Verifiable Quantum Advantage without Structure

Takashi Yamakawa*[1] and Mark Zhandry[2,3]

[1]NTT Social Informatics Laboratories
[2]Princeton University
[3]NTT Research

April 5, 2022

Abstract

We show the following hold, unconditionally unless otherwise stated, relative to a random oracle with probability 1:

- There are NP *search* problems solvable by BQP machines but not BPP machines.
- There exist functions that are one-way, and even collision resistant, against classical adversaries but are easily inverted quantumly. Similar separations hold for digital signatures and CPA-secure public key encryption (the latter requiring the assumption of a classically CPA-secure encryption scheme). Interestingly, the separation does not necessarily extend to the case of other cryptographic objects such as PRGs.
- There are unconditional publicly verifiable proofs of quantumness with the minimal rounds of interaction: for uniform adversaries, the proofs are non-interactive, whereas for non-uniform adversaries the proofs are two message public coin.
- Our results do not appear to contradict the Aaronson-Ambanis conjecture. Assuming this conjecture, there exist publicly verifiable certifiable randomness, again with the minimal rounds of interaction.

By replacing the random oracle with a concrete cryptographic hash function such as SHA2, we obtain plausible Minicrypt instantiations of the above results. Previous analogous results all required substantial structure, either in terms of highly structured oracles and/or algebraic assumptions in Cryptomania and beyond.

*This work was done in part while the author was visiting Princeton University.

Contents

1 Introduction

Can NP *search problems have a super-polynomial speed-up on quantum computers?* This is one of the oldest and most important questions in quantum complexity.

The first proposals for such quantum advantage were relative to highly structured oracles. Examples include Simon's oracle [Sim97], or more generally periodic oracles, as well as the Bernstein–Vazirani oracle [BV93] and welded trees [CCD+03].

The first non-relativized quantum advantage for NP problems is due to Shor's famous algorithm for factoring integers and computing discrete logarithms [Sho94]. Since Shor's algorithm, other non-relativized NP problems with quantum advantage include solving Pell's equation [Hal02] and matrix group membership [BBS09]. While the technical details of all these examples are very different, these problems can all be seen as non-relativizing instantiations of *periodic* oracles.

While the above non-relativizing problems are certainly easy on a quantum computer, the classical hardness can only be conjectured since, in particular, the classical hardness would imply $\mathsf{P} \neq \mathsf{NP}$, or an analogous statement if one considers probabilistic algorithms. The problem is that, when instantiating an oracle with real-world computational tasks, non-black-box algorithms may be available that render the problem classically easy, despite the oracle problem being hard. For example, index calculus methods [Adl79] yield sub-exponential time attacks for factoring and discrete logarithms, despite black box period-finding being exponentially hard.

To make matters worse, for the known NP search problems with plausible quantum advantage, the classical hardness is widely believed to be a much stronger assumption than $\mathsf{P} \neq \mathsf{NP}$, since the problems have significant algebraic structure and are not believed to be NP-complete. In particular, all NP search problems we are aware of yielding a super-polynomial quantum advantage are *Cryptomania* assumptions [Imp95], in the sense that their classical hardness yields public key encryption.[1] This puts the assumptions needed for an NP quantum advantage quite high in the assumption hierarchy.

Quantum speed-ups and structure. The above tasks demonstrating speed-ups, both relativized and non-relativized, all have one thing in common: significant "structure." It is natural to wonder whether such structure is necessary. In the oracle-free non-relativized setting, a natural interpretation of this question could be if Minicrypt assumptions—those that give symmetric key but not public key cryptography—can be used to give a quantum advantage. Minicrypt assumptions, such as the one-wayness of SHA2, lack the algebraic structure needed in typical super-polynomial quantum speed-ups. In the oracle setting, this could mean, for example, proving unconditional quantum advantage relative to a uniformly *random* oracle, which is generally seen as beeing structure-less.

Prior work on this topic could be interpreted as negative. As observed above, all non-relativizing NP problems demonstrating quantum advantage imply, or are closely related to problems that imply, public key cryptography. In the random oracle setting, the evidence is even stronger. The most natural problems to reason about—one-wayness and collision resistance of the random oracle, and generalizations—provably only have a polynomial quantum advantage [BBBV97, AS04, Yue14, Zha15]. Additional evidence is given by Aaronson and Ambanis [AA14], who build on work of Beals et al. [BBC+98]. They consider the following conjecture, dating back to at least 1999:

[1] Matrix group membership includes discrete logarithms as a special case. For a public key system based on Pell's equations, see [Pad06].

Conjecture 1.1 (Paraphrased from [AA14])**.** *Let Q be a quantum algorithm with* Boolean *output that makes T queries to a random oracle $\mathcal{O}$, and let $\epsilon, \delta > 0$. Then there exists a deterministic classical algorithm C that makes $\mathsf{poly}(T, 1/\epsilon, 1/\delta)$ queries, such that*

$$\Pr_{\mathcal{O}}\left[\ \left|\ C^{\mathcal{O}}() - \Pr[Q^{\mathcal{O}}() = 1]\ \right| \leq \epsilon\ \right] \geq 1 - \delta\ ,$$

Where the expectation is over the randomness of Q.

Aaronson and Ambanis give some evidence for Conjecture 1.1, by reducing it to a plausible *mathematical* conjecture closely related to known existing results. If Conjecture 1.1 is true, any quantum *decision* algorithm Q making queries to a random oracle can be simulated classically with only polynomially-more queries.

Note that the conjectured classical simulator may be *computationally* inefficient, and indeed we would expect it to be if, say, Q ignored its oracle and just factored integers. But for any particular algorithm Q, proving computational inefficiency amounts to an unconditional hardness result, which is beyond the reach of current complexity theory. Thus, Conjecture 1.1, if true, essentially shows that random oracles are equivalent to the non-relativizing world with respect to NP decision problems, and cannot be used to provide provable quantum advantage for such problems.

1.1 Our Results

In this work, we make progress toward justifying super-polynomial quantum advantage for NP problems, under less structured oracles or milder computational assumptions. We show, perhaps surprisingly, that for certain *search* problems in NP, random oracles do in fact give provable unconditional super-polynomial quantum speed-ups.

Random oracles. Our starting point is to prove the following theorem:

Theorem 1.2 (Informal)**.** *Relative to a random oracle with probability 1, there exists a non-interactive proof of quantumness, with unconditional security against any computationally-unbounded uniform adversary making a polynomial number of classical queries.*

Here, a proof of quantumness [BCM+18] is a protocol between a quantum prover and classical verifier (meaning in particular that messages are classical) where no cheating classical prover can convince the verifier. By being non-interactive, our protocol is also publicly verifiable. Prior LWE-based proofs of quantumness [BCM+18, BKVV20] lacked verifiability. The only previous publicly verifiable proof of quantumness [AGKZ20] required highly non-trivial structured oracles.

Remark 1. *We note the restriction to uniform adversaries is necessary in the non-interactive setting, as a non-uniform adversary can simply have a proof hardcoded. Our protocol also readily gives a two-message public coin (and hence also publicly verifiable) protocol against non-uniform adversaries, which is the best one can hope for in the non-uniform setting.*

Theorem 1.2 has a number of interesting immediate consequences:

Corollary 1.3. *Relative to a random oracle, there exists an* NP *search problem that is solvable by* BQP *machines but not by* BPP *machines.*

Our construction also readily adapts to give one-way functions that are classically secure but quantum insecure. We can alternatively use minimal-round proofs of quantumness generically to give a one-way function separation, and even a collision resistance separation:

Theorem 1.4. *Relative to a random oracle, there exists a compressing function that is collision resistant against any computationally unbounded uniform adversary making a polynomial number of classical queries, but is not even one-way against quantum adversaries.*

Using results from [YZ21], we also obtain an unconditional analogous separation for digital signatures and CPA-secure public key encryption (the latter requiring assuming classically CPA-secure public key encryption). Previous such results required LWE (in the case of signatures) or highly structured additional oracles (in the case of CPA-secure encryption).

Our results do not appear to contradict Conjecture 1.1, since they are for *search* problems as opposed to *decision* problems. In particular, our quantum algorithm for generating proofs of quantumness/breaking the one-wayness does not compute a function, but rather sample from a set of possible values. Assuming Conjecture 1.1 shows that this is inherent. We leverage this feature to yield the following:

Theorem 1.5. *Assuming Conjecture 1.1, relative to a random oracle there exists a one- (resp. two-) message certifiable randomness protocol against a single uniform (resp. non-uniform) quantum device. By adding a final message from the verifier to the prover, our protocols become public coin and publicly verifiable.*

Here, certifiable randomness [BCM+18] means the classical verifier, if it accepts, is able to expand a small random seed s into a truly random bit-string $x, |x| \gg |s|$, with the aid of a single quantum device. Conditioned on the verifier accepting, x remains truly random even if the device is adversarial.

We note that our results are the best possible: if the final message is from prover to verifier, the protocols cannot be publicly verifiable. Indeed, the prover could force, say, the first output bit to be 0 by generating a candidate final message, computing the what the outputted string would be, and then re-sampling the final message until the first output bit is 1. Our one- and two-message protocols therefore require verifier random coins that are kept from the prover. In our protocols, however, these secret random coins can be sampled and even published after the prover's message. The result is that, by adding a final message from the verifier, our protocols are public coin and publicly verifiable.

Instantiating the random oracle. We next instantiate the random oracle in the above construction with a standard-model cryptographic hash, such as SHA2. We cannot hope to prove security unconditionally. Nevertheless, the resulting construction is quite plausibly secure. Indeed, it is common practice in cryptography to prove security of a hash-based protocol relative to random oracles [BR93], and then assume that security also applies when the random oracle is replaced with a concrete well-designed cryptographic hash. While there are known counter-examples to the random oracle assumption [CGH98], they are quite contrived and are not known to apply to our construction.

We thus obtain a plausible assumption on, say SHA2, under which non-interactive proofs of quantumness exist. This gives a completely new approach to non-relativized quantum advantage. What's more, it is widely believed that SHA2 is only capable of yielding symmetric key cryptosystems. Impagliazzo and Rudich [IR89] show that there is no classical black box construction of public key encryption from cryptographic hash functions, and no quantum or non-black box techniques are known to overcome this barrier. In fact, what [IR89] show is that, in the world of computationally unbounded but query bounded (classical) attackers, random oracles cannot be used to construct public key encryption. But this is exactly the setting of the random oracle model we consider.

Therefore, by instantiating the random oracle with a well-designed hash such as SHA2, we obtain a Minicrypt construction of a proof of quantumness. We likewise obtain candidate Minicrypt examples of NP search problems in BQP\BPP, functions that are classically one-way but quantumly easy, and even certifiable randomness.

1.2 Discussion

Other sources of quantum advantage. Other candidates for super-polynomial quantum speed-ups are known. Aaronson and Arkhipov [AA11] and Bremner, Jozsa, and Shepherd [BJS10] give a sampling task with such a speed-up, based on plausible complexity-theoretic constructions. Similar sampling tasks are at the heart of current real-world demonstrations of quantum advantage. More recently, Brakerski et al. [BCM+18] provided a proof of quantumness from the Learning With Errors (LWE) assumption.

We note, however, that none of the these alternate sources of quantum advantage correspond to NP search problems.

Why NP search problems? Most real-life problems of interest can be phrased as NP search problems, so it is a natural class of problems to study. Our work gives the first evidence besides period finding of a quantum advantage for this class.

Moreover, NP means that solutions can be efficiently verified. For existing sampling-based demonstrations of quantum advantage [AA11, BJS10], verification is roughly as hard as classically sampling. Proofs of quantumness from LWE [BCM+18] do admit verification, but the verifier must use certain secrets computed during the protocol in order to verify. This means that only the verifier involved in the protocol is convinced of the quantumness of the prover.

In contrast, using an NP problem means anyone can look at the solution and verify that it is correct. Moreover, our particular instantiation allows for sampling the problems obliviously, meaning we obtain a *public coin* proof of quantumness where the verifier's message is simply uniform random coins. Against uniform adversaries, we can even just set the verifier's message to $000\cdots$, eliminating the verifier's message altogether.

The QROM. In classical cryptography, the Random Oracle Model (ROM)[BR93] models a hash function as a truly random function, and proves security in such a world. This model is very important for providing security justifications of many practical cryptosystems.

Boneh et al. [BDF+11] explain that, when moving to the quantum setting, one needs to model the random oracle as a *quantum random oracle model* (QROM). Many works (e.g. [Zha12, TU16, SXY18, KLS18, KYY18, LZ19, DFMS19, CMS19]) have been devoted to lifting classical ROM results to the QROM. To date, most of the main classcal ROM results have successfully been lifted. This leads to a natural question: do all ROM results lift to the QROM?

Recently, Yamakawa and Zhandry [YZ21], leveraging recent proofs of quantumness [BKVV20] in the random oracle, give a counter-example assuming the hardness of learning with errors (LWE). Their counter-examples were limited to highly interactive security models such as digital signatures and CCA-secure public key encryption.

By relying on LWE, [YZ21] left open the possibility that *unconditional* ROM results may all lift to the QROM. Our proof of quantumness refutes this, showing that the ROM and QROM are separated even in the unconditional setting. Our results also give separations for many more objects, especially for objects like one-way functions and collision resistance which have essentially non-interactive security experiments.

1.3 Overview

Let Σ be an exponentially-sized alphabet, and $C \subseteq \Sigma^n$ be an error correcting code over Σ. Let $O : \Sigma \to \{0,1\}$ be a function. Consider the following function $f_C^O : C \to \{0,1\}^n$ derived from C, O:

$$f_C^O(c_1, \ldots, c_n) = (O(c_1), \ldots, O(c_n))$$

In other words, f_C^O simply applies O independently to each symbol in the input codeword. We will model O as a uniformly random function. Note that if f were applied to arbitrary words in Σ^n, then it would just be the parallel application of a function with one-bit outputs, which can be trivially inverted. By restricting the domain to only codewords, we show, under a suitable choice of code elaborated on below, that:

- f_C^O is unconditionally one-way against classical probabilistic algorithms making polynomially-many queries to O. It is even infeasible to find $c \in C$ such that $f_C^O(c) = 0^n$.
- There exists a quantum algorithm which, given any $y \in \{0,1\}^n$, samples statistically close to uniformly from the set of pre-images $c \in C$ such that $f_C^O(c) = y$.

From these properties, we immediately obtain a weak version of Theorem 1.4 which only considers classical one-wayness. We explain in Section 7.2 how to obtain the full Theorem 1.4. To prove quantumness, one simply produces $c \in C$ such that $f_C^O(c) = 0^n$, giving Theorem 1.2. Since one-way functions are in NP, this also immediately gives Corollary 1.3. We now explain how we justify these facts about f_C^O.

Classical hardness. Assume C satisfies the following properties: (1) the set of symbols obtained at each position are distinct, and (2) C is information-theoretically **list-recoverable**. Here, we take list-recoverability to mean that, given polynomial-sized sets $S_i, i \in [n]$ of possible symbols for each position, there exist a sub-exponential sized (in n) list of codewords c such that $c_i \in S_i$ for all $i \in [n]$. The list size remains sub-exponential even if we include codewords such that $c_i \notin S_i$ for a few positions.

Property (1) can be obtained generically by replacing $\Sigma \mapsto [n] \times \Sigma$, where $(c_1, \ldots, c_n) \mapsto ((1, c_1), \ldots, (n, c_n))$. Property (2) is satisfied by folded Reed-Solomon codes, as shown by Guruswami and Rudra [GR08].

Assuming (1) and (2), we can show classical hardness. Fix an image y. We can assume without loss of generality that the adversary always evaluates $f_C^O(c)$ for any pre-image c it outputs. Suppose for our discussion here that all queries to O were made in parallel. Then any polynomial-sized set of queries corresponds to a collection of S_i. List recoverability means that there are at most $2^{n^c}, c < 1$ codewords consistent with the S_i. For each consistent codeword, the probability of being a pre-image of y is at most 2^{-n} over the choice of random oracle. Union-bounding over the list of consistent codewords shows that the probability that *any* consistent codeword is a pre-image is exponentially small. With some effort, we can show the above holds even for adaptively chosen queries.

