\left[\frac{|G(R_1)|}{pN(1-\varepsilon/4)} \cdot \Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)]\right] \geq \frac{1}{2} + \varepsilon/2,$$

which implies

$$\mathop{\mathbb{E}}_{R_1}\left[\frac{|G(R_1)|}{\mathbb{E}[|G|]} \cdot \Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)]\right] \geq \left(\frac{1}{2} + \varepsilon/2\right) \cdot (1-\varepsilon/4) \geq \frac{1}{2} + \varepsilon/4$$

since $pN \geq \mathbb{E}[|G|]$.

Now we consider another probabilistic measure $\nu(R) = \frac{|G(R)|}{\mathbb{E}[|G|]} \cdot \mu(R)$. If we consider ν instead of μ, we obtain

$$\mathop{\mathbb{E}}_{\nu}[\Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f, x) = f(x)]] \geq \frac{1}{2} + \varepsilon/4.$$

We say R_1 is very bad if it does not satisfy (C), i.e. $|G(R_1)| \leq \varepsilon/8 \cdot \mathbb{E}[|G|]$, and denote the set of all very bad elements by B. Then it holds that

$$\nu(B) = \int_B 1 d\nu = \int_B |G(R_1)|/\mathbb{E}[|G|] d\mu \leq \int_B \varepsilon/8 \cdot d\mu \leq \int_{\mathcal{R}} \varepsilon/8 \cdot d\mu = \varepsilon/8$$

which implies

$$\int_{\neg B} \Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f,x)=f(x)]d\nu \geq \frac{1}{2}+\varepsilon/8.$$

Note that this implies

$$\mathop{\mathbb{E}}_{\nu:R_1\in\neg B}[\Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f,x)=f(x)]] \geq \frac{1}{2}+\varepsilon/8$$

since

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\nu:R_1\in\neg B}[\Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f,x)=f(x)]]\\
&=\int_{\neg B} \Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f,x)=f(x)]d\nu/\nu(\neg B)\\
&\geq \int_{\neg B} \Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f,x)=f(x)]d\nu.
\end{aligned}$$

By the standard averaging argument, at least $(\varepsilon/8)$-fraction of not very bad R_1 (with respect to ν), denoted by S, satisfies

$$\Pr_{\mathcal{A},x\in G}[\mathcal{A}^f(\mathsf{st}_f,x)=f(x)] \geq \frac{1}{2}+\varepsilon/16.$$

Note that S is a set of good elements and we have

$$\nu(S)\geq \varepsilon/8\cdot\nu(\neg B)\geq \varepsilon/8-\varepsilon^2/64\geq\varepsilon/16.$$

Since $|G|\leq N$, we have $|G(R_1)|/\mathbb{E}[|G|]=O(T^2/\varepsilon^3)$. Thus, it holds that

$$\begin{aligned}
\mu(S)&=\int_S \mathbb{E}[|G|]/|G(R_1)|d\nu\\
&=\int_S \Omega(\varepsilon^3/T^2)d\nu=\Omega(\varepsilon^3/T^2)\nu(S)=\Omega(\varepsilon^4/T^2).
\end{aligned}$$

Thus, at least $\Omega(\varepsilon^4/T^2)$-fraction of R_1 satisfies (C) and (D) simultaneously. □

B.4 Proofs for Function Variant of Yao's Box Problem

Proof of Lemma 7 There exists at least $(\varepsilon/2)$-fraction of $\mathcal{O}$, denoted by $\mathcal{F}$, that satisfies

$$\Pr_{\mathcal{A},a,k}[\mathcal{A}^{|\mathcal{O}\rangle,\mathcal{O}(a,k,\cdot)}(\mathsf{st}_{\mathcal{O}},a,k)\to(m,t)\wedge t=\mathcal{O}(a,k,m)]\geq\frac{1}{2}+\frac{\varepsilon}{2}.$$

We will construct the encoding and decoding algorithms for $\mathcal{F}$. Fix $f\in\mathcal{F}$. We define another algorithm $\mathcal{B}$ that utilizes $\mathcal{A}$ as follows:

$\mathcal{B}^{|f\rangle,g}(\mathsf{st},a,k)$ $\mathcal{B}$ runs $\mathcal{A}^{|f\rangle,g}(\mathsf{st},a,k)$. For the output $z=(m,t)$ of $\mathcal{A}$, $\mathcal{B}$ checks if $t=g(m)$ by querying m to its second oracle. If so $\mathcal{B}$ outputs 1, and 0 otherwise.