Remark 2. *Haitner et al. [HIOS15] construct a very similar hash function from list-recoverable codes. Their hash functions assumes a multi-bit O, but then XORs the results together, rather than concatenating them. They prove that their hash function is collision-resistant. Our proof of one-wayness is based on a similar idea to their proof of collision-resistance. Our novelty, and what does not appear to be possible for their construction, is the quantum pre-image finder, which we discuss next.*

We note that we could, similar to [HIOS15], prove the collision resistance of f_C^O by choosing C to have an appropriate rate. However, our quantum pre-image finder constrains C to having a rate where we only know how to prove one-wayness. Proving Theorem 1.4 therefore requires a different construction, which we elaborate on in Section 7.2.

Quantum easiness. Our algorithm can be seen as loosely inspired by Regev's quantum reduction between SIS and LWE [Reg05]. Given an image y, our goal will be to create a uniform superposition over pre-images of y:

$$|\psi_y\rangle \propto \sum_{c\in C: f_C^O(c)=y} |c\rangle$$

We can view $|\psi_y\rangle$ as the point-wise product of two vectors:

$$|\phi\rangle \propto \sum_{c\in C} |c\rangle\,, \qquad \text{and} \qquad |\tau_y\rangle \propto \sum_{c\in \mathbf{\Sigma^n}: f_C^O(c)=y} |c\rangle$$

Obseve that $|\tau_y\rangle$ looks like $|\psi_y\rangle$, except that the domain is no longer constrained to codewords. Once we have the state $|\psi_y\rangle$, we can simply measure it to obtain a random pre-image of y. We will show how to construct $|\psi_y\rangle$ in reverse: we will show a sequence of reversible transformations that transform $|\psi_y\rangle$ into states we can readily construct. By applying these transformations in reverse we obtain $|\psi_y\rangle$. To do so, we will now impose that Σ is a vector space over $\mathbb{F}_q$ for some prime q, and that C is **linear** over $\mathbb{F}_q$. This means there is a dual code $C^\perp$, such that $c\cdot d = 0$ for all $c\in C, d\in C^\perp$.

We now consider the quantum Fourier transform QFT of $|\psi_y\rangle$. Write:

$$|\widehat{\phi}\rangle := \mathsf{QFT}|\phi\rangle \propto \sum_{c\in\Sigma^n} \alpha_c|c\rangle = \sum_{c\in C^\perp} |c\rangle$$
$$|\widehat{\tau_y}\rangle := \mathsf{QFT}|\tau_y\rangle \propto \sum_{c\in\Sigma^n} \beta_{y,c}|c\rangle$$

Above, we used the fact that the QFT of a uniform superposition over a liner space is just the uniform superposition over the dual space. Then, by the Convolution Theorem, the QFT of $|\psi_y\rangle$ is the convolution of $|\widehat{\phi}\rangle$ and $|\widehat{\tau_y}\rangle$:

$$|\widehat{\psi_y}\rangle := \mathsf{QFT}|\psi_y\rangle \propto \sum_{c,e\in\Sigma^n} \alpha_c\beta_{y,e}|c+e\rangle = \sum_{c\in C^\perp, e\in\Sigma^n} \beta_{y,e}|c+e\rangle$$

The next step is to decode c and e from $c+e$; assuming we had such a decoding, we can apply it to obtain the state proportional to

$$\sum_{c\in C^\perp, e\in\Sigma^n} \beta_{y,e}|c,e\rangle = |\widehat{\phi}\rangle|\widehat{\tau_y}\rangle$$

We can then construct $|\widehat{\phi}\rangle$ as the QFT of $|\phi\rangle$, which we can generate using the generator matrix for C. We will likewise construct $|\widehat{\tau_y}\rangle$ as the QFT of $|\tau_y\rangle$. To construct $|\tau_y\rangle$, we note that $|\tau_y\rangle$ is a product of n states that look like:

$$|\tau_{i,y_i}\rangle \propto \sum_{\sigma\in\Sigma: O(\sigma)=y_i} |\sigma\rangle$$

Since each y_i is just a single bit, we can construct such states by applying O to a uniform superposition of inputs, measuring the result, and starting over if we obtain the incorrect y_i.

It remains to show how to decode c, e from $c + e$. We observe that $|\widehat{\tau}_{i,y_i}\rangle$ has roughly half of its weight on 0, whereas the remaining half the weight is essentially uniform (though with complex phases) since O is a random function. This means we can think of e as a vector where each symbol is 0 with probability $1/2$, and random otherwise. In other words, $c + e$ is a noisy version of c following an analog of the binary symmetric channel generalized to larger alphabets. If the dual code $C^\perp$ were efficiently decodable under such noise, then can decode c (and hence e) from $c + e$.

Toward that end, we show that c is uniquely information-theoretically decodable (whp) provided the rate of $C^\perp$ is not too high. In our case where C is a folded Reed-Solomon code, $C^\perp$ is essentially another Reed-Solomon code, and we can decode efficiently using **list-decoding** algorithms [GS99]. We can show that the list-decoding results in a unique codeword (whp) for the above described error distribution assuming C to have an appropriate rate.

There are a couple important caveats with the above. First is that, to use list-recoverability to prove one-wayness, we actually needed to augment C, which broke linearity. This is easily overcome by only applying the QFT to the linear part of C.

More importantly, and much more challenging, we can only decode $c + e$ as long as e has somewhat small Hamming weight. While such e occur with overwhelming probability, there will always be a negligible fraction of decoding errors. The problem is that the constant of proportionality in the Convolution Theorem is exponentially large, and therefore the negligible decoding errors from our procedure could end up being blown up and drowning out $|\widehat{\psi}_y\rangle$. This is not just an issue with our particular choice of decoding algorithm, as for large enough Hamming wieght decoding errors are guaranteed. What this means is that the map $|\widehat{\phi}\rangle|\widehat{\tau}_y\rangle \mapsto |\widehat{\psi}_y\rangle$ is not even unitary, and $|\widehat{\psi}_y\rangle$ is not even unit norm.

By exploiting the particular structure of our coding problem and the uniform randomness of the oracle O, we are able to resolve the above difficulties and show that our algorithm does, in fact, produce pre-images of y as desired.

Certifiable randomness. We next explain that *any* efficient quantum algorithm for inverting f_C^O likely produces random inputs. After all, suppose there was an alternative quantum algorithm which inverted f_C^O, such that its output on any given y is deterministic. If we look at any single bit of the output, then Conjecture 1.1 would imply that this bit can be simulated by a polynomial-query classical algorithm. By applying Conjecture 1.1 to every bit of output, we thus obtain a classical query algorithm for inverting f_C^O, which we know is impossible.

This immediately gives us a proof of entropy: the prover generates a pre-image c of an arbitrary y (even $y = 0^n$), and the verifier checks that $f_C^O(c) = y$. If the check passes, the verifier can be convinced that c was not deterministically generated, and therefore has some randomness. By using the fact that f_C^O is one-way even against sub-exponential-query algorithms, we can show that the min-entropy must be polynomial.

Once we have a string with min-entropy, we can easily get uniform random bits by having the verifier extract using a private random seed.

Extension to non-uniform adversaries. Note that the above results all considered fixing an adversary first, and then sampling a random oracle. A standard complexity theoretic argument shows that, in the case of uniform adversaries, we can switch the order of quantifiers, and choose the random oracle first and then the adversary.

For non-uniform adversaries, we have to work harder, and direct analogs of the results above may in fact be impossible: for example, a non-uniform adversary (chosen after the random oracle) could have a valid proof of quantumness hardcoded.

For proofs of quantumness, we can leverage the "salting defeats preprocessing" result of [CGLQ20] to readily get a two-message public coin proof of quantumness against non-uniform attackers. For certifiable entropy/randomness, this also works, except the known bounds would end up requiring the verifiers message to be longer than the extracted string. This is a consequence of leveraging the sub-exponential one-wayness of f_C^O to obtain polynomially-many random bits. Since the verifier's message must be uniform, this would somewhat limit the point of a proof of randomness. We show via careful arguments how to overcome this limitation, obtaining two message proofs of randomness where the verifier's message remains small.

1.4 Organization

The remainder of the paper is organized as follows. Section 2 gives some basic preliminaries, including for quantum computation. Section 3 defines the various objects we will be considering and gives some basic relations. Section 4 discusses the properties of error correcting codes we will need. Section 5 gives a technical lemma that is needed to prove the correctness of our protocol, that may be more broadly useful. Section 6 gives our proof of quantumness, while Section 7 uses this to give separations for various cryptographic primitives. Finally, Section 8 gives our proofs of randomness.

2 Preliminaries

Basic notations. We use λ to mean the security parameter throughout the paper. For a set X, $|X|$ is the cardinality of X. We denote by $x \xleftarrow{\$} X$ to mean that x is uniformly taken from X. For a distribution D over a set X, we denote by $x \xleftarrow{\$} D$ to mean that $x \in X$ is taken according to the distribution D. For sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathsf{Func}(\mathcal{X},\mathcal{Y})$ denotes the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. For a positive integer n, $[n]$ means a set $\{1,...,n\}$. For a random variable X, $\mathbb{E}[X]$ denotes its expected value. For random variables X and X', $\Delta(X,X')$ denotes the statistical distance between X and X'. For a random variable X, $H_\infty(X)$ denotes the min-entropy of X, i.e., $H_\infty(X) = -\log \max_x \Pr[X=x]$. For a quantum or randomized classical algorithm $\mathcal{A}$, we denote $y \xleftarrow{\$} \mathcal{A}(x)$ to mean that $\mathcal{A}$ outputs y on input x. For a randomized classical algorithm $\mathcal{A}$, we denote $y \leftarrow \mathcal{A}(x;r)$ to mean that $\mathcal{A}$ outputs y on input x and randomness r.

Notations for quantum states. For a not necessarily normalized state $|\psi\rangle$, we denote by $\| |\psi\rangle \|$ to mean its Euclidean norm. For not necessarily normalized quantum states $|\psi\rangle$ and $|\phi\rangle$ and $\epsilon > 0$, we denote by $|\psi\rangle \approx_\epsilon |\phi\rangle$ to mean $\| |\psi\rangle - |\phi\rangle \| \leq \epsilon$. We simply write $|\psi\rangle \approx |\phi\rangle$ to mean $|\psi\rangle \approx_{\mathsf{negl}(\lambda)} |\phi\rangle$. By the triangle inequality, if we have $|\psi\rangle \approx_\epsilon |\phi\rangle$ and $|\phi\rangle \approx_\delta |\tau\rangle$, then we have $|\psi\rangle \approx_{\epsilon+\delta} |\tau\rangle$.

For not necessarily normalized quantum states $|\psi\rangle$ and $|\phi\rangle$, we denote by $|\psi\rangle \propto |\phi\rangle$ to mean that $|\psi\rangle = C|\phi\rangle$ for some $C \in \mathbb{C}$.

Classical/quantum random oracle model. In the classical random oracle model (CROM) [BR93], a random function H is chosen at the beginning, and every party (including honest algorithms of a protocol whose security is analyzed and an adversary) can classically access H.[2] The quan-

[2]The classical random oracle model is often just referred to as the ROM, but we call it CROM to emphasize that the oracle access is classical.

tum random oracle model (QROM) [BDF+11] is defined similarly except that the access to H can be quantum. In other words, a quantumly-accessible classical oracle that applies a unitary $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle$ is available. See Section 3 for more detailed treatment of these models.

2.1 Finite Fields

For a prime power $q = p^r$, $\mathbb{F}_q$ denotes a field of order q. We use this notation throughout the paper, and whenever we write $\mathbb{F}_q$, q should be understood as a prime power. We denote by $\mathbf{0}$ to mean $(0, ..., 0) \in \mathbb{F}_q^n$ where n will be clear from the context. For $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{F}_q^n$ and $\mathbf{y} = (y_1, ..., y_n) \in \mathbb{F}_q^n$, we define $\mathbf{x} \cdot \mathbf{y} := \sum_{i=1}^n x_i y_i$.

We often consider vectors $\mathbf{x} \in \Sigma^n$ over the alphabet $\Sigma = \mathbb{F}_q^m$. We identify Σ^n and $\mathbb{F}_q^{nm}$ in the canonical way, i.e., we identify $((x_1, \dots, x_m), \dots, (x_{(n-1)m+1}, \dots, x_{nm})) \in \Sigma^n$ and $(x_1, x_2, \dots, x_{nm}) \in \mathbb{F}_q^{nm}$. For $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n) \in \Sigma^n$ and $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_n) \in \Sigma^n$, we define $\mathbf{x} \cdot \mathbf{y} := \sum_{i=1}^n \mathbf{x}_i \cdot \mathbf{y}_i$.

The trace function $\mathrm{Tr} : \mathbb{F}_q \to \mathbb{F}_p$ is defined by[3]

$$\mathrm{Tr}(x) := \sum_{i=0}^{r-1} x^{p^i}.$$

The trace function is $\mathbb{F}_p$-linear, i.e., for any $a, b \in \mathbb{F}_p$ and $x, y \in \mathbb{F}_q$, we have

$$\mathrm{Tr}(ax + by) = a\mathrm{Tr}(x) + b\mathrm{Tr}(y).$$

For any $\mathbf{x} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\}$, we have

$$\sum_{\mathbf{y} \in \mathbb{F}_q^n} \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{y})} = 0. \tag{1}$$

The multiplicative group $\mathbb{F}_q^*$ of $\mathbb{F}_q$ is cyclic, and thus there is an element $\gamma \in \mathbb{F}_q^*$ such that

$$\{\gamma^i\}_{i \in [q-1]} = \mathbb{F}_q^*.$$

For $\mathbf{x} \in \mathbb{F}_q^n$, we denote by $\mathsf{hw}(\mathbf{x})$ to mean the Hamming weight of $\mathbf{x}$, i.e., $\mathsf{hw}(\mathbf{x}) := |\{i \in [n] : x_i \neq 0\}|$ where $\mathbf{x} = (x_1, \dots, x_n)$. For $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^n$ and a subset $S \subseteq [n]$, we denote by $\mathbf{x}_S$ to mean $(x_i)_{i \in S}$.

2.2 Quantum Fourier Transform over Finite Fields

We review known facts on quantum Fourier transform over finite fileds [dBCW02, vDHI06]. Though it is usually considered over quantum systems whose alphabet is a finite filed $\mathbb{F}_q$, we consider those over the alphabet $\Sigma = \mathbb{F}_q^m$ for some positive integer m. Since Σ^n can be identified with $\mathbb{F}_q^{nm}$, this is no more than notational convention.

On a quantum system over Σ^n, a quantum Fourier transform is a unitary denoted by QFT such that

$$\mathsf{QFT}|\mathbf{x}\rangle = \frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{z} \in \Sigma^n} \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{z})} |\mathbf{z}\rangle$$

where $\omega_p := e^{2\pi i/p}$. A quantum Fourier transform over $\mathbb{F}_q$ is known to be implementable by a polynomial-size quantum circuit.

[3] It may not be immediately clear from the definition below that $\mathrm{Tr}(x) \in \mathbb{F}_p$, but this is a well-known fact.

For a function $f : \Sigma^n \to \mathbb{C}$, we define

$$\hat{f}(\mathbf{z}) := \frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{x}\in\Sigma^n} f(\mathbf{x})\omega_p^{\mathrm{Tr}(\mathbf{x}\cdot\mathbf{z})}.$$

Then it is easy to see that we have

$$\mathsf{QFT} \sum_{\mathbf{x}\in\Sigma^n} f(\mathbf{x}) |\mathbf{x}\rangle = \sum_{\mathbf{z}\in\Sigma^n} \hat{f}(\mathbf{z}) |\mathbf{z}\rangle .$$

For functions $f : \Sigma^n \to \mathbb{C}$ and $g : \Sigma^n \to \mathbb{C}$, $f \cdot g$ and $f * g$ denote the point-wise product and convolution of f and g, respectively, i.e.,

$$(f \cdot g)(\mathbf{x}) := f(\mathbf{x}) \cdot g(\mathbf{x})$$
$$(f * g)(\mathbf{x}) := \sum_{\mathbf{y}\in\Sigma^n} f(\mathbf{y}) \cdot g(\mathbf{x} - \mathbf{y}).$$

We have the following standard lemmas. We include the proofs for completeness.

Lemma 2.1 (Parceval's equality)**.** *For any $f : \Sigma^n \to \mathbb{C}$, we have*

$$\sum_{\mathbf{x}\in\Sigma^n} |f(\mathbf{x})|^2 = \sum_{\mathbf{z}\in\Sigma^n} |\hat{f}(\mathbf{z})|^2. \tag{2}$$

Proof. This immediately follows from the fact that QFT is a unitary, which is shown in [dBCW02, vDHI06]. □

Lemma 2.2. *Let m be a positive integer that divides n. Suppose that we have $f_i : \Sigma \to \mathbb{C}$ for $i \in [n]$ and $f : \Sigma^n \to \mathbb{C}$ is defined by*

$$f(\mathbf{x}) := \prod_{i\in[n]} f_i(\mathbf{x}_i) \tag{3}$$

where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$. Then, we have

$$\hat{f}(\mathbf{z}) = \prod_{i\in[n} \hat{f}_i(\mathbf{z}_i)$$

where $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n)$.

Proof. This can be proven by the following equalities:

$$\begin{aligned}
\hat{f}(\mathbf{z}) &= \frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{x}\in\Sigma^n} f(\mathbf{x})\omega_p^{\mathrm{Tr}(\mathbf{x}\cdot\mathbf{z})} \\
&= \frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{x}_1\in\Sigma} \cdots \sum_{\mathbf{x}_n\in\Sigma} \prod_{i\in[n]} f_i(\mathbf{x}_i)\omega_p^{\mathrm{Tr}(\mathbf{x}_i\cdot\mathbf{z}_i)} \\
&= \prod_{i\in[n]} \frac{1}{|\Sigma|^{1/2}} \sum_{\mathbf{x}_i\in\Sigma} f_i(\mathbf{x}_i)\omega_p^{\mathrm{Tr}(\mathbf{x}_i\cdot\mathbf{z}_i)} \\
&= \prod_{i\in[n]} \hat{f}_i(\mathbf{z}_i)
\end{aligned}$$

where the second equality follows from Equation (3) and the linearity of Tr. □

Lemma 2.3 (Convolution theorem). *For functions $f : \Sigma^n \to \mathbb{C}$, $g : \Sigma^n \to \mathbb{C}$, and $h : \Sigma^n \to \mathbb{C}$, the following equations hold.*

$$\widehat{f \cdot g} = \frac{1}{|\Sigma|^{n/2}}(\hat{f} * \hat{g}), \tag{4}$$

$$\widehat{f * g} = |\Sigma|^{n/2}(\hat{f} \cdot \hat{g}), \tag{5}$$

$$\widehat{f \cdot (g * h)} = (\hat{f} * (\hat{g} \cdot \hat{h})). \tag{6}$$

Proof. For any $\mathbf{x} \in \Sigma^n$, we have

$$\begin{aligned}
(\hat{f} * \hat{g})(\mathbf{x}) &= \sum_{\mathbf{y} \in \Sigma^n} \hat{f}(\mathbf{y})\hat{g}(\mathbf{x} - \mathbf{y}) \\
&= \sum_{\mathbf{y} \in \Sigma^n} \left(\frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{z} \in \Sigma^n} f(\mathbf{z}) \omega_p^{\mathrm{Tr}(\mathbf{y} \cdot \mathbf{z})} \right) \left(\frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{z}' \in \Sigma^n} g(\mathbf{z}') \omega_p^{\mathrm{Tr}((\mathbf{x} - \mathbf{y}) \cdot \mathbf{z}')} \right) \\
&= \frac{1}{|\Sigma|^n} \sum_{\mathbf{y} \in \Sigma^n} \sum_{\mathbf{z} \in \Sigma^n} \sum_{\mathbf{z}' \in \Sigma^n} f(\mathbf{z}) g(\mathbf{z}') \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{z}')} \omega_p^{\mathrm{Tr}(\mathbf{y} \cdot (\mathbf{z} - \mathbf{z}'))} \\
&= \frac{1}{|\Sigma|^n} \sum_{\mathbf{y} \in \Sigma^n} \sum_{\mathbf{z} \in \Sigma^n} f(\mathbf{z}) g(\mathbf{z}) \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{z})} \\
&= \sum_{\mathbf{z} \in \Sigma^n} f(\mathbf{z}) g(\mathbf{z}) \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{z})} \\
&= |\Sigma|^{n/2} \widehat{(f \cdot g)}(\mathbf{x})
\end{aligned}$$

where the third equality follows from the linearity of Tr and the fourth equality follows from Equation (1). This implies Equation (4).

For any $\mathbf{x} \in \Sigma^n$, we have

$$\begin{aligned}
\widehat{(f * g)}(\mathbf{x}) &= \frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{z} \in \Sigma^n} (f * g)(\mathbf{z}) \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{z})} \\
&= \frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{z} \in \Sigma^n} \sum_{\mathbf{y} \in \Sigma^n} f(\mathbf{y}) g(\mathbf{z} - \mathbf{y}) \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{y})} \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot (\mathbf{z} - \mathbf{y}))} \\
&= \frac{1}{|\Sigma|^{n/2}} \left(\sum_{\mathbf{y} \in \Sigma^n} f(\mathbf{y}) \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{y})} \right) \left(\sum_{\mathbf{z}' \in \Sigma^n} g(\mathbf{z}') \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{z}')} \right) \\
&= |\Sigma|^{n/2} (\hat{f} \cdot \hat{g})(\mathbf{x})
\end{aligned}$$

where the second equality follows from the linearity of Tr. This implies Equation (5). Equation (6) immediately follows from Equations (4) and (5). □

2.3 Chernoff Bound

We rely on the following form of Chernoff bound.

Lemma 2.4 (Chernoff Bound). *Let $X_1, ..., X_n$ be independent random variables taking values in $\{0, 1\}$, $X := \sum_{i \in [n]} X_i$, and $\mu := \mathbb{E}[X]$. For any $\delta \geq 0$, it holds that*

$$\Pr[X \geq (1 + \delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2 + \delta}}.$$

3 Cryptographic Definitions in the Random Oracle Model

Here, we define various cryptographic notions we will be constructing. There is some subtlety in defining these notions relative to random oracles, which we discuss here.

We associate the set of oracles $H : \{0,1\}^* \to \{0,1\}$ with the set of all infinite-length bit-strings in the natural way: each input $x \in \{0,1\}^*$ is bijectively mapped to an integer i, and the output of H is the ith bit of the infinite string. Oracles with multi-bit outputs can be analogously associated with infinite-length strings as well. Then a random oracle is the oracle associated with a random infinite-length bit-string.

We now fix the computational models we will be considering. We consider three settings:

- **Oracle-independent.** Here, a query algorithm $\mathcal{A}$ is fixed, and then a random oracle H is chosen. In this case, we always allow $\mathcal{A}$ unbounded computation, but require the number of queries $\mathcal{A}$ makes to H to be polynomial in its input length. We consider such $\mathcal{A}$ "efficient." We distinguish between $\mathcal{A}$ that can make classical queries and $\mathcal{A}$ that can make quantum queries. We say such $\mathcal{A}$ are efficient oracle-independent adversaries in the CROM or QROM, respectively.

- **Uniform oracle-dependent.** Here, the random oracle H is chosen, and then finite-length string a is chosen based on H. $\mathcal{A}$ then gets a as advice. Like before, efficient $\mathcal{A}$ are taken to be those that are potentially computational unbounded but make only a polynomial (in their input length) number of queries to H.

- **Non-uniform oracle-dependent.** Here, the random oracle H is chosen, and then for each n, a string a_n of length polynomial in n is determined based on H. $\mathcal{A}$, on input of length n, is additionally given a_n as advice. Like before, efficient $\mathcal{A}$ are taken to be those that are potentially computational unbounded but make only a polynomial (in their input length) number of queries to H.

Remark 3. *Note that the cryptographic literature typically (but not always) considers the oracle-independent setting. On the other hand, complexity-theoretic results are often phrased in terms of the oracle-dependent models. The oracle-dependent models are meant to capture the standard model as closely as possible, where H is replaced with a fixed hash function. In such a setting, the adversary is designed potentially with H in mind, and so is chosen after H. Modeling in this way captures trivial standard model impossibilities, such as the impossibility of keyless collision resistant hash functions against non-uniform adversaries, which are not captured by the oracle-independent model.*

Definition 3.1 (Family of oracle-aided functions.)**.** *For functions $\ell_{\mathsf{key}} = \ell_{\mathsf{key}}(\lambda), \ell_{\mathsf{in}} = \ell_{\mathsf{in}}(\lambda), \ell_{\mathsf{out}} = \ell_{\mathsf{out}}(\lambda)$, a family $\{f_\lambda : \{0,1\}^{\ell_{\mathsf{key}}} \times \{0,1\}^{\ell_{\mathsf{in}}} \to \{0,1\}^{\ell_{\mathsf{out}}}\}_{\lambda\in\mathbb{N}}$ of* efficiently computable oracle-aided keyed functions relative to oracles *H is a family of functions f_λ that is implemented by a polynomial-time (deterministic) classical machine with an oracle access to H. The family of functions is* keyless *if $\ell_{\mathsf{key}} = 0$. If we do not specify keyed or keyless, we mean keyless. We denote by f_λ^H to mean f_λ relative to a specific oracle H.*

One-way functions. We now define what it means for an oracle-aided function to be one-way relative to a random oracle. We provide several definitions, capturing classical vs quantum computation, uniform vs non-uniform computation, and the order of quantitifiers between the choice of oracle and choice of adversary.

For one-way functions, we only consider keyless functions, as it is well known that keyless and keyed one-way functions are equivalent.

Definition 3.2 (One-way functions with random oracles)**.** *We say that a family* $\{f_\lambda : \{0,1\}^{\ell_{\mathsf{in}}} \to \{0,1\}^{\ell_{\mathsf{out}}}\}_{\lambda\in\mathbb{N}}$ *of efficiently computable oracle-aided functions relative to oracles* H *is one-way against oracle-independent adversaries in the CROM (resp. QROM) if for all unbounded oracle-independent* $\mathcal{A}$ *that make* $\mathsf{poly}(\lambda)$ *classical (resp. quantum) queries to* H*, there exists a negligible function* negl *such that:*

$$\Pr_H[y = f_\lambda^H(x') : x \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{in}}}, y = f_\lambda^H(x), x' \xleftarrow{\$} \mathcal{A}^H(1^\lambda, y)] < \mathsf{negl}(\lambda). \tag{7}$$

We say that $\{f_\lambda\}_\lambda$ *is one-way against uniform (resp. non-uniform) adversaries (in the CROM or QROM) if, with probability 1 over the choice of* H*, it holds that for all oracle-dependent uniform (resp. non-uniform) adversaries* $\mathcal{A}$*, there exists a negligible* negl *such that inequality (7) holds.*

Collision-resistance. We now define collision-resistant hashing. Paralleling the case of one-way functions, we provide several definitions.

Definition 3.3 (Collision-resistance with random oracles)**.** *We say that a family* $\{f_\lambda : \{0,1\}^{\ell_{\mathsf{key}}} \times \{0,1\}^{\ell_{\mathsf{in}}} \to \{0,1\}^{\ell_{\mathsf{out}}}\}_{\lambda\in\mathbb{N}}$ *of efficiently computable oracle-aided* keyed *functions relative to oracles* H *is collision-resistant against oracle-independent adversaries in the CROM (resp. QROM) if for all unbounded-time uniform adversaries* $\mathcal{A}$ *that make* $\mathsf{poly}(\lambda)$ *classical (resp. quantum) queries to* H*, there exists a negligible function* negl *such that:*

$$\Pr_H[f_\lambda^H(k,x_0) = f_\lambda^H(k,x_1) \ \wedge \ x_0 \neq x_1 : (x_0,x_1) \xleftarrow{\$} \mathcal{A}^H(k), k \leftarrow \{0,1\}^{\ell_{\mathsf{key}}}] = \mathsf{negl}(\lambda).$$

(Uniform and non-uniform) oracle-dependent collision-resistance is defined analogously.

A *keyless* hash function has $\ell_{\mathsf{key}} = 0$. Note that unlike one-way functions, *keyless* collision resistant hash functions cannot have security against non-uniform oracle-dependent adversaries, since a non-uniform adversary can have collisions for every security parameter hard-coded.

Proofs of quantumness. We now define proofs of quantumness, which have a quantum prover prove that they are quantum to a classical verifier. Like before, we will consider various definitions.

Definition 3.4. *A (keyed non-interactive publicly verifiable) proof of quantumness in the QROM consists of algorithms* $(\mathsf{Prove}, \mathsf{Verify})$*.*

$\mathsf{Prove}^H(k)$**:** *This is a QPT algorithm that takes a key* $k \in \{0,1\}^{\ell_{\mathsf{key}}}$ *as input, makes* $\mathsf{poly}(\lambda)$ *quantum queries to the random oracle* H*, and outputs a classical proof* π*.*

$\mathsf{Verify}^H(k,\pi)$**:** *This is a deterministic classical polynomial-time algorithm that takes* k *and a proof* π*, makes* $\mathsf{poly}(\lambda)$ *queries to the random oracle* H*, and outputs* $\top$ *indicating acceptance or* $\bot$ *indicating rejection.*

We require a proof of quantumness to satisfy the following properties.

Correctness. *We have*

$$\Pr_{H,k}\left[\mathsf{Verify}^H(k,\pi) = \bot : \ \pi \xleftarrow{\$} \mathsf{Prove}^H(k) \ \right] \leq \mathsf{negl}(\lambda).$$

Soundness. *A proof of quantumness is $(Q(\lambda), \epsilon(\lambda))$-sound against oracle-independent adversaries if, for any unbounded-time oracle-independent adversary $\mathcal{A}$ that makes $Q(\lambda)$* classical *oracle queries to H, we have*

$$\Pr_{H,k}\left[\mathsf{Verify}^H(k,\pi^*)=\top:\ \ \pi^*\xleftarrow{\$}\mathcal{A}^H(k)\ \right]\leq\epsilon(\lambda).$$

When we do not specify Q and ϵ, we require the proof of quantumness to satisfy $(Q(\lambda),\mathsf{negl}(\lambda))$-soundness for all polynomials Q. (Uniform and non-uniform) oracle-dependent soundness is defined analogously.

A keyless *proof of quantumness has $\ell_{\mathsf{key}}=0$, in which case all algorithms additionally take 1^λ as input.*

Note that, as with collision resistance, there cannot be keyless proofs of quantumness with soundness against non-uniform oracle-dependent adversaries. Indeed, a non-uniform adversary could have a proof π hardcoded for every input length.

Proofs of randomness. We now define proofs of (min-)entropy and proofs of randomness, also referred to as certifiable randomness. These are protocols by which a classical verifier with very little entropy can produce significant entropy with the help of a potentially untrusted quantum device.

We note that Brakerski et al.'s [BCM+18] work giving the first certifiable randomness protocol relative to a single device actually did not provide a formal definition. The work of Amos et al. [AGKZ20] provide a definition of certifiable min-entropy, but we observe that it is technically unsatisfiable. Their definition says that, conditioned on the verifier accepting, the string produced by the verifier must have min-entropy. We note, however, that a malicious device may always output a deterministic value. This value may be accepted with negligible but non-zero probability. Conditioned on accepting, the entropy remains zero. We give new definitions for certifiable entropy and randomness, overcoming this limitation.

We also note that defining certifiable randomness relative to a random oracle is subtle, since the random oracle itself is an infinite source of randomness. To accurately model entropy that comes from the protocol as opposed to the random oracle, we insist that the random string produced by the verifier has min-entropy or is uniformly random, even conditioned on the random oracle.

We note that for a proof of min-entropy, the situation is analogous to collision resistance where it is potentially feasible in the uniform setting or with a key, but trivially impossible in the oracle-dependent non-uniform keyless setting. However, for a proof of randomness, it is inherent in the non-interactive setting that the verifier must have some local randomness. This is because, in the non-interactive setting without verifier randomness, a malicious prover can keep generating samples until, say, the first bit of the output is 0. Such a string clearly will not be uniformly random. This shows that the actual string obtained by the verifier must be kept secret from the prover, at least until after the prover's message is sent.

We now give the definitions.

Definition 3.5. *A (keyed non-interactive publicly verifiable) proof of min-entropy in the QROM consists of algorithms* $(\mathsf{Prove},\mathsf{Verify})$.