From the definition, the probability that $\mathcal{B}$ outputs 1 is equal to the probability $\mathcal{A}$ outputs a correct answer. We choose a random set $R_1\subset[K]\times[N]$ where each $(a,k)\in[K]\times[N]$ is included in R_1 with probability $p=\delta\varepsilon^3/T(T+1)$ for some fixed constant $\delta\leq 1/65536$. We also choose a random coin R_2 for simulating measurements. We say that an element $(a,k)\in[K]\times[N]$ is good if

$$(A)\quad (a,k)\in R_1,\qquad\qquad (B)\quad \Pr_{\mathcal{B}}[\mathsf{Find}:\mathcal{B}^{|f\rangle\setminus R_1,f(a,k,\cdot)}(\mathsf{st}_f,a,k)]\leq\frac{16\delta\varepsilon^2}{T+1}$$

holds for some constant c.

Claim 8. $\Pr_{R_1}[(a,k)$ *is good*$]\geq\delta\varepsilon^3/T^2\cdot(1-\varepsilon/4)$ *holds for all* $(a,k)\in[K]\times[N]$.

We defer the proof to the end of this section. The set of all good elements is denoted by $G=G(R_1)$. We say R_1 is good if

$$(C)\quad \frac{|G(R_1)|}{\mathbb{E}[|G|]}\geq\frac{\varepsilon}{8},\qquad\qquad (D)\quad \Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|f\rangle,f(a,k,\cdot)}(\mathsf{st}_f,a,k)\to 1]\geq\frac{1}{2}+\frac{\varepsilon}{16}$$

hold simultaneously.

Claim 9. *$\Omega(\varepsilon^4/T^2)$-fraction of R_1 is good.*

We defer the proof of this claim. Fix a good R_1. Define $g : [K] \times [N] \times [L] \to [M] \cup \{0\}$ by

$$g(z, m) = \begin{cases} f(z, m) & \text{for } z \in [K] \times [N] \setminus R_1 \\ 0 & \text{otherwise.} \end{cases}$$

Then, by Lemma 2,

$$\left| \Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|f\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1] - \Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|g\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1] \right|$$

is bounded by

$$2\sqrt{(T+1) \cdot \Pr_{\mathcal{B},x\in G}[\mathsf{Find} : \mathcal{B}^{|f\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k)]} \le 2\varepsilon\sqrt{16\delta} \le \frac{\varepsilon}{32}.$$

because (a, k) is good and thus satisfies (B). This implies

$$\Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|g\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1] \ge \frac{1}{2} + \frac{\varepsilon}{32}.$$

Note that this is equivalent to

$$\Pr_{\mathcal{A},(a,k)\in G}[\mathcal{A}^{|g\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to (m, t) \wedge t = f(a, k, m)] \ge \frac{1}{2} + \frac{\varepsilon}{32}.$$

Now we consider the simulation of the algorithm $\mathcal{A}$ with a random coin R_2 as the measurement randomness. Then, for at least $(\varepsilon/32)$-fraction of R_2, it holds that

$$\Pr_{(a,k)\in G}[\mathsf{Sim}_{R_2}(\mathcal{A}^{|g\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k)) \to (m, t) \wedge t = f(a, k, m)] \ge \frac{1}{2} + \frac{\varepsilon}{64}.$$

We say (R_1, R_2) is good if R_1 is good and (R_1, R_2) satisfies the above inequality. Let G_0 be a set of good elements (a, k) such that the simulation with the coin R_2 outputs the correct answer, and $G_1 = G \setminus G_0$. For good $R = (R_1, R_2)$, which is at least $\Omega(\varepsilon^5/T^2)$-fraction, it holds that

$$(C') \quad |G(R_1, R_2)| = \Omega(\varepsilon^3 N/T^2) \qquad \text{and} \qquad (D') \quad |G_0| - |G_1| = \Omega(\varepsilon|G|).$$

Now we are ready to describe our encoding and decoding procedures, which only work well for good (R_1, R_2). Note that a quantum algorithm with given coins can be simulated *deterministically*, so the encoder can simulate the algorithm $\mathcal{A}$ with public coins a priori. This implies that the encoder can determine the required queries to the second oracle in the decoder's simulation. Our encoding consists of the following components.