$\mathsf{Prove}^H(k,h)$**:** *This is a QPT algorithm that takes a key $k\in\{0,1\}^{\ell_{\mathsf{key}}}$ as input, as well as a min-entropy threshold h. It makes $\mathsf{poly}(\lambda)$ quantum queries to the random oracle H, and outputs a classical proof π.*

$\mathsf{Verify}^H(k, h, \pi; r)$**:** *This is a potentially randomized polynomial-time algorithm that takes k, h, a proof π, and random coins r; it makes $\mathsf{poly}(\lambda)$ queries to the random oracle H, and outputs either a string x, or $\perp$ indicating rejection.*

We require a proof of min-entropy to satisfy the following properties:

Correctness. *This is identical to the correctness requirement of a proof of quantumness. The one stipulation we make is that the length of the randomness r used by* Verify *should be* $\mathsf{poly}(\lambda)$ *bits, independent of h.*

Min-entropy. *A proof of min-entropy has uniform (resp. non-uniform) min-entropy against oracle-independent adversaries if, for any polynomial $h = h(\lambda)$, any unbounded adversary $\mathcal{A}$ that makes a polynomial number of quantum oracle queries to H, and for any inverse polynomial δ, there is a negligible* negl *such that the following holds. Let $\mathcal{A}^H_\top(k, h; r)$ be the distribution $\mathsf{Verify}^H(k, h, \mathcal{A}^H(k, h); r)$, conditioned on the output not being $\perp$. Then:*

$$\Pr_{k,H}\left[\Pr[\mathsf{Verify}^H(k, h, \mathcal{A}^H(k, h); r) \neq \perp] \geq \delta(\lambda) \wedge H_\infty\left(\mathcal{A}^H_\top(k, h; r) \mid k, H, r\right) \leq h(\lambda)\right] \leq \mathsf{negl}(\lambda)$$

Uniform (resp. non-uniform) oracle-dependent min-entropy is defined as follows: with probability 1 over the choice of H, for any unbounded uniform (resp. non-uniform) oracle-dependent adversary $\mathcal{A}$ that makes a polynomial number of quantum oracle queries to H and any inverse polynomial δ, there is a negligible negl *such that the following holds:*

$$\Pr_{k}\left[\Pr[\mathsf{Verify}^H(k, h, \mathcal{A}^H(k, h); r) \neq \perp] \geq \delta(\lambda) \wedge H_\infty\left(\mathcal{A}^H_\top(k, h; r) \mid k, H, r\right) \leq h(\lambda)\right] \leq \mathsf{negl}(\lambda)$$

A keyless *proof of min-entropy has $\ell_{\mathsf{key}} = 0$, in which case all algorithms additionally take 1^λ as input.*

Note that min-entropy and correctness together imply that the output of Verify when interacting with the honest Prove algorithm must have min-entropy at least h.

A proof of *randomness* has the same syntax as a proof of min-entropy, except that we require the output of $\mathsf{Verify}^H(k, h, \pi; r)$ to be exactly h bits. However, we upgrade the min-entropy requirements to the following:

True randomness. A proof of randomness has true randomness if, for any polynomial $h = h(\lambda)$ and any unbounded adversary $\mathcal{A}$ that makes a polynomial number of quantum oracle queries to H, and for any inverse polynomial δ, there is a negligible negl such that the following holds. If for a given k, H it holds that $\Pr[\mathsf{Verify}^H(k, h, \mathcal{A}^H(k, h); r) \neq \perp] \geq \delta$, then

$$\Delta\left(\ (r, U)\ ,\ (r, \mathcal{A}^H_\top(k, h; r))\ )\right) \leq \mathsf{negl}(\lambda)$$

Here, Δ is statistical distance and U is the uniform distribution over h-bit-strings. (Uniform and non-uniform) oracle-dependent true randomness is defined analogously.

In other words, provided that Verify actually outputs a string with inverse polynomial probability, that string will be statistically close to random.

3.1 Relations between Variants

We now discuss relations between variants of the various cryptographic definitions above. For one-wayness and collision resistance, QROM security implies CROM security. Clearly, non-uniform

oracle-dependent security implies uniform oracle-dependent security, which in turn implies oracle-independent security. In the following, we discuss various results in the other direction. These largely follow in a straightforward way from standard arguments, but we include them here for completeness.

The first theorem shows uniform oracle-dependent security follows from oracle-independent security.

Theorem 3.6. *If $\{f_\lambda\}_\lambda$ is one-way against oracle-independent adversaries in the CROM (resp. QROM), then it is also one-way against uniform oracle-dependent adversaries in the CROM (resp. QROM). Analogous statements hold for collision resistance, proofs of quantumness, proofs of min-entropy and proof of randomness.*

Proof. In all cases, we first fix the advice a. Security in the oracle-independent setting implies that the set of random oracles for which the adversary breaks security has measure 0. Unioning over all countably many a results in the measure being 0. This then gives uniform oracle-dependent security. □

In the non-uniform case, the above fails. This is because the advice for $\mathcal{A}$ is now an infinite-length string $a_1, a_2, \ldots$, and therefore the advice strings are uncountable. This, moreover, is inherent, with keyless collision resistance giving an example that is uniformly secure but not non-uniformly secure. Moreover, in the case of one-way functions and proofs of quantumness, it is not hard to come up with counter-example constructions that are uniformly secure, but not non-uniformly secure.

However, by thinking of a_n as being a polynomial amount of advice about the oracle, we can use known results in the pre-processing setting to lift from oracle-independent to non-uniform security. Concretely, [CDGS18] and [CGLQ20] show that salting generically defeats pre-processing in the classical and quantum random oracle models, respectively. Note that the results require it to be efficiently verifiable when the adversary wins; this applies to one-way functions, collision resistance, and proofs of quantumness, but not to proofs of min-entropy/randomness, where it cannot be efficiently checked if the adversary produced a low entropy or non-uniform string.

Re-interpreting, the pre-processing results show that salting generically lifts oracle-independent to non-uniform security. This salt can be interpreted as a key. In the case of one-way functions, this salt can be thought of as another part of the input. We thus obtain the following as immediate corollaries:

Theorem 3.7. *If $\{f_\lambda\}_\lambda$ is one-way against oracle-independent adversaries in the CROM (resp. QROM), then $\{g_\lambda\}_\lambda$ where $g_\lambda^H(s,x) = s||f_\lambda^{H(s||\cdot)}(s,x)$ and where $s \in \{0,1\}^\lambda$ is one-way against non-uniform oracle-dependent adversaries in the CROM (resp. QROM).*

Theorem 3.8. *If $\{f_\lambda\}_\lambda$ is a potentially keyed function family that collision resistant against oracle-independent adversaries in the CROM (resp. QROM), then the* keyed *function $\{g_\lambda\}_\lambda$ where $g_\lambda(k_0||k_1,x) = f_\lambda^{H(k_1||\cdot)}(k_0,x)$ and where $k_1 \in \{0,1\}^\lambda$ is collision resistant against non-uniform oracle-dependent adversaries in the CROM (resp. QROM). Analogous statements hold for proofs of quantumness.*

We next discuss how salting actually does lift security for proofs of min-entropy and randomness from the uniform to non-uniform case. We note that [CGLQ20] actually *does* work, by fixing a particular string, and having the adversary win if it can cause the verifier to output that string. This event occurs with exponentially-small probability, but [CGLQ20] would handle exponentially small probabilities by setting the salt to be appropriately larger than the min-entropy requirement.

This limits the utility of a proof of min-entropy, since the large salt could have just been used as the source of randomness. In the following, we show that small salts can, in fact, be used, though it requires a more careful proof and cannot simply rely on the prior theorem statements.

Theorem 3.9. *If* $(\mathsf{Prove}_0, \mathsf{Verify}_0)$ *has min-entropy (resp. true randomness) against oracle-independent adversaries in the QROM, then* $(\mathsf{Prove}, \mathsf{Verify})$ *has min-entropy (resp. true randomness) against non-uniform oracle-dependent adversaries in the QROM, where* $\mathsf{Prove}^H(\ k_0||k_1, h\ ) = \mathsf{Prove}_0^{H(k_1||\cdot)}(k_0, h+1)$ *and* $\mathsf{Verify}^H(k_0||k_1, h, \pi) = \mathsf{Verify}_0^{H(k_1||\cdot)}(k_0, h+1, \pi)$ *and where* $k_1 \in \{0,1\}^\lambda$.

Proof. We prove the min-entropy case, the true randomness case being essentially identical. Consider a non-uniform oracle-dependent adversary $\mathcal{A}$ for the min-entropy of $(\mathsf{Prove}, \mathsf{Verify})$.

Suppose $\mathcal{A}$ breaks min-entropy. This means there is a polynomial h, an inverse polynomial δ, and a non-negligible ϵ such that, the following simulteneously hold with probability at least ϵ over the choice of H_1, k_0, k_1:

$$\Pr[\mathsf{Verify}^{H_1}(k_0||k_1, h+1, \mathcal{A}^{H_1}(a_n, k_0||k_1, h+1)\ ;\ r) \neq \perp] \geq \delta(\lambda) \tag{8}$$

$$H_\infty\left(\mathcal{A}_\top^{H_1}(a_n, k_0||k_1, h+1\ ;\ r) \mid a_n, k_0, k_1, H_1, r\right) \leq h+1 \tag{9}$$

Above, $a_n = a_n(H_1)$ is the advice $\mathcal{A}$ is provided for oracle H_1, where $n = |k_0| + \lambda$, the length of the input $k_0||k_1$ to $\mathcal{A}$.

Consider choosing a random set $S \subseteq \{0,1\}^\lambda \backslash \{k_1\}$ of size $\ell - 1$, for an ℓ to be chosen momentarily. We now consider a modified oracle

$$H_1'(s,x) = \begin{cases} 0 & \text{if } s \in S \\ H_1(s,x) & \text{otherwise} \end{cases}$$

Lemma 3.10. *For any* ℓ *such that* $\ell/2^\lambda$ *is negligible, there exists a negligible function* negl *such that, with overwhelming probability over the choice of* S*, Equations 8 and 9 hold when making the following replacements:* $H_1 \mapsto H_1'$*,* $\delta \mapsto \delta' = \delta - \mathsf{negl}$*,* $\epsilon \mapsto \epsilon' = \epsilon - \mathsf{negl}$*, and* $h+1 \mapsto h$*.*

Proof. This is a now-standard quantum query complexity argument. Consider the state $|\phi_t\rangle = \sum \alpha_{x,y}|x,y\rangle$ of a quantum query algorithm when it makes its t-th quantum query. Define $q_x(|\phi_t\rangle)$ to be the magnitude squared of x in the superposition of query t, that is $q_x(|\phi_t\rangle) = \sum_y |\alpha_{x,y}|^2$. Call this the query magnitude of x. Let $q_x = \sum_t q_x(|\phi_t\rangle)$ be the total query magnitude of x. For a set S, let $q_S = \sum_{x \in S} q_x$ be the total query magnitude of S.

Since S is random but also a negligible fraction of all inputs, q_S is negligible with overwhelming probability. H_1 and H_1' only differ on points in S. We will now use the following lemmas to argue that replacing H_1 with H_1' negligibly affects the output distribution of $\mathcal{A}$:

Lemma 3.11 ([BBBV97] Theorem 3.1)**.** *Let* $|\phi\rangle \approx_\epsilon |\psi\rangle$*, performing any measurement measurement on* $|\phi\rangle$ *and* $|\psi\rangle$ *yields distributions with statistical distance at most* 4ϵ*.*

Lemma 3.12 ([BBBV97] Theorem 3.3)**.** *Let* $\mathcal{A}$ *be a quantum query algorithm making* T *queries to an oracle* O*. Let* $\epsilon > 0$ *and let* S *be a set such that* $q_S \leq \epsilon^2/T$*. Let* O' *be another oracle that is identical to* O *on all points not in* S*. Let* $|\phi\rangle, |\psi\rangle$ *be the final state of* A *when given* O, O'*, respectively. Then* $|\phi\rangle \approx_\epsilon |\psi\rangle$

Therefore, conditioned on the query amplitude being negligible, the output distribution of $\mathcal{A}$ is negligibly affected. □

Lemma 3.13. *Assume ℓ is super-polynomial. Then with overwhelming probability over the choice of S, k_1, the following holds: conditioned on a_n, the function H'_1 is statistically close to H''_1 defined as:*

$$H''_1(s,x) = \begin{cases} 0 & \text{if } s \in S \\ H(x) & \text{if } s = k_1 \\ H_1(s,x) & \text{otherwise} \end{cases}$$

where H is an independent random oracle.

Proof. For a random S, k_1, we can equivalently sample H'_1 as follows: first choose a random set $T \subseteq \{0,1\}^\lambda$ of size ℓ, and then set k_1 to be a random element of T and $S = T \setminus \{k_1\}$.

Fix T. If we do not condition on a_n, we know that the truth tables of $H_1(s,\cdot)$ for each $s \in T$ are uniform and independent of each other as well as independent of $H_1(s,\cdot)$ for $s \notin T$. When we condition on a_n, the entire set of $|T|$ truth tables only loses $|a_n|$ bits of entropy, a polynomial. By sub-additivity, conditioning on a_n only reduces the average entropy (over s) of the $H_1(s,\cdot)$ by $|a_n|/|T| = \mathsf{negl}$. This means that an overwhelming fraction of the $H_1(s,\cdot)$ have entropy reduced by a negligible amount, and are therefore statistically close to uniform even conditioned on a_n, and even conditioned on $H_1(s,\cdot)$ for $s \notin T$. Thus for a random $s \in T$, we can replace $H_1(s,\cdot)$ with a random and independent $H(\cdot)$ and only negligibly affect the distribution. □

An immediate consequence is the following:

Corollary 3.14. *Assume ℓ is such that (1) ℓ is superpolynomial, and (2) $\ell/2^\lambda$ is negligible. Then there exists a negligible function* negl *such that, with overwhelming probability over the choice of S, Equations 8 and 9 hold when making the following replacements: $H_1 \mapsto H''_1$, $\delta \mapsto \delta' = \delta - \mathsf{negl}$, $\epsilon \mapsto \epsilon' = \epsilon - \mathsf{negl}$, and $h+1 \mapsto h$.*

Now we construct an oracle-independent adversary $\mathcal{B}$ for the min-entropy of $(\mathsf{Prove}_0, \mathsf{Verify}_0)$ for parameter h. $\mathcal{B}(k_0)$, which has access to random oracle H, chooses its own oracle H_1 as above to satisfy Equations 8 and 9. Then it computes the advice a_n that $\mathcal{A}$ would get if given H_1. It chooses a salt $k_1 \xleftarrow{\$} \{0,1\}^\lambda$ and set S, and it defines H''_1 as above. Note that $\mathcal{B}$ can simulate a (quantum) query to H''_1 using two quantum queries to H: one to compute the output, and another to un-compute any scratch space needed to answer the query. $\mathcal{B}$ now runs $\mathcal{A}^{H''_1}(k_0||k_1)$, and outputs whatever $\mathcal{A}$ outputs.

If $\mathcal{B}$ were to choose S, k_1 uniformly, then $\mathcal{B}$ perfectly simulates the view of $\mathcal{A}$ as in Corollary 3.14. This means there must *exist* some S, k_1 such that if $\mathcal{B}$ runs $\mathcal{A}$ using this S, k_1, with probability at least $\epsilon(\lambda) - \mathsf{negl}(\lambda)$ over the choice of k_0, H,

$$\Pr[\mathsf{Verify}_0^H(k_0, h, \mathcal{B}^H(k_0), h; r) \neq \bot] \geq \delta(\lambda) - \mathsf{negl}(\lambda)$$
$$H_\infty\left(\mathcal{B}_\top^H(k_0, h; r) \mid k_0, H, r\right) \leq h(\lambda) + \mathsf{negl}(\lambda)$$

$\mathcal{B}$ therefore breaks the oracle-independent security of $\mathsf{Prove}_0, \mathsf{Verify}_0$. □

From min-entropy to true randomness. Here we discuss how proofs of min-entropy imply proofs of true randomness. This is an immediate application of extractors:

Theorem 3.15. *If proofs of min-entropy against oracle-independent adversaries in the QROM exist, then so to proofs of true randomness. If the proof of min-entropy is secure against uniform or non-uniform oracle-dependent adversaries, then so is the proof of randomness. If the proof of min-entropy is keyless, then so is the proof of randomness.*

Proof. We simply have a new Verify' which chooses a random seed for a strong extractor, which it applies to the result of Verify, outputting whatever the extractor outputs. By choosing the min-entropy h sufficiently higher than the desired output length according to the parameters of the extractor, the output of Verify' will be statistically close to random and the desired length. □

We note that the verifier's random seed for the extractor can be sampled after the prover's message, and can also be made public afterward. The result is that if the proof of min-entropy is public coin and publicly verifiable, the proof of randomness will be as well, at the cost of a single final message from the verifier.

4 Error Correcting Codes.

In this section, we first review basic definitions and facts on error correcting codes. Then, we state requirements of codes that are needed for our purpose. Then, we show that such a code exists based on known results.

4.1 Definitions

A code of length $n \in \mathbb{N}$ over an alphabet Σ (which is a finite set) is a subset $C \subseteq \Sigma^n$.

Linear codes. A code C is said to be a linear code if its alphabet is $\Sigma = \mathbb{F}_q$ for some prime power q and $C \subseteq \mathbb{F}_q^n$ is a linear subspace of $\mathbb{F}_q^n$. We call the dimension of C as a linear subspace the rank of C.

Folded linear codes. A code C is said to be a folded linear code [Kra03, GR08] if its alphabet is $\Sigma = \mathbb{F}_q^m$ for some prime power q and a positive integer m and $C \subseteq \Sigma^n$ is a linear subspace of $\mathbb{F}_q^{nm}$ where n is the length of C and we embed C into $\mathbb{F}_q^{nm}$ in the canonical way. Linear codes are the special case of folded linear codes where $m = 1$. For a linear code $C \subseteq \mathbb{F}_q^n$ and a positive integer m that divides n, we define its m-folded version $C^{(m)}$ as follows:

$$C^{(m)} := \{((x_1, \ldots, x_m), (x_{m+1}, \ldots, x_{2m}) \ldots, (x_{n-m+1}, \ldots, x_n)) : (x_1, \ldots, x_n) \in C\}.$$

Clearly, $C^{(m)}$ is a folded linear code. Conversely, any folded linear code can be written as $C^{(m)}$ for some linear code C and a positive integer m.

Dual codes. Let C be a linear code of length n and rank k over $\mathbb{F}_q$. The *dual code* $C^\perp$ of C is defined as the orthogonal complement of C as a linear space over $\mathbb{F}_q$, i.e.,

$$C^\perp := \{\mathbf{z} \in \mathbb{F}_q^n : \mathbf{x} \cdot \mathbf{z} = 0 \text{ for all } \mathbf{x} \in C\}.$$

$C^\perp$ is a linear code of length n and rank $n - k$ over $\mathbb{F}_q$.[4]

We define dual codes for folded linear codes similarly. That is, for a folded linear code $C \in \Sigma^n$ over the alphabet $\Sigma = \mathbb{F}_q^m$, its dual $C^\perp$ is defined as

$$C^\perp := \{\mathbf{z} \in \Sigma^n : \mathbf{x} \cdot \mathbf{z} = 0 \text{ for all } \mathbf{x} \in C\}.$$

It is clear from the definition that $(C^\perp)^{(m)} = (C^{(m)})^\perp$ for any linear codes C of length n and positive integer m that divides n.

[4]Note that it does not always hold that $\mathbb{F}_q^n = C \oplus C^\perp$ since the bilinear form $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{x} \cdot \mathbf{y}$ does not satisfy the axioms of the inner product (i.e., there may exist $\mathbf{x} \neq 0$ such that $\mathbf{x} \cdot \mathbf{x} = 0$).

Lemma 4.1. *For a folded linear code $C \subseteq \Sigma^n$, if we define*

$$f(\mathbf{x}) := \begin{cases} \frac{1}{\sqrt{|C|}} & \mathbf{x} \in C \\ 0 & \textit{otherwise} \end{cases},$$

then we have

$$\hat{f}(\mathbf{z}) = \begin{cases} \frac{1}{\sqrt{|C^\perp|}} & \mathbf{z} \in C^\perp \\ 0 & \textit{otherwise} \end{cases}.$$

Proof. For $\mathbf{z} \in C^\perp$, we have

$$\begin{aligned} \hat{f}(\mathbf{z}) &= \frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{x} \in \Sigma^n} f(\mathbf{x}) \omega_p^{\mathrm{Tr}(\mathbf{x} \cdot \mathbf{z})} \\ &= \frac{1}{|\Sigma|^{n/2}} \sum_{\mathbf{x} \in C} \frac{1}{\sqrt{|C|}} \\ &= \frac{1}{\sqrt{|C^\perp|}} \end{aligned}$$

where the final equality follows from $|C| \cdot |C^\perp| = |\Sigma|^n$. $\hat{f}(\mathbf{z}) = 0$ for $\mathbf{z} \notin C^\perp$ immediately follows from the above and Lemma 2.1. □

List recovery. We say that a code $C \subseteq \Sigma^n$ is $(\zeta, \ell.L)$-list recoverable if for any subsets $S_i \subseteq \Sigma$ such that $|S_i| \leq \ell$ for $i \in [n]$, we have

$$|\{(x_1, ..., x_n) \in C : |\{i \in [n] : x_i \in S_i\}| \geq (1 - \zeta)n\}| \leq L.$$

Note that list recoverability in the literature usually requires that the list of all codewords $(x_1, ..., x_n) \in C$ satisfying $|\{i \in [n] : x_i \in S_i\}| \geq (1 - \zeta)n$ can be computed from $\{S_i\}_{i \in [n]}$ in time polynomial in $|\Sigma|, n, \ell$. However, we will not require this.

4.2 Suitable Codes

The following lemma claims the existence of codes that are suitable for our purpose.

Lemma 4.2 (Suitable codes)**.** *For any constants $0 < c < c' < 1$, there is an explicit family $\{C_\lambda\}_{\lambda \in \mathbb{N}}$ of folded linear codes over the alphabet $\Sigma = \mathbb{F}_q^m$ of length n where $|\Sigma| = 2^{\lambda^{\Theta(1)}}$, $n = \Theta(\lambda)$, and $|C_\lambda| \geq 2^{n+\lambda}$ that satisfies the following.*[5]

1. *C_λ is (ζ, ℓ, L)-list recoverable where $\zeta = \Omega(1)$, $\ell = 2^{\lambda^c}$ and $L = 2^{\tilde{O}(\lambda^{c'})}$.*
2. *There is an efficient deterministic decoding algorithm $\mathsf{Decode}_{C^\perp}$ for $C^\perp$ that satisfies the following. Let $\mathcal{D}$ be a distribution over Σ that takes $\mathbf{0}$ with probability $1/2$ and otherwise takes a uniformly random element of $\Sigma \setminus \{\mathbf{0}\}$. Then, it holds that*

$$\Pr_{\mathbf{e} \xleftarrow{\$} \mathcal{D}^n}[\forall \mathbf{x} \in C^\perp,\ \mathsf{Decode}_{C^\perp}(\mathbf{x} + \mathbf{e}) = \mathbf{x}] = 1 - 2^{-\Omega(\lambda)}.$$

[5]Item 3 is not needed for the construction of a proof of quantumness given in Section 6. It is used only in the separation of one-way functions given in Section 7.1.

3. *For all* $j \in [n-1]$, $\Pr_{\mathbf{x} \xleftarrow{\$} C_\lambda}[\mathsf{hw}(\mathbf{x}) = n - j] \leq \left(\frac{n}{|\Sigma|}\right)^j$.

Our instantiation of C_λ is just folded Reed-Solomon codes with an appropriate parameter setting. Item 1 is a direct consequence of the list recoverability of folded Reed-Solomon codes in a certain parameter regime [GR08, Rud07]. For proving Item 2, we first remark that the dual of folded Reed-Solomon codes is folded *generalized* Reed-Solomon codes, which have efficient list decoding algorithm [GS99]. Then, we prove that the list decoding algorithm returns a unique decoding result when the error comes from the distribution $\mathcal{D}^n$. Item 3 follows from a simple combinatorial argument. The proof of Lemma 4.2 is given in Section 4.3.

4.3 Proof of Lemma 4.2

In this subsection, we prove Lemma 4.2, i.e., we give a construction of codes that satisfy the properties stated in Lemma 4.2.

4.3.1 Preparation

Before giving the construction, we need some preparations.

Generalized Reed-Solomon codes. We review the definition and known facts on (generalized) Reed-Solomon codes. See e.g., [Lin10, Section 6] for more details.

A generalized Reed-Solomon code $\mathrm{GRS}_{\mathbb{F}_q,\gamma,k,\mathbf{v}}$ over $\mathbb{F}_q$ w.r.t. a generator γ of $\mathbb{F}_q^*$, the degree parameter $0 \leq k \leq N$, and $\mathbf{v} = (v_1, ..., v_N) \in \mathbb{F}_q^{*N}$ where $N := q - 1$ is defined as follows:

$$\mathrm{GRS}_{\mathbb{F}_q,\gamma,k,\mathbf{v}} := \{(v_1 f(\gamma), v_2 f(\gamma^2) ... v_N f(\gamma^N)) : f \in \mathbb{F}_q[x]_{deg \leq k}\}$$

where $\mathbb{F}_q[x]_{deg \leq k}$ denotes the set of polynomials over $\mathbb{F}_q$ of degree at most k.[6] We remark that $\mathrm{GRS}_{\mathbb{F}_q,\gamma,k,\mathbf{v}}$ is a linear code over $\mathbb{F}_q$ that has length $N = q - 1$ and rank $k + 1$. A Reed-Solomon code is a special case of a generalized Reed-Solomon code where $\mathbf{v} = (1, 1, \ldots, 1)$. We denote it by $\mathrm{RS}_{\mathbb{F}_q,\gamma,k}$ (which is equivalent to $\mathrm{GRS}_{\mathbb{F}_q,\gamma,k,(1,1,...,1)}$). The dual of $\mathrm{RS}_{\mathbb{F}_q,\gamma,k}$ is $\mathrm{GRS}_{\mathbb{F}_q,\gamma,N-k-2,\mathbf{v}}$ for some $\mathbf{v} \in \mathbb{F}_q^N$ [Lin10, Claim 6.3].[7]

There is a classical polynomial-time deterministic list decoding algorithm $\mathrm{GRSListDecode}_{\mathbb{F}_q,\gamma,k,\mathbf{v}}$ for $\mathrm{GRS}_{\mathbb{F}_q,\gamma,k,\mathbf{v}}$ that corrects up to $N - \sqrt{kN}$ errors [GS99].[8] More precisely, for any $\mathbf{z} \in \mathbb{F}_q^N$, $\mathrm{GRSListDecode}_{\mathbb{F}_q,\gamma,k,\mathbf{v}}(\mathbf{z})$ returns the list of all $\mathbf{x} \in \mathrm{GRS}_{\mathbb{F}_q,\gamma,k,\mathbf{v}}$ such that $\mathsf{hw}(\mathbf{x} - \mathbf{z}) < N - \sqrt{kN}$.

Folded Reed-Solomon codes. Let m be a positive integer that divides $N = q-1$. The m-folded version $\mathrm{RS}^{(m)}_{\mathbb{F}_q,\gamma,k}$ of $\mathrm{RS}_{\mathbb{F}_q,\gamma,k}$ is a folded linear code of length $n = N/m$.[9] It is known that $\mathrm{RS}^{(m)}_{\mathbb{F}_q,\gamma,k}$ is list recoverable in the following parameter regime [GR08, Rud07].[10]

[6] Reed-Solomon codes whose length N is smaller than $q - 1$ are often considered. But we focus on the case of $N = q - 1$.

[7] Recall that the rank of (generalized) Reed-Solomon codes is the degree parameter k plus one.

[8] [GS99] described the list decoding algorithm for Reed-Solmon codes, but that can be extended to one for generalized Reed-Solomon codes in a straightforward manner since scalar multiplications in each coordinate do not affect the decodability.

[9] We remark that the roles of n and N are swapped compared with [GR08, Rud07].

[10] The following lemma is based on Rudra's PhD thesis [Rud07]. The same result is also presented in the journal version [GR08], but note that there is a notational difference in the definition of list recovery: the definition of (ζ, ℓ, L)-list recovery of [GR08] means $((1-\zeta), \ell, L)$-list recovery of [Rud07] and this paper. Also remark Footnote 9.

Lemma 4.3 ([Rud07, Sec. 3.6]). *Let q be a prime power, $\gamma \in \mathbb{F}_q^*$ be a generator, $N := q-1$, $k < N$ be a positive integer, and m be a positive integer that divides N. For positive integers ℓ, r, and $s \leq m$ and a real $0 < \zeta < 1$, suppose that the following inequalities hold:*

$$\frac{(1-\zeta)N}{m} \geq \left(1+\frac{s}{r}\right)\frac{\sqrt[s+1]{N\ell k^s}}{m-s+1} \tag{10}$$

$$(r+s)\sqrt[s+1]{\frac{N\ell}{k}} < q. \tag{11}$$

Then, $\mathrm{RS}^{(m)}_{\mathbb{F}_q,\gamma,k}$ is (ζ,ℓ,L)-list recoverable where $L = q^s$.

4.3.2 Construction

We show that folded Reed-Solomon codes satisfy the requirements of Lemma 4.2 if we set parameters appropriately. In the following, whenever we substitute non-integer values into integer variables, there is an implicit flooring to integers which we omit writing. Fix $0 < c < c' < 1$, which defines $\ell = 2^{\lambda^c}$. Our choices of parameters are as follows:

- $q = 2^{2\lfloor \log \lambda \rfloor}$ (which automatically defines $N = q-1$), $m = 2^{\lfloor \log \lambda \rfloor} + 1$, and $n = N/m = 2^{\lfloor \log \lambda \rfloor} - 1$.[11]
- γ is an arbitrary generator of $\mathbb{F}_q^*$.
- $k = \alpha N$ for an arbitrary constant $5/6 < \alpha < 1$.

We set $C_\lambda := \mathrm{RS}^{(m)}_{\mathbb{F}_q,\gamma,k}$. By the above parameter setting, it is easy to see that we have $|\Sigma| = 2^{\lambda^{\Theta(1)}}$, $n = \Theta(\lambda)$, and $|C_\lambda| = q^{k+1} \geq 2^{n+\lambda}$. We show that $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ satisfies the requirements of Lemma 4.2. For notational simplicity, we omit λ from the subscript of C.

First item. We prove Item 1 of Lemma 4.2. First, we remark that we only have to prove that the requirement is satisfied for sufficiently large λ since we can set $L = q^N$ for finitely many λ for which (ζ,ℓ,L)-list recoverability is trivially satisfied for any ζ and ℓ. We apply Lemma 4.3 with the following parameters:

- $s = \lambda^{c'}$. Note that this satisfies the requirement $s \leq m$ in Lemma 4.3 for sufficiently large λ since $m = \Omega(\lambda)$ and $c' < 1$.
- $r = \lambda^{c''}$ for a constant $c' < c'' < 1$.
- $0 < \zeta < 1-\alpha$ is an arbitrary constant.

Based on the above parameter setting, we have $\lim_{\lambda\to\infty}(1+\frac{s}{r}) = 1$, $\lim_{\lambda\to\infty}\frac{m}{m-s+1} = 1$, and $\lim_{\lambda\to\infty}\sqrt[s+1]{\ell} = 1$ where we used $\ell = 2^{\lambda^c}$ and $c < c'$. Therefore, Equation (10) can be rearranged as follows:

$$1-\zeta \geq (1+o(1))\left(\frac{k}{N}\right)^{\frac{s}{s+1}} \tag{12}$$

[11] This is an example of the parameter choice. Any prime power of the form $q = nm+1$ where n and m are positive integers such that $n = \Omega(\lambda)$ and $m = \Omega(\lambda)$ suffices.

This is satisfied for sufficiently large s (which occurs for sufficiently large λ) since we assume $k = \alpha N$ and $\zeta < 1 - \alpha$.

Similarly, by our choice of parameters, the LHS of Equation (11) is $O(\lambda^{c''})$ and the RHS is $\Omega(\lambda^2)$. Since $c'' < 1$, Equation (11) also holds for sufficiently large λ.

Thus, by Lemma 4.3, $\mathrm{RS}^{(m)}_{\mathbb{F}_q,\gamma,k}$ with the above parameter setting is (ζ, ℓ, L)-list recoverable where $L = q^s \leq (\lambda^2)^{\lambda^{c'}} = 2^{\widetilde{O}(\lambda^{c'})}$. This means that Item 1 of Lemma 4.2 is satisfied.