- Advice string st_f with size S,
- Values of f on $([K] \times [N] \setminus R_1) \times [L]$ with size $(KN - |R_1|)L$, in lexicographic order,
- For all $(a, k) \in G$, the required queries answers for the simulation

$$\mathsf{Sim}_{R_2}(\mathcal{A}^{|g\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k)),$$

- All other information of function f on $R_1 \times [L]$, except the queried values and the answer of simulations, and
- The position of G_0 in G, taking $|G| \cdot \mathbb{H}(1/2 + \Omega(\varepsilon))$ bits (by the condition (D')), which is at most $|G|(1 - O(\varepsilon^2))$.

The decoder recovers the table for f as in Lemma 6, except that it recovers some function values on $G \times [L]$ as the answer of simulations; in Lemma 6, the decoder recovers all function values on G. Note that the decoder can determine the set G since the condition (B) can be checked by the function g instead of f, which can be computed by the decoder's partial function table $f([K] \times [N] \setminus R_1)$.

Since the encoding includes all function values except the answers of simulation, which has the size $|G|$, the overall encoding size is at most $S + KNL - \Omega(\varepsilon^2|G|) = S + KNL - \Omega(\varepsilon^6 KN/T^2)$.

The overall success probability of recovering f is $\Omega(\varepsilon^5/T^2)$. By Lemma 4, we obtain the inequality

$$S + KNL - \Omega(\varepsilon^5 KN/T^2) \ge KNL + \log(T^2/\varepsilon^5),$$

which implies $\varepsilon^6 = \Omega(ST^2/KN)$.

Proof of Claim 8

Proof of Claim 8. Fix $(a,k) \in [K] \times [N]$. Note that two events (A) and (B) are independent since (A) only depends on if $(a,k) \in R_1$ and (B) only depends on if other points are in R_1. The probability that (A) holds is exactly $\delta\varepsilon^3/T(T+1)$. For (B), by Lemma 3,

$$\mathop{\mathbb{E}}_{R_1}\left[\Pr_{\mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle\setminus R_1, f(a,k,\cdot)}(\mathsf{st}_f, a, k)]\right] \le \frac{4\delta\varepsilon^3}{T+1}$$

holds. By Markov's inequality, we have

$$\Pr_{R_1}\left[\Pr_{\mathcal{B}}[\mathcal{B}^{|f\rangle\setminus R_1, f(a,k,\cdot)}(\mathsf{st}_f, a, k)] \ge \frac{16\delta\varepsilon^2}{T+1}\right] \le \frac{4\delta\varepsilon}{16\delta} = \frac{\varepsilon}{4}.$$

Overall, the probability that x is good is at least $\delta\varepsilon^3/T(T+1) \cdot (1-\varepsilon/4)$. □

Proof of Claim 9

Proof of Claim 9. The proof is essentially the same as the proof of Claim 7, so we omit most detailed computations here. Let $p := \delta\varepsilon^3/T(T+1)$ and recall that $\Pr[x \in G] \ge p(1-\varepsilon/4)$ for all x, $\mathbb{E}[|G|] \ge pN(1-\varepsilon/4)$ and $\mathbb{E}[|G|] \le pN$. Let $\mathcal{R}$ be the probabilistic space of R_1 and μ be the uniform probabilistic measure on $\mathcal{R}$. Then we have

$$\begin{aligned}&\mathop{\mathbb{E}}_{R_1}[|G| \Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|f\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1]\\ &\ge pN(1-\varepsilon/4) \cdot \Pr_{\mathcal{B},a,k}[\mathcal{B}^{|f\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1].\end{aligned}$$

From this, we obtain

$$\mathop{\mathbb{E}}_{R_1}\left[\frac{|G(R_1,R_2)|}{\mathbb{E}[|G|]} \cdot \Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|f\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1]\right] \ge \frac{1}{2} + \varepsilon/4. \tag{4}$$

If we define another probabilistic measure $\nu(R_1) = \frac{|G(R_1)|}{\mathbb{E}[|G|]} \cdot \mu(R_1)$, then we obtain

$$\mathop{\mathbb{E}}_{\nu(R_1)}[\Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|f\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1]] \ge \frac{1}{2} + \varepsilon/4.$$