Second item. Next, we prove Item 2 of Lemma 4.2. Since $C = \mathrm{RS}^{(m)}_{\mathbb{F}_q,\gamma,k}$ is a folded Reed-Solomon code, its dual $C^\perp$ is a folded generalized Reed-Solomon code $\mathrm{GRS}^{(m)}_{\mathbb{F}_q,\gamma,N-k-2,\mathbf{v}}$ for some $\mathbf{v} \in \mathbb{F}_q^N$. In the following, we think of an element of Σ^n as an element of $\mathbb{F}_q^N$ in the canonical way. Then, $C^\perp = \mathrm{GRS}^{(m)}_{\mathbb{F}_q,\gamma,N-k-2,\mathbf{v}}$ is identified with $\mathrm{GRS}_{\mathbb{F}_q,\gamma,N-k-2,\mathbf{v}}$. Let $d := N - k - 2$ and $0 < \epsilon < 0.09$ be a constant specified later. We define $\mathsf{Decode}_{C^\perp}$ as follows.

$\mathsf{Decode}_{C^\perp}(\mathbf{z})$**:** On input $\mathbf{z} \in \mathbb{F}_q^N$, it runs the list decoding algorithm $\mathrm{GRSListDecode}_{\mathbb{F}_q,\gamma,N-k-2,\mathbf{v}}(\mathbf{z})$ to get a list of codewords. If there is a unique $\mathbf{x}$ in the list such that $\mathsf{hw}(\mathbf{z}-\mathbf{x}) \leq (1/2+\epsilon)N$, it outputs $\mathbf{x}$, and otherwise outputs $\perp$.

We define a subset $\mathcal{G} \subseteq \mathbb{F}_q^N$ as follows.

$$\mathcal{G} := \{\mathbf{e} \in \mathbb{F}_q^N : \mathsf{hw}(\mathbf{e}) \leq (1/2+\epsilon)N \ \wedge \ \forall \mathbf{y} \in C^\perp \setminus \{\mathbf{0}\},\ \mathsf{hw}(\mathbf{e}-\mathbf{y}) > (1/2+\epsilon)N\}.$$

For any $\mathbf{x} \in C^\perp$ and $\mathbf{e} \in \mathcal{G}$, by the definition of $\mathcal{G}$, $\mathbf{x}$ is the only codeword of $C^\perp$ whose Hamming distance from $\mathbf{x}+\mathbf{e}$ is smaller than or equal to $(1/2+\epsilon)N$. Moreover, since $k = \alpha N$ for $\alpha > 5/6$ and $\epsilon < 0.09$, it holds that $N - \sqrt{dN} = N - \sqrt{(1-\alpha)N^2 - 2N} \geq (1-\sqrt{1-\alpha})N > 0.59N > (1/2+\epsilon)N$. Thus, for any $\mathbf{x} \in C^\perp$ and $\mathbf{e} \in \mathcal{G}$, the list output by $\mathrm{GRSListDecode}_{\mathbb{F}_q,\gamma,N-k-2,\mathbf{v}}(\mathbf{x}+\mathbf{e})$ must contain $\mathbf{x}$, which implies

$$\mathsf{Decode}_{C^\perp}(\mathbf{x}+\mathbf{e}) = \mathbf{x}.$$

Thus, it suffices to prove

$$\Pr_{\mathbf{e} \xleftarrow{\$} \mathcal{D}^n}[\mathbf{e} \notin \mathcal{G}] = 2^{-\Omega(\lambda)}$$

where $\mathcal{D}$ is the distribution as defined in Lemma 4.2.[12] For $\mathbf{e} \in \mathbb{F}_q^N$, we parse it as $\mathbf{e} = (\mathbf{e}_1, ..., \mathbf{e}_n) \in \Sigma^n$ and define $S_\mathbf{e} \subseteq [N]$ as the set of indices on which $\mathbf{e}_i = \mathbf{0}$, i.e.,

$$S_\mathbf{e} := \bigcup_{i \in [n]: \mathbf{e}_i = \mathbf{0}} \{(i-1)m+1, (i-1)m+2, \ldots, im\}.$$

By the definition of $\mathcal{D}$ and $n = \Theta(\lambda)$, Chernoff bound (Lemma 2.4) gives

$$\Pr_{\mathbf{e} \xleftarrow{\$} \mathcal{D}^n}[(1/2-\epsilon)N \leq |S_\mathbf{e}| \leq (1/2+\epsilon)N] \geq 1 - 2^{\Omega(\lambda)}.$$

Therefore, it suffices to prove

$$\Pr_{\mathbf{e} \xleftarrow{\$} \mathcal{D}^n}[\mathbf{e} \notin \mathcal{G} \mid S_\mathbf{e} = S^*] = 2^{-\Omega(\lambda)} \tag{13}$$

[12] $\mathcal{D}^n$ is defined as a distribution over Σ^n, but its sample can be interpreted as an element of $\mathbb{F}_q^N$ in the canonical way.

for all $S^* \subseteq [N]$ such that $(1/2-\epsilon)N \leq |S^*| \leq (1/2+\epsilon)N$. Fix such S^*. When $S_{\mathbf{e}} = S^*$, it is clear that we have $\mathsf{hw}(\mathbf{e}) \leq (1/2+\epsilon)N$ since $|S^*| \geq (1/2-\epsilon)N$. Thus, when $S_{\mathbf{e}} = S^*$ and $\mathbf{e} \notin \mathcal{G}$, there exists $\mathbf{y} \in C^\perp \setminus \{\mathbf{0}\}$ such that

$$\mathsf{hw}(\mathbf{e}-\mathbf{y}) \leq (1/2+\epsilon)N. \tag{14}$$

Let $\bar{S}^* := [N] \setminus S^*$. Then, it holds that[13]

$$\mathsf{hw}(\mathbf{e}-\mathbf{y}) = \mathsf{hw}(\mathbf{e}_{S^*} - \mathbf{y}_{S^*}) + \mathsf{hw}(\mathbf{e}_{\bar{S}^*} - \mathbf{y}_{\bar{S}^*}). \tag{15}$$

Since we assume $S^* = S_{\mathbf{e}}$, we have $\mathbf{e}_{S^*} = \mathbf{0}$. On the other hand, since $\mathbf{y} \neq \mathbf{0}$ and degree d non-zero polynomials have at most d roots, $\mathbf{y}$ can take 0 on at most d indices. In particular, we have

$$\mathsf{hw}(\mathbf{e}_{S^*} - \mathbf{y}_{S^*}) \geq |S^*| - d. \tag{16}$$

By combining Equations (14) to (16), we have

$$\mathsf{hw}(\mathbf{e}_{\bar{S}^*} - \mathbf{y}_{\bar{S}^*}) \leq (1/2+\epsilon)N - (|S^*| - d) \leq d + 2\epsilon N \tag{17}$$

where we used $|S^*| \geq (1/2-\epsilon)N$. That is, conditioned on $S_{\mathbf{e}} = S^*$, Equation (17) holds for some $\mathbf{y} \in C^\perp \setminus \{\mathbf{0}\}$ whenever $\mathbf{e} \notin \mathcal{G}$. Moreover, conditioned on $S_{\mathbf{e}} = S^*$, the distribution of $\mathbf{e}_{\bar{S}^*}$ is a direct product of $|\bar{S}^*|/m$ copies of the uniform distribution over $\mathbb{F}_q^m \setminus \{\mathbf{0}\}$ by the definition of $\mathcal{D}$. Since $q^m = 2^{\Omega(\lambda)}$, the distribution is statistically $2^{-\Omega(\lambda)}$-close to the uniform distribution over $\mathbb{F}_q^N$. Combining these observations, it holds that[14]

$$\Pr_{\mathbf{e} \xleftarrow{\$} \widetilde{\mathcal{D}}}[\mathbf{e} \notin \mathcal{G} \mid S_{\mathbf{e}} = S^*] \leq \Pr_{\mathbf{e}_{\bar{S}^*} \xleftarrow{\$} \mathbb{F}_q^{|\bar{S}^*|}}[\exists \mathbf{y} \in C^\perp \ \mathsf{hw}(\mathbf{e}_{\bar{S}^*} - \mathbf{y}_{\bar{S}^*}) \leq d + 2\epsilon N] + 2^{-\Omega(\lambda)}. \tag{18}$$

When there exists $\mathbf{y} \in C^\perp$ such that $\mathsf{hw}(\mathbf{e}_{\bar{S}^*} - \mathbf{y}_{\bar{S}^*}) \leq d + 2\epsilon N$, there is a subset $T \subseteq \bar{S}^*$ such that $|T| = |\bar{S}^*| - \lceil d + 2\epsilon N \rceil$ and $\mathbf{e}_T = \mathbf{y}_T$.[15] On the other hand, since a codeword of $C^\perp$ is determined by values on $d+1$ indices, for any fixed $T \subseteq S^*$, we have

$$\Pr_{\mathbf{e}_{\bar{S}^*} \xleftarrow{\$} \mathbb{F}_q^{|\bar{S}^*|}}[\exists \mathbf{y} \in C^\perp \ \mathbf{e}_T = \mathbf{y}_T] = q^{-(|T|-(d+1))} \leq q^{-\left(\frac{1}{2}-3\epsilon\right)N+2d+1} \tag{19}$$

where we used $|T| \geq |\bar{S}^*| - d - 2\epsilon N$ and $|\bar{S}^*| \geq (1/2-\epsilon)N$. Since there are $\binom{|\bar{S}^*|}{\lceil d+2\epsilon N \rceil}$ possible choices of T, combined with Equation (19), it holds that

$$\begin{aligned}
\Pr_{\mathbf{e}_{\bar{S}^*} \xleftarrow{\$} \mathbb{F}_q^{|\bar{S}^*|}}[\exists \mathbf{y} \in C^\perp \ \mathsf{hw}(\mathbf{e}_{\bar{S}^*} - \mathbf{y}_{\bar{S}^*}) \leq d + 2\epsilon N] &\leq \binom{|\bar{S}^*|}{\lceil d+2\epsilon N \rceil} \cdot q^{-\left(\frac{1}{2}-3\epsilon\right)N+2d+1} \\
&\leq q^{d+2\epsilon N+1} \cdot q^{-\left(\frac{1}{2}-3\epsilon\right)N+2d+1} \\
&\leq q^{-\left(\frac{1}{2}-3(1-\alpha)-5\epsilon\right)N-4}
\end{aligned} \tag{20}$$

where we used $|\bar{S}^*| \leq N < q$ in the second inequality and $d = N-k-2 = (1-\alpha)N-2$ in the third inequality. Since $5/6 < \alpha < 1$, we can choose $0 < \epsilon < 0.09$ in such a way that $\frac{1}{2} - 3(1-\alpha) - 5\epsilon > 0$. (For example, $\epsilon := -\frac{1}{4} + \frac{3}{10}\alpha$ suffices.) Then, by combining Equations (18) and (20) together with $q = \Omega(\lambda)$ and $\frac{1}{2} - 3(1-\alpha) - 5\epsilon = \Omega(1)$, we obtain Equation (13).

[13] Recall the notation $\mathbf{x}_S = (x_i)_{i \in S}$ for $\mathbf{x} = (x_1, \ldots, x_N) \in \mathbb{F}_q^N$ and $S \subseteq [N]$.

[14] We can take $\exists \mathbf{y} \in C^\perp$ instead of $\exists \mathbf{y} \in C^\perp \setminus \{\mathbf{0}\}$ in the RHS since this does not decrease the probability. Indeed, one can see that the probabilities are the same noting that $\mathbf{e}_{\bar{S}^*}$ does not take 0 on any index and $|\bar{S}^*| > d + 2\epsilon N$ by our parameter choices.

[15] We always have $|\bar{S}^*| > \lceil d + 2\epsilon N \rceil$ by our parameter choices. Even without checking this, we can assume it without loss of generality since otherwise the probability in the RHS of Equation (18) is 0 and thus Equation (13) trivially holds.

Third item. Finally, we prove Item 3 of Lemma 4.2. For $\lceil \frac{k+1}{m} \rceil < j < n$, there does not exist a codeword $\mathbf{x}$ such that $\mathsf{hw}(\mathbf{x}) = n - j$. This is because if $\mathsf{hw}(\mathbf{x}) = n - j$, the corresponding polynomial f to $\mathbf{x}$ has at least $mj \geq k + 1$ roots, which means that $\mathbf{x} = \mathbf{0}$ since the degree of f is at most k. This contradicts $\mathsf{hw}(\mathbf{x}) = n - j > 0$.

The case of $j \leq \lceil \frac{k+1}{m} \rceil$ is proven below. In this case, since a polynomial of degree at most k is determined by evaluated values on $k + 1$ points, for any subset $S \subseteq [n]$ such that $|S| = j$, $\mathbf{x}_S$ is uniformly distributed over Σ^j when $\mathbf{x} \xleftarrow{\$} C_\lambda$. Therefore, we have

$$\begin{aligned}\Pr_{\mathbf{x} \xleftarrow{\$} C_\lambda}[\mathsf{hw}(\mathbf{x}) = n - j] &\leq \sum_{S \subseteq [n] \text{ s.t. } |S| = j} \Pr_{\mathbf{x} \xleftarrow{\$} C_\lambda}[\mathbf{x}_S = \mathbf{0}] \\ &\leq \binom{n}{j} |\Sigma|^{-j} \\ &\leq \left(\frac{n}{|\Sigma|}\right)^j .\end{aligned}$$

This completes the proof of Lemma 4.2.

5 Technical Lemma

We prepare a lemma that is used in the proof of correctness of our proof of quantumness constructed in Section 6. The lemma is inspired by the quantum step of Regev's reduction from LWE to worst-case lattice problems [Reg05].

Lemma 5.1. *Let $|\psi\rangle$ and $|\phi\rangle$ be quantum states on a quantum system over an alphabet $\Sigma = \mathbb{F}_q^m$ written as*

$$\begin{aligned}|\psi\rangle &= \sum_{\mathbf{x} \in \Sigma^n} V(\mathbf{x}) |\mathbf{x}\rangle \\ |\phi\rangle &= \sum_{\mathbf{e} \in \Sigma^n} W(\mathbf{e}) |\mathbf{e}\rangle .\end{aligned}$$

Let $F : \Sigma^n \to \Sigma^n$ be a function. Let $\mathsf{GOOD} \subseteq \Sigma^n \times \Sigma^n$ be a subset such that for any $(\mathbf{x}, \mathbf{e}) \in \mathsf{GOOD}$, we have $F(\mathbf{x} + \mathbf{e}) = \mathbf{x}$. Let BAD be the complement of GOOD, i.e., $\mathsf{BAD} := (\Sigma^n \times \Sigma^n) \setminus \mathsf{GOOD}$. Suppose that we have

$$\sum_{(\mathbf{x},\mathbf{e}) \in \mathsf{BAD}} |\hat{V}(\mathbf{x})\hat{W}(\mathbf{e})|^2 \leq \epsilon \tag{21}$$

$$\sum_{\mathbf{z} \in \Sigma^n} \left| \sum_{(\mathbf{x},\mathbf{e}) \in \mathsf{BAD} : \mathbf{x}+\mathbf{e}=\mathbf{z}} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \right|^2 \leq \delta. \tag{22}$$

Let U_{add} and U_F be unitaries defined as follows:

$$|\mathbf{x}\rangle |\mathbf{e}\rangle \xrightarrow{U_{\mathsf{add}}} |\mathbf{x}\rangle |\mathbf{x} + \mathbf{e}\rangle \xrightarrow{U_F} |\mathbf{x} - F(\mathbf{x} + \mathbf{e})\rangle |\mathbf{x} + \mathbf{e}\rangle .$$

Then we have

$$(I \otimes \mathsf{QFT}^{-1}) U_F U_{\mathsf{add}} (\mathsf{QFT} \otimes \mathsf{QFT}) |\psi\rangle |\phi\rangle) \approx_{\sqrt{\epsilon}+\sqrt{\delta}} |\Sigma|^{n/2} \sum_{\mathbf{z} \in \Sigma^n} (V \cdot W)(\mathbf{z}) |0\rangle |\mathbf{z}\rangle .$$

Proof. Equations (21) and (22) immediately imply the following inequalities, respectively:

$$\left\| \sum_{(\mathbf{x},\mathbf{e})\in \mathsf{BAD}} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|\mathbf{x}\right\rangle \left|\mathbf{e}\right\rangle \right\| \leq \sqrt{\epsilon}$$

and

$$\left\| \sum_{(\mathbf{x},\mathbf{e})\in \mathsf{BAD}} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|\mathbf{x}+\mathbf{e}\right\rangle \right\| \leq \sqrt{\delta}.$$