We say R_1 is very bad if it does not satisfy (C), and we denote the set of all very bad elements by B. Then it holds that $\nu(B) \le \varepsilon/8$, which implies

$$\mathop{\mathbb{E}}_{\nu:(R_1)\in\neg B}[\Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|f\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1]] \ge \frac{1}{2} + \varepsilon/8.$$

By standard averaging argument, for at least $(\varepsilon/8)$-fraction (with respect to ν) of not very bad R_1, denoted by S, we have

$$\Pr_{\mathcal{B},(a,k)\in G}[\mathcal{B}^{|f\rangle, f(a,k,\cdot)}(\mathsf{st}_f, a, k) \to 1] \ge \frac{1}{2} + \varepsilon/16.$$

Note that S is a set of good elements and we have

$$\nu(S) \ge \varepsilon/8 \cdot \nu(\neg B) \ge \varepsilon/8 - \varepsilon^2/64 \ge \varepsilon/16.$$

This implies

$$\mu(S) = \Omega(\varepsilon^4/T^2).$$

as desired. □

B.5 Proof of Claim 3

Proof of Claim 3. The proof is essentially the same as the proof of Claim 5. Recall that $p = c\varepsilon/T(T+1)$. Let H be the intersection of I and R_1. Then, since $\mathbb{E}[|H|] = p|I|$, by the Chernoff bound

$$\Pr_R[|H| \geq p|I|/2] \geq 1 - \exp(-p|I|/8) \geq 1 - \exp(-C/16) \geq 0.9$$

holds for $C \geq 16 \log 10$. Also,

$$\Pr_{R_1, \mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus (R_1 \setminus \{(a,k)\}) \times [L], f(a,k,\cdot)}(\mathsf{st}_f, a, k)] \leq 4c\varepsilon/(T+1)$$

holds by Lemma 3 and the similar argument to Claim 2. Markov's inequality implies

$$\Pr_{R_1}\left[\Pr_{\mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus (R_1 \setminus \{(a,k)\}) \times [L], f(a,k,\cdot)}(\mathsf{st}_f, a, k)] \geq \frac{160c}{T+1}\right] \leq \frac{1}{40}.$$

Let J be a subset of I whose elements satisfy (A) but not (B). Since two events are independent and (a, x) not satisfying (B) implies

$$\Pr_{\mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus (R_1 \setminus \{(a,k)\}) \times [L], f(a,k,\cdot)}(\mathsf{st}_f, a, k)] \geq \frac{160c}{T+1},$$

the probability that $(a, k) \in J$ is at most $p/40 = c\varepsilon/40T(T+1)$ for any $(a, k) \in I$. Thus, by Markov's inequality

$$\Pr_{R_1}\left[|J| \leq \frac{c\varepsilon|I|}{4T(T+1)}\right] \geq 0.9$$

holds. Overall, with probability at least 0.8, it holds that

$$|G| = |H| - |J| \geq \frac{c|I|}{2T(T+1)} - \frac{c|I|}{4T(T+1)} = \Omega\left(\frac{\varepsilon KN}{T^2}\right).$$

□

C Omitted Proofs in Section 4

C.1 Proof of Lemma 9

Before proving Lemma 9, we introduce the gentle measurement lemma [Win99, Aar05] and an auxiliary lemma shown by Aaronson and Rothblum [AR19].

Lemma 11 ([Aar05, Lemma 2.2]). *Suppose a 2-outcome measurement of a (mixed) quantum state ρ yields outcome 1 with probability at least $1 - \varepsilon$. Then we can recover a state ρ' such that $\mathrm{tr}(\rho, \rho') \leq \sqrt{\varepsilon}$ after the measurement where $\mathrm{tr}(\rho, \rho')$ denotes the trace distance between ρ and ρ'.*

Lemma 12 ([AR19, Corollary 16]). *Let ρ be a mixed state and let $S_1, \dots, S_m$ be quantum operations. Suppose that for all i, we have*

$$\mathrm{tr}(S_i(\rho), \rho) \leq \varepsilon_i.$$

Then

$$\mathrm{tr}(S_m(S_{m-1}(\cdots(S_1(\rho))), \rho) \leq \varepsilon_1 + \cdots + \varepsilon_m.$$

Then we prove Lemma 9.

Proof of Lemma 9. $\mathcal{B}$ works as follows: Set $\rho_0 = \rho$. For each $i = 1, 2, \cdots, n$, $\mathcal{B}$ applies the unitary $\mathcal{A}_i$ to ρ_{i-1}, measure the output register to obtain x_i', "recovers" the state before the measurement by using Lemma 11, applies the inverse unitary $\mathcal{A}_i'$, and lets ρ_i be the resulting state.

Then we analyze the $\mathcal{B}$'s success probability. Let S_i be the quantum operator that corresponds to the i-th loop in the execution of $\mathcal{B}$. By Lemma 11 (where we consider a projective measurement $(M_0 = I - |x_i\rangle\langle x_i|, M_1 = |x_i\rangle\langle x_i|)$, we have $\mathrm{tr}(S_i(\rho), \rho) \leq \frac{1}{3n^2}$. Then we have $\mathrm{tr}(\rho_i, \rho) = \mathrm{tr}(S_i(S_{i-1}(\dots S_1(\rho))), \rho) \leq \frac{i}{3n^2} \leq \frac{1}{3n}$ by Lemma 12. Therefore we have

$$\Pr[x_i' \neq x_i] \leq \mathrm{tr}(\rho_i, \rho) \leq \frac{1}{3n}.$$

By union bound, we obtain

$$\Pr[x_i' = x_i \text{ for all } i \in [n]] \geq \frac{2}{3}.$$

□

C.2 Proof of Claim 4

Proof of Claim 4. This can be proven similarly to the proof of Claim 2. Let H be the intersection of I and R. Then, since $\mathbb{E}[|H|] = p|I|$, by Chernoff's bound, we have

$$\Pr_R[|H| \geq p|I|/2] \geq 1 - \exp(-p|I|/8) \geq 1 - \exp(-C/16) \geq 0.9$$

where we used $C \geq 16 \ln 10$. Let

$$p_{\mathsf{Find}} := \Pr_{R,\mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus (R \setminus \{a,x\})}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x))].$$

By Lemma 3 and Remark 1, we have

$$p_{\mathsf{Find}} \leq 4D' \cdot \Pr_{R,\mathcal{C}}[(R \setminus \{a,x\}) \cap M \neq \emptyset : M \leftarrow \mathcal{C}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x))],$$

where $D' = T + 1$ is the query depth of $\mathcal{B}$ and $\mathcal{C}$ is an algorithm that works as follows:

$\mathcal{C}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x))$ Chooses $i \leftarrow \{1, \cdots, D'\}$; runs $\mathcal{B}^{\mathcal{O}_\emptyset^{SC}}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x))$ until (just before) the i-the query; then measures all query input registers in the computational basis and outputs the set M of measurement outcomes.

This is bounded by

$$4D' \cdot \Pr[R \cap M \neq \emptyset : M \leftarrow \mathcal{C}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x))] \leq 4T' \cdot p = \frac{4d}{T+2}$$

by the latter part of Lemma 3 and Remark 1. Markov's inequality implies

$$\Pr_R\left[\Pr_{\mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus (R \setminus \{(a,x)\})}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x), L)] \geq \frac{1}{576(T+2)}\right] \leq 2304d.$$

Let J be a subset of I whose elements satisfy (A) but not (B). Note that two events (A) and (B) are independent since (A) only depends on if $f(a,x) \in R$ and (B) depends on the other points (i.e., in $[K] \times [N] \setminus \{(a,x)\}$) being included in R. Since $(a,x) \in I$ not satisfying (B) implies

$$\Pr_{\mathcal{B}}[\mathsf{Find} : \mathcal{B}^{|f\rangle \setminus (R \setminus \{(a,x)\})}(|\widetilde{\mathsf{st}}_f\rangle, a, f(a,x))] \geq \frac{1}{576(T+2)},$$

the probability that $(a,x) \in J$ is at most $2304d \cdot p = 2304d^2/T'(T+2)$ for any $(a,x) \in I$. Thus by Markov's inequality

$$\Pr_R\left[|J| \leq \frac{23040d^2|I|}{T'(T+2)}\right] \geq 0.9$$

holds. Overall, with probability at least 0.8, it holds that

$$|G| = |H| - |J| \geq \frac{d|I|}{2T'(T+2)} - \frac{23040d^2|I|}{T'(T+2)} = \Omega\left(\frac{\varepsilon^2 KN}{T^2}\right)$$

as desired where we used $d \leq 1/46080$, $|I| \geq \varepsilon KN/4$, and $T' = \Theta(T/\varepsilon)$. ∎

D On Definition of Inversion Advantage in [NABT15]

Here, we discuss the difference between definitions of the inversion advantage of random permutations in this paper and in [NABT15].