Since BAD is the complement of GOOD, the above imply the following:

$$\sum_{(\mathbf{x},\mathbf{e})\in \Sigma^n\times\Sigma^n} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|\mathbf{x}\right\rangle \left|\mathbf{e}\right\rangle \approx_{\sqrt{\epsilon}} \sum_{(\mathbf{x},\mathbf{e})\in \mathsf{GOOD}} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|\mathbf{x}\right\rangle \left|\mathbf{e}\right\rangle \tag{23}$$

and

$$\sum_{(\mathbf{x},\mathbf{e})\in \Sigma^n\times\Sigma^n} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|\mathbf{x}+\mathbf{e}\right\rangle \approx_{\sqrt{\delta}} \sum_{(\mathbf{x},\mathbf{e})\in \mathsf{GOOD}} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|\mathbf{x}+\mathbf{e}\right\rangle . \tag{24}$$

Then, we have

$$\begin{aligned}
U_F U_{\mathsf{add}}(\mathsf{QFT}\otimes\mathsf{QFT})\left|\psi\right\rangle\left|\phi\right\rangle &= U_F U_{\mathsf{add}} \sum_{(\mathbf{x},\mathbf{e})\in \Sigma^n\times\Sigma^n} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|\mathbf{x}\right\rangle \left|\mathbf{e}\right\rangle \\
&\approx_{\sqrt{\epsilon}} U_F U_{\mathsf{add}} \sum_{(\mathbf{x},\mathbf{e})\in \mathsf{GOOD}} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|\mathbf{x}\right\rangle \left|\mathbf{e}\right\rangle \\
&= \sum_{(\mathbf{x},\mathbf{e})\in \mathsf{GOOD}} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|0\right\rangle \left|\mathbf{x}+\mathbf{e}\right\rangle \\
&\approx_{\sqrt{\delta}} \sum_{(\mathbf{x},\mathbf{e})\in \Sigma^n\times\Sigma^n} \hat{V}(\mathbf{x})\hat{W}(\mathbf{e}) \left|0\right\rangle \left|\mathbf{x}+\mathbf{e}\right\rangle \\
&= \sum_{\mathbf{z}\in\Sigma^n} (\hat{V} * \hat{W})(\mathbf{z}) \left|0\right\rangle \left|\mathbf{z}\right\rangle \\
&= |\Sigma|^{n/2} \sum_{\mathbf{z}\in\Sigma^n} \widehat{(V\cdot W)}(\mathbf{z}) \left|0\right\rangle \left|\mathbf{z}\right\rangle \\
&= (I\otimes\mathsf{QFT})|\Sigma|^{n/2} \sum_{\mathbf{z}\in\Sigma^n} (V\cdot W)(\mathbf{z}) \left|0\right\rangle \left|\mathbf{z}\right\rangle
\end{aligned}$$

where we used Equation (23) for the second line, Equation (24) for the fourth line, and the convolution theorem (Equation (4) in Lemma 2.3) for the sixth line. This completes the proof of Lemma 5.1. □

6 Proofs of Quantumness

In this section, we give a construction of proofs of quantumness in the QROM, which is the main result of this paper.

Theorem 6.1. *There exists a keyless proof of quantumness in the QROM with soundness against uniform oracle-dependent adversaries.*

By Theorem 3.8, we immediately obtain the following corollary.

Corollary 6.2. *There exists a keyed proof of quantumness in the QROM with soundness against non-uniform oracle-dependent adversaries.*

The rest of this subsection is devoted to a proof of Theorem 6.1.

Construction. Let $\{C_\lambda\}_\lambda$ be a family of codes over an alphabet $\Sigma = \mathbb{F}_q^m$ that satisfies the requirements of Lemma 4.2 with arbitrary $1 < c < c' < 1$. In the following, we omit λ from the subscript of C since it is clear from the context. We use notations defined in Lemma 4.2 (e.g., n, m, ζ, ℓ, L etc). Let $H : \Sigma \to \{0,1\}^n$ be a random oracle.[16] For $i \in [n]$, let $H_i : \Sigma \to \{0,1\}$ be a function that on input x outputs the i-th bit of $H(x)$. Then, we construct a proof of quantumness in the QROM as follows.

$\mathsf{Prove}^H(1^\lambda)$**:** For $i \in [n]$, it generates a state

$$|\phi_i\rangle \propto \sum_{\mathbf{e}_i \in \Sigma \text{ s.t. } H_i(\mathbf{e}_i)=1} |\mathbf{e}_i\rangle .$$

This is done as follows. It generates a uniform superposition over Σ, coherently evaluates H, and measures its value. If the measurement outcome is 1, then it succeeds in generating the above state. It repeats the above procedure until it succeeds or it fails λ times. If it fails to generate $|\phi_i\rangle$ within λ trials for some $i \in [n]$, it just aborts. Otherwise, it sets

$$|\phi\rangle := |\phi_1\rangle \otimes |\phi_2\rangle \otimes \ldots \otimes |\phi_n\rangle .$$

Note that we have

$$|\phi\rangle \propto \sum_{\substack{\mathbf{e}=(\mathbf{e}_1,\ldots,\mathbf{e}_n)\in\Sigma^n \text{ s.t.} \\ H_i(\mathbf{e}_i)=1 \text{ for all } i\in[n]}} |\mathbf{e}\rangle .$$

It generates a state

$$|\psi\rangle \propto \sum_{\mathbf{x}\in C} |\mathbf{x}\rangle .$$

Then it applies QFT to both $|\psi\rangle$ and $|\phi\rangle$. At this point, it has the state

$$|\eta\rangle := \mathsf{QFT}\,|\psi\rangle \otimes \mathsf{QFT}\,|\phi\rangle .$$

Let U_{add} and U_{decode} be unitaries on the Hilbert space of $|\eta\rangle$ defined by the following:

$$|\mathbf{x}\rangle\,|\mathbf{e}\rangle \xrightarrow{U_{\mathsf{add}}} |\mathbf{x}\rangle\,|\mathbf{x}+\mathbf{e}\rangle \xrightarrow{U_{\mathsf{decode}}} |\mathbf{x}-\mathsf{Decode}_{C^\perp}(\mathbf{x}+\mathbf{e})\rangle\,|\mathbf{x}+\mathbf{e}\rangle$$

where $\mathsf{Decode}_{C^\perp}$ is the decoder for $C^\perp$ as required in Item 2 of Lemma 4.2. Then it applies $(I \otimes \mathsf{QFT}^{-1})U_{\mathsf{decode}}U_{\mathsf{add}}$ to $|\eta\rangle$, measures the second register, and outputs the measurement outcome $\mathbf{x} \in \Sigma^n$ as π.

$\mathsf{Verify}^H(1^\lambda, \pi)$**:** It parses $\pi = \mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and outputs $\top$ if $\mathbf{x} \in C$ and $H_i(\mathbf{x}_i) = 1$ for all $i \in [n]$ and $\bot$ otherwise.

[16] Strictly speaking, we consider a random oracle with the domain $\{0,1\}^*$. However, since our construction only makes queries to H on (bit representaions of) elements of Σ for a fixed security parameter, we simply denote by H to mean the restriction of H to (bit representations of) Σ.

Correctness.

Lemma 6.3. Π *satisfies correctness.*

Proof. Let $T_i^{H_i} \subseteq \Sigma$ be the subset consisting of $\mathbf{e}_i \in \Sigma$ such that $H_i(\mathbf{e}_i) = 1$ and $T^H := T_1^{H_1} \times T_2^{H_2} \times \ldots \times T_n^{H_n} \subseteq \Sigma^n$. Let $\widetilde{\mathcal{H}} \subseteq \mathsf{Func}(\Sigma, \{0,1\}^n)$ be the subset that consists of all $H \in \mathsf{Func}(\Sigma, \{0,1\}^n)$ such that $\frac{1}{3} < \frac{|T_i^{H_i}|}{|\Sigma|} < \frac{2}{3}$ for all $i \in [n]$. By the Chernoff bound (Lemma 2.4) and union bound, we can see that $(1 - n \cdot 2^{-\Omega(|\Sigma|)})$-fraction of $H \in (\Sigma, \{0,1\}^n)$ belongs to $\widetilde{\mathcal{H}}$. Since we have $n \cdot 2^{-|\Sigma|} = \mathsf{negl}(\lambda)$ by our parameter choices specified in Lemma 4.2, it suffices to prove the correctness assuming that H is uniformly chosen from $\widetilde{\mathcal{H}}$ instead of from $\mathsf{Func}(\Sigma, \{0,1\}^n)$. We prove this below.

First, we show that the probability that Prove aborts is negligible. In each trial to generate $|\phi_i\rangle$, the success probability is $\frac{|T_i^{H_i}|}{|\Sigma|} < \frac{2}{3}$. Thus, the probability that it fails to generate $|\phi_i\rangle$ λ times is negligible.

Let $V : \Sigma^n \to \mathbb{C}$, $W_i^{H_i} : \Sigma \to \mathbb{C}$, and $W^H : \Sigma^n \to \mathbb{C}$ be functions defined as follows:[17]

$$V(\mathbf{x}) = \begin{cases} \frac{1}{\sqrt{|C|}} & \mathbf{x} \in C \\ 0 & \text{otherwise} \end{cases}$$

$$W_i^{H_i}(\mathbf{e}_i) = \begin{cases} \frac{1}{\sqrt{|T_i^{H_i}|}} & \mathbf{e}_i \in T_i^{H_i} \\ 0 & \text{otherwise} \end{cases}$$

$$W^H(\mathbf{e}) = \begin{cases} \frac{1}{\sqrt{|T^H|}} & \mathbf{e} \in T^H \\ 0 & \text{otherwise} \end{cases}$$

Then we have

$$|\psi\rangle = \sum_{\mathbf{x} \in \Sigma^n} V(\mathbf{x}) |\mathbf{x}\rangle$$

$$|\phi\rangle = \sum_{\mathbf{e} \in \Sigma^n} W^H(\mathbf{e}) |\mathbf{e}\rangle$$

where $|\psi\rangle$ and $|\phi\rangle$ are as in the description of Prove. For using Lemma 5.1, we prove the following claim.

Claim 6.4. *For an overwhelming fraction of* $H \in \widetilde{\mathcal{H}}$*, there is a subset* $\mathsf{GOOD} \subseteq \Sigma^n \times \Sigma^n$ *such that* $\mathsf{Decode}_{C^\perp}(\mathbf{x} + \mathbf{e}) = \mathbf{x}$ *for any* $(\mathbf{x}, \mathbf{e}) \in \mathsf{GOOD}$ *and we have*

$$\sum_{(\mathbf{x},\mathbf{e}) \in \mathsf{BAD}} |\hat{V}(\mathbf{x}) \hat{W}^H(\mathbf{e})|^2 \leq \mathsf{negl}(\lambda),$$

$$\sum_{\mathbf{z} \in \Sigma^n} \left| \sum_{(\mathbf{x},\mathbf{e}) \in \mathsf{BAD} : \mathbf{x}+\mathbf{e}=\mathbf{z}} \hat{V}(\mathbf{x}) \hat{W}^H(\mathbf{e}) \right|^2 \leq \mathsf{negl}(\lambda).$$

where $\mathsf{BAD} = (\Sigma^n \times \Sigma^n) \setminus \mathsf{GOOD}$.

[17] Since we assume that H is sampled from $\widetilde{\mathcal{H}}$, we do not define them when $|T_i^{H_i}| = 0$ for some i.

We prove Claim 6.4 later. We complete the proof of Lemma 6.3 by using Claim 6.4. By Lemma 5.1 and Claim 6.4 where we set $F := \mathsf{Decode}_{C^\perp}$, for an overwhelming fraction of $H \in \widetilde{\mathcal{H}}$, we have

$$(I \otimes \mathsf{QFT}^{-1})U_{\mathsf{decode}}U_{\mathsf{add}}\left|\eta\right\rangle \approx |\Sigma|^{n/2}\sum_{\mathbf{x}\in\Sigma^n}(V\cdot W^H)(\mathbf{x})\left|0\right\rangle\left|\mathbf{x}\right\rangle \tag{25}$$

where $|\eta\rangle$ is as in the description of Prove. Since $(V\cdot W^H)(\mathbf{x}) = 0$ for $\mathbf{x} \notin C \cap T^H$, if we measure the second register of the RHS of Equation (25), the outcome is in $C \cap T^H$ with probability 1. Thus, if we measure the second register of the LHS of Equation (25), the outcome is in $C \cap S$ with probability $1 - \mathsf{negl}(\lambda)$. This means that an honestly generated proof π passes the verification with probability $1 - \mathsf{negl}(\lambda)$. □

To complete the proof of correctness, we prove Claim 6.4 below.

Proof of Claim 6.4. We use the notations defined in the proof of Lemma 6.3 above. For each $i \in [n]$, let $\widetilde{\mathcal{H}}_i \subseteq \mathsf{Func}(\Sigma, \{0,1\})$ be the subset that consists of all $H_i \in \mathsf{Func}(\Sigma, \{0,1\})$ such that $\frac{1}{3} < \frac{|T_i^{H_i}|}{|\Sigma|} < \frac{2}{3}$.[18] Choosing $H \xleftarrow{\$} \widetilde{\mathcal{H}}$ is equivalent to choosing $H_i \xleftarrow{\$} \widetilde{\mathcal{H}}_i$ independently for each $i \in [n]$. In the following, whenever we write H or H_i in subscripts of $\mathbb{E}$, they are uniformly taken from $\widetilde{\mathcal{H}}$ or $\widetilde{\mathcal{H}}_i$, respectively.

By Lemma 4.1 and the definition of V, we have

$$\hat{V}(\mathbf{x}) = \begin{cases} \frac{1}{\sqrt{|C^\perp|}} & \mathbf{x} \in C^\perp \\ 0 & \text{otherwise} \end{cases}.$$

Let $\mathcal{G} \subseteq \Sigma^n$ be a subset defined as follows:

$$\mathcal{G} := \{\mathbf{e} \in \Sigma^n : \forall \mathbf{x} \in C^\perp,\ \mathsf{Decode}_{C^\perp}(\mathbf{x}+\mathbf{e}) = \mathbf{x}\}.$$

Let $\mathcal{B} := \Sigma^n \setminus \mathcal{G}$. Item 2 of Lemma 4.2 implies

$$\Pr_{\mathbf{e}\xleftarrow{\$}\mathcal{D}^n}[\mathbf{e} \in \mathcal{B}] = \mathsf{negl}(\lambda) \tag{26}$$

where $\mathcal{D}$ is the distribution as defined in Item 2 of Lemma 4.2. We define $\mathsf{GOOD} := C^\perp \times \mathcal{G}$ and $\mathsf{BAD} := (\Sigma^n \times \Sigma^n) \setminus \mathsf{GOOD}$. Then, we have $\mathsf{Decode}_{C^\perp}(\mathbf{x}+\mathbf{e}) = \mathbf{x}$ for all $(\mathbf{x},\mathbf{e}) \in \mathsf{GOOD}$ by definition.

Noting that $\hat{V}(\mathbf{x}) = 0$ for $\mathbf{x} \notin C^\perp$, it is easy to see that we have the following:

$$\sum_{(\mathbf{x},\mathbf{e})\in\mathsf{BAD}} |\hat{V}(\mathbf{x})\hat{W}^H(\mathbf{e})|^2 = \sum_{\mathbf{e}\in\mathcal{B}} |\hat{W}^H(\mathbf{e})|^2, \tag{27}$$

$$\sum_{\mathbf{z}\in\Sigma^n}\left|\sum_{(\mathbf{x},\mathbf{e})\in\mathsf{BAD}:\mathbf{x}+\mathbf{e}=\mathbf{z}} \hat{V}(\mathbf{x})\hat{W}^H(\mathbf{e})\right|^2 = \sum_{\mathbf{z}\in\Sigma^n}\left|\sum_{\substack{\mathbf{x}\in C^\perp,\mathbf{e}\in\mathcal{B}\\ :\mathbf{x}+\mathbf{e}=\mathbf{z}}} \hat{V}(\mathbf{x})\hat{W}^H(\mathbf{e})\right|^2. \tag{28}$$

[18] Mathematically, the set $\widetilde{\mathcal{H}}_i$ does not depend on i. We index it by i for notational convenience.