For a random permutation $\mathcal{O} : [N] \to [N]$, Nayebi et al. [NABT15] defined the inversion advantage of $\mathcal{A}$ with advice $\mathsf{st}_{\mathcal{O}}$ as

$$\Pr_{\mathcal{O},x}[\Pr_{\mathcal{A}}[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, f(x)) \to x] > 2/3].$$

However, this definition is problematic in a cryptographic sense. For example, suppose that we have

$$\Pr_{\mathcal{A}}[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, f(x)) \to x] = 1/2$$

for all $\mathcal{O}$ and $x \in [N]$. In a cryptographic sense, this should be considered to be a fatal attack against the one-wayness. However, its advantage is 0 according to their definition.

To capture such an adversary, we define the inversion advantage as

$$\Pr_{\mathcal{O},\mathcal{A},x}[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, f(x)) \to x]$$

where the salting is omitted. By this definition, the advantage of the above described $\mathcal{A}$ is $1/2$, and the problem is resolved.

We note that we can convert an adversary $\mathcal{A}$ that has a large advantage in our definition into $\mathcal{A}'$ that has a large advantage in their definition by the amplitude amplification (Lemma 10 in Appendix A) as follows. Suppose that $\mathcal{A}$ is an adversary that makes at most T queries and has an advantage ε in our definition, i.e., we have

$$\Pr_{\mathcal{O},\mathcal{A},x}[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, f(x)) \to x] \geq \varepsilon.$$

Then by averaging argument, we have

$$\Pr_{\mathcal{O},x}[\Pr_{\mathcal{A}}[\mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, f(x)) \to x] \geq \varepsilon/2] \geq \varepsilon/2.$$

Then by the amplitude amplification, there exists an algorithm $\mathcal{A}'$ that runs $\mathcal{A}$ (and its inverse) $O(\varepsilon^{-1/2})$ times and satisfies

$$\Pr_{\mathcal{O},x}[\Pr_{\mathcal{A}'}[\mathcal{A}'^{|\mathcal{O}\rangle}(\mathsf{st}_{\mathcal{O}}, f(x)) \to x] \geq 2/3] \geq \varepsilon/2.$$

By using the above conversion, we can translate Nayebi et al.'s bound into a bound with our definition. From the above argument, if the number of queries by $\mathcal{A}$ is T, then the number of queries by $\mathcal{A}'$ is $T' = O(T\varepsilon^{-1/2})$. By their result, we have $\varepsilon = \widetilde{O}(\frac{ST'^2}{N})$, which implies $\varepsilon^2 = \widetilde{O}(\frac{ST^2}{N})$. Therefore, there occurs a quadratic loss if we translate their bound into one in our setting.

E Extended Comparison Table

Table 2 compares our bounds in the QROM-AI with the previously shown bounds in the ROM-AI. The "Security bounds in ROM-AI" and "Best known attacks in ROM-AI" columns are taken from [DGK17] and [CDGS18], and the "Security bounds in QROM-AI" and "Best known attacks in QROM-AI" columns are taken from Table 1.

Here, we briefly explain how we derive "Best known attacks in QROM-AI". Basically, the best known attacks we are aware of are to just apply quantum attacks *without auxiliary inputs* (i.e., Grover's algorithm [Gro96] for OWFs, PRGs, (pq/q)PRFs, and (pq/q)MACs, BHT algorithm [BHT97] for CRHFs, and Boneh-Zhandry's attack [BZ13] for qMACs) or best known *classical* attacks that make use of auxiliary inputs (i.e., Hellman's attack [Hel80] for OWFs and (pq/q)MACs, De et al.'s attack [DTT10] for PRGs and (pq/q)PRFs, and the trivial attack for CRHFs). Though quantum attacks possibly exist that utilize classical auxiliary inputs that achieve better bounds than classical ones, we are not aware of such an algorithm for these primitives. One can find a discussion on why we cannot directly combine the Grover's algorithm and known classical attacks that utilize auxiliary inputs in [NABT15].

	Security bounds in ROM-AI	Security bounds in QROM-AI (Ours)	Best known attacks in ROM-AI	Best known attacks in QROM-AI
OWFs	$\frac{ST}{K\alpha} + \frac{T}{\alpha}$	$\left(\frac{ST^2}{K\alpha} + \frac{T^2N}{\alpha^2}\right)^{1/2}$	$\min\left\{\frac{ST}{K\alpha}, \left(\frac{S^2T}{K^2\alpha^2}\right)^{1/3}\right\} + \frac{T}{\alpha}$	$\min\left\{\frac{ST}{K\alpha}, \left(\frac{S^2T}{K^2\alpha^2}\right)^{1/3}\right\} + \frac{T^2}{\alpha}$
CRHFs	$\frac{S}{K} + \frac{T^2}{M}$	unknown	$\frac{S}{K} + \frac{T^2}{M}$	$\frac{S}{K} + \frac{T^3}{M}$
PRGs	$\left(\frac{ST}{KN}\right)^{1/2} + \frac{T}{N}$	$\left(\frac{ST^4}{KN} + \frac{T^4}{N}\right)^{1/6}$	$\left(\frac{ST}{KN}\right)^{1/2} + \frac{T}{N}$	$\left(\frac{ST}{KN}\right)^{1/2} + \frac{T^2}{N}$
c/pqPRFs	$\left(\left(\frac{S(T+Q_{\mathsf{prf}})}{KN}\right)^{1/2} + \frac{T}{N}\right)$	$\left(\frac{ST^4}{KN} + \frac{T^4}{N}\right)^{1/4} + Q_{\mathsf{prf}}\left(\frac{ST^2}{KN}\right)^{1/6}$	$\left(\frac{ST}{KN}\right)^{1/2} + \frac{T}{N}$	$\left(\frac{ST}{KN}\right)^{1/2} + \frac{T^2}{N}$
c/pqMACs	$\frac{S(T+Q_{\mathsf{mac}})}{KN} + \frac{T}{N} + \frac{1}{M}$	$\left(\frac{ST^4}{KN} + \frac{T^4}{N} + \frac{1}{M}\right)^{1/3}$	$\min\left\{\frac{ST}{KN}, \left(\frac{S^2T}{K^2N^2}\right)^{1/3}\right\} + \frac{T}{N} + \frac{1}{M}$	$\min\left\{\frac{ST}{KN}, \left(\frac{S^2T}{K^2N^2}\right)^{1/3}\right\} + \frac{T^2}{N} + \frac{1}{M}$
qPRFs	N/A	unknown	N/A	$\left(\frac{ST}{KN}\right)^{1/2} + \frac{T^2}{N}$
qMACs	N/A	unknown	N/A	$\min\left\{\frac{ST}{KN}, \left(\frac{S^2T}{K^2N^2}\right)^{1/3}\right\} + \frac{T^2}{N} + \frac{Q_{\mathsf{mac}}}{M}$

Table 2. Security bounds and best known attacks using an S-bit classical auxiliary input and T classical/quantum queries for "salted" constructions of primitives in the ROM-AI and QROM-AI. The first three primitives (unkeyed primitives) are constructed from a random oracle $\mathcal{O} : [K] \times [N] \to [M]$ where $[K]$ is the domain of the salt, $[N]$ is the domain of the input (or the seed for PRGs), $[M]$ is the domain of the outputs, and we let $\alpha := \min(N, M)$. The latter four primitives (keyed primitives) are constructed from a random oracle $\mathcal{O} : [K] \times [N] \times [L] \to [M]$ where $[K]$ is the domain of the salt, $[N]$ is the domain of the key, $[L]$ is the domain of the inputs, and $[M]$ is the domain of the outputs (or authenticators for MACs). Q_{prf} denotes the number of queries to the PRF oracle and Q_{mac} denotes the number of queries to the MAC oracle. In the "c/pqPRFs" and "c/pqMACs" rows, the bounds and attacks in the ROM-AI refer to them in the fully classical setting whereas those in the QROM-AI refer to them in the post-quantum setting where only the random oracle is quantumly accessible. We omit constant factors and logarithmic terms for simplicity.