We should prove that values of Equations (27) and (28) are negligible for an overwhelming fraction of $H \in \widetilde{\mathcal{H}}$. By a standard averaging argument, it suffices to prove that their expected values are negligible, i.e.,

$$\mathop{\mathbb{E}}_{H}\left[\sum_{\mathbf{e}\in\mathcal{B}}|\hat{W}^{H}(\mathbf{e})|^{2}\right] \leq \mathsf{negl}(\lambda), \tag{29}$$

$$\mathop{\mathbb{E}}_{H}\left[\sum_{\mathbf{z}\in\Sigma^{n}}\left|\sum_{\substack{\mathbf{x}\in C^{\perp},\mathbf{e}\in\mathcal{B}\\ :\mathbf{x}+\mathbf{e}=\mathbf{z}}}\hat{V}(\mathbf{x})\hat{W}^{H}(\mathbf{e})\right|^{2}\right] \leq \mathsf{negl}(\lambda). \tag{30}$$

Before proving them, we remark an obvious yet useful claim.

Claim 6.5. *Let π be a permutation over Σ (resp. Σ^n). Then, the distributions of H_i and $H_i \circ \pi$ (resp. H and $H \circ \pi$) are identical when $H_i \xleftarrow{\$} \widetilde{\mathcal{H}}_i$ (resp. $H \xleftarrow{\$} \widetilde{\mathcal{H}}$).*

Proof of Claim 6.5. Recall that $\widetilde{\mathcal{H}}_i$ is the set of all $H_i : \Sigma \to \{0,1\}$ such that $\frac{|\Sigma|}{3} < |\{\mathbf{e}_i \in \Sigma : H(\mathbf{e}_i) = 1\}| < \frac{2|\Sigma|}{3}$. Clearly, we have $|\{\mathbf{e}_i \in \Sigma : H(\mathbf{e}_i) = 1\}| = |\{\mathbf{e}_i \in \Sigma : H \circ \pi(\mathbf{e}_i) = 1\}|$. Thus, π induces a permutation over $\widetilde{\mathcal{H}}_i$, and thus $H_i \circ \pi$ is uniformly distributed over $\widetilde{\mathcal{H}}_i$ when $H_i \xleftarrow{\$} \widetilde{\mathcal{H}}_i$. A similar argument works for $\widetilde{\mathcal{H}}$ as well. □

Then, we prove Equations (29) and (30).

Proof of Equation (29). First, we prove the following claim.

Claim 6.6. *For all $i \in [n]$ and $\mathbf{e}, \mathbf{e}' \in \Sigma \setminus \{0\}$, it hold that*

$$\mathop{\mathbb{E}}_{H_i}\left[|\hat{W}_i(\mathbf{0})|^2\right] = \frac{1}{2} \tag{31}$$

and

$$\mathop{\mathbb{E}}_{H_i}\left[|\hat{W}_i(\mathbf{e})|^2\right] = \mathop{\mathbb{E}}_{H_i}\left[|\hat{W}_i(\mathbf{e}')|^2\right]. \tag{32}$$

Proof of Claim 6.6. Equation (31) is proven as follows.

$$\mathop{\mathbb{E}}_{H_i}\left[|\hat{W}_i(\mathbf{0})|^2\right] = \mathop{\mathbb{E}}_{H_i}\left[\left|\frac{1}{\sqrt{|\Sigma|}}\sum_{\mathbf{z}\in\Sigma} W_i^{H_i}(\mathbf{z})\right|^2\right] = \frac{\mathbb{E}_{H_i}\left[|T_i^{H_i}|\right]}{|\Sigma|} = \frac{1}{2}.$$

Since $\mathbf{e} \neq \mathbf{0}$, for any $w \in \mathbb{F}_q$, the number of $\mathbf{z} \in \Sigma$ such that $\mathbf{e} \cdot \mathbf{z} = w$ is $|\Sigma|/q$. A similar statement holds for $\mathbf{e}'$ too. Therefore, there is a permutation $\pi_{\mathbf{e},\mathbf{e}'} : \Sigma \to \Sigma$ such that $\mathbf{e} \cdot \mathbf{z} = \mathbf{e}' \cdot \pi_{\mathbf{e},\mathbf{e}'}(\mathbf{z})$ for

all $\mathbf{z} \in \Sigma$. Then, Equation (32) is proven as follows.

$$\begin{aligned}
\mathop{\mathbb{E}}_{H_i}\left[|\hat{W}_i(\mathbf{e})|^2\right] &= \mathop{\mathbb{E}}_{H_i}\left[\left|\frac{1}{\sqrt{|\Sigma|}}\sum_{\mathbf{z}\in\Sigma} W_i^{H_i}(\mathbf{z})\omega_p^{\mathrm{Tr}(\mathbf{e}\cdot\mathbf{z})}\right|^2\right] \\
&= \mathop{\mathbb{E}}_{H_i}\left[\left|\frac{1}{\sqrt{|\Sigma|}}\sum_{\mathbf{z}\in\Sigma} W_i^{H_i\circ\pi_{\mathbf{e},\mathbf{e}'}^{-1}}(\pi_{\mathbf{e},\mathbf{e}'}(\mathbf{z}))\omega_p^{\mathrm{Tr}(\mathbf{e}'\cdot\pi_{\mathbf{e},\mathbf{e}'}(\mathbf{z}))}\right|^2\right] \\
&= \mathop{\mathbb{E}}_{H_i}\left[\left|\frac{1}{\sqrt{|\Sigma|}}\sum_{\mathbf{z}\in\Sigma} W_i^{H_i\circ\pi_{\mathbf{e},\mathbf{e}'}^{-1}}(\mathbf{z})\omega_p^{\mathrm{Tr}(\mathbf{e}'\cdot\mathbf{z})}\right|^2\right] \\
&= \mathop{\mathbb{E}}_{H_i}\left[\left|\frac{1}{\sqrt{|\Sigma|}}\sum_{\mathbf{z}\in\Sigma} W_i^{H_i}(\mathbf{z})\omega_p^{\mathrm{Tr}(\mathbf{e}'\cdot\mathbf{z})}\right|^2\right] \\
&= \mathop{\mathbb{E}}_{H_i}\left[|\hat{W}_i(\mathbf{e}')|^2\right]
\end{aligned}$$

where the fourth equality follows from Claim 6.5. □

Claim 6.6 means that we have

$$\mathcal{D}(\mathbf{e}_i) = \mathop{\mathbb{E}}_{H_i}\left[|\hat{W}_i(\mathbf{e}_i)|^2\right] \tag{33}$$

for all $\mathbf{e}_i \in \Sigma$ where $\mathcal{D}(\cdot)$ is the probability density function of the distribution $\mathcal{D}$ as defined in Item 2 of Lemma 4.2. Moreover, for any $\mathbf{e} = (\mathbf{e}_1, \ldots, \mathbf{e}_n) \in \Sigma^n$ and $H \in \widetilde{\mathcal{H}}$, since we have $W^H(\mathbf{e}) = \prod_{i=1}^n W_i^{H_i}(\mathbf{e}_i)$, by Lemma 2.2, we have

$$\hat{W}^H(\mathbf{e}) = \prod_{i=1}^n \hat{W}_i^{H_i}(\mathbf{e}_i). \tag{34}$$

By combining Equations (33) and (34), we obtain

$$\mathcal{D}^n(\mathbf{e}) = \mathop{\mathbb{E}}_{H}\left[|\hat{W}(\mathbf{e})|^2\right] \tag{35}$$

for all $\mathbf{e} \in \Sigma^n$ where $\mathcal{D}^n(\cdot)$ is the probability density function of $\mathcal{D}^n$. By Equation (26), Equation (35), and the linearity of expectation, we obtain Equation (29).

Proof of Equation (30). We define a function $B : \Sigma^n \to \mathbb{C}$ so that $\hat{B}$ satisfies the following:[19]

$$\hat{B}(\mathbf{e}) = \begin{cases} 1 & \mathbf{e} \in \mathcal{B} \\ 0 & \text{otherwise} \end{cases}.$$

We prove the following claims.

Claim 6.7. *For any $H \in \widetilde{\mathcal{H}}$, it holds that*

$$\sum_{\mathbf{z}\in\Sigma^n}\left|\sum_{\substack{\mathbf{x}\in C^\perp, \mathbf{e}\in\mathcal{B}\\ :\mathbf{x}+\mathbf{e}=\mathbf{z}}} \hat{V}(\mathbf{x})\hat{W}^H(\mathbf{e})\right|^2 = \sum_{\mathbf{z}\in\Sigma^n}\left|(V\cdot(B * W^H))(\mathbf{z})\right|^2.$$

[19] That is, we first define $\hat{B}$ and then define B as its inverse discrete Fourier transform.

Proof of Claim 6.7. For any $\mathbf{z} \in \Sigma^n$, we have

$$\begin{aligned}\sum_{\substack{\mathbf{x}\in C^\perp, \mathbf{e}\in\mathcal{B}\\ :\mathbf{x}+\mathbf{e}=\mathbf{z}}} \hat{V}(\mathbf{x})\hat{W}^H(\mathbf{e}) &= \sum_{\substack{\mathbf{x}\in\Sigma^n, \mathbf{e}\in\Sigma^n\\ :\mathbf{x}+\mathbf{e}=\mathbf{z}}} \hat{V}(\mathbf{x})(\hat{B}\cdot\hat{W}^H)(\mathbf{e})\\ &= (\hat{V} * (\hat{B}\cdot\hat{W}^H))(\mathbf{z})\\ &= (V\cdot\widehat{(B * W^H)})(\mathbf{z})\end{aligned}$$

where we used $\hat{V}(\mathbf{x}) = 0$ for $\mathbf{x} \notin C^\perp$ in the first equality and the convolution theorem (Equation (6) in Lemma 2.3) in the third equality. Claim 6.7 follows from the above equation and Parceval's equality (Lemma 2.1). □

Claim 6.8. *For any $\mathbf{z} \in \Sigma^n$, it holds that*

$$\mathop{\mathbb{E}}_{H}\left[|(B * W^H)(\mathbf{z})|^2\right] \le \mathsf{negl}(\lambda).$$

Proof of Claim 6.8. First, we observe that $\mathbb{E}_H\left[|(B * W^H)(\mathbf{z}_0)|^2\right] = \mathbb{E}_H\left[|(B * W^H)(\mathbf{z}_1)|^2\right]$ for any $\mathbf{z}_0, \mathbf{z}_1$. Indeed, if we define a permutation $\pi : \Sigma^n \to \Sigma^n$ as $\pi(\mathbf{z}) := \mathbf{z} + \mathbf{z}_0 - \mathbf{z}_1$, we have

$$\begin{aligned}&\mathop{\mathbb{E}}_{H}\left[\left|(B * W^H)(\mathbf{z}_0)\right|^2\right]\\ =&\mathop{\mathbb{E}}_{H}\left[\left|\sum_{\mathbf{x}\in\Sigma^n} B(\mathbf{x})W^H(\mathbf{z}_0-\mathbf{x})\right|^2\right]\\ =&\mathop{\mathbb{E}}_{H}\left[\left|\sum_{\mathbf{x}\in\Sigma^n} B(\mathbf{x})W^{H\circ\pi}(\mathbf{z}_1-\mathbf{x})\right|^2\right]\\ =&\mathop{\mathbb{E}}_{H}\left[\left|\sum_{\mathbf{x}\in\Sigma^n} B(\mathbf{x})W^H(\mathbf{z}_1-\mathbf{x})\right|^2\right]\\ =&\mathop{\mathbb{E}}_{H}\left[\left|(B * W^H)(\mathbf{z}_1)\right|^2\right]\end{aligned}$$

where the third equality follows from Claim 6.5.

Then, for any $\mathbf{z} \in \Sigma^n$, we have

$$\begin{aligned}&\mathop{\mathbb{E}}_{H}\left[\left|(B * W^H)(\mathbf{z})\right|^2\right]\\ =&\frac{1}{|\Sigma|^n}\sum_{\mathbf{z}\in\Sigma^n}\mathop{\mathbb{E}}_{H}\left[\left|(B * W^H)(\mathbf{z})\right|^2\right]\\ =&\frac{1}{|\Sigma|^n}\mathop{\mathbb{E}}_{H}\left[\sum_{\mathbf{z}\in\Sigma^n}\left|(B * W^H)(\mathbf{z})\right|^2\right]\\ =&\frac{1}{|\Sigma|^n}\mathop{\mathbb{E}}_{H}\left[\sum_{\mathbf{z}\in\Sigma^n}\left||\Sigma|^{n/2}(\hat{B}\cdot\hat{W}^H)(\mathbf{z})\right|^2\right]\\ =&\mathop{\mathbb{E}}_{H}\left[\sum_{\mathbf{z}\in\mathcal{B}}\left|\hat{W}^H(\mathbf{z})\right|^2\right]\\ \le&\mathsf{negl}(\lambda).\end{aligned}$$

where the third equality follows from the convolution theorem (Equation (5) in Lemma 2.3) and Parceval's equality (Lemma 2.1) and the final inequality follows from Equation (29). □

Then, we prove Equation (30) as follows:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{H}\left[\sum_{\mathbf{z}\in\Sigma^n}\left|\sum_{\substack{\mathbf{x}\in C^\perp,\mathbf{e}\in\mathcal{B}\\:\mathbf{x}+\mathbf{e}=\mathbf{z}}}\hat{V}(\mathbf{x})\hat{W}^H(\mathbf{e})\right|^2\right]\\
=&\mathop{\mathbb{E}}_{H}\left[\sum_{\mathbf{z}\in\Sigma^n}\left|(V\cdot(B*W^H))(\mathbf{z})\right|^2\right]\\
=&\mathop{\mathbb{E}}_{H}\left[\sum_{\mathbf{z}\in C}\frac{1}{|C|}\left|(B*W^H))(\mathbf{z})\right|^2\right]\\
=&\frac{1}{|C|}\sum_{\mathbf{z}\in C}\mathop{\mathbb{E}}_{H}\left[\left|(B*W^H))(\mathbf{z})\right|^2\right]\\
\leq&\mathsf{negl}(\lambda).
\end{aligned}$$

where the first equality follows from Claim 6.7, the second equality follows from the definition of V, and the final inequality follows from Claim 6.8.

This completes the proof of Claim 6.4. □

Soundness.

Lemma 6.9. Π *satisfies* $(2^{\lambda^c}, 2^{-\Omega(\lambda)})$*-soundness against uniform oracle-dependent adversaries.*

Proof. By Theorem 3.6, we only have to prove $(2^{\lambda^c}, 2^{-\Omega(\lambda)})$-soundness against oracle-*independent* adversaries. We prove it below.

Let $\mathcal{A}$ be an oracle-independent adversary that makes $Q \leq 2^{\lambda^c}$ classical queries to H. Without loss of generality, we assume that $\mathcal{A}$ queries $\mathbf{x}_i^*$ to H at some point for all $i \in [n]$ where $\mathbf{x}^* = (\mathbf{x}_1^*, ..., \mathbf{x}_n^*) \in \Sigma^n$ is $\mathcal{A}$'s final output. Since a query to H can be replaced with queries to each of $H_1, \ldots, H_n$, there is an adversary $\mathcal{A}'$ that makes Q queries to each of H_1,...,H_n and succeeds with the same probability as $\mathcal{A}$. We denote $\mathcal{A}'$'s total number of queries by $Q' = nQ$. We remark that $\mathcal{A}'$ queries $\mathbf{x}_i^*$ to H_i at some point by our simplifying assumption on $\mathcal{A}$.

For each $i \in [n]$ and $j \in [Q']$, let $S_i^j \subseteq \Sigma$ be the set of elements that $\mathcal{A}'$ ever queried to H_i by the point when it has just made the j-th query counting queries to any of $H_1, ..., H_n$ in total. After the j-th query, we say that a codeword $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n) \in C$ is *K-queried* if there is a subset $I \in [n]$ such that $|I| = K$, $\mathbf{x}_i \in S_i^j$ for all $i \in I$, and $\mathbf{x}_i \notin S_i^j$ for all $i \notin I$. By our assumption, the final output $\mathbf{x}^*$ must be n-queried at the end. Since a K-queried codeword either becomes $(K+1)$-queried or remains K-queried by a single query, $\mathbf{x}^*$ must be K-queried at some point of the execution of $\mathcal{A}'$ for all $K = 0, 1, ..., n$.

We consider the number of codewords that ever become K-queried for $K = \lceil (1-\zeta)n \rceil$ where ζ is the constant as in Item 1 of Lemma 4.2. If $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n) \in C$ is $\lceil (1-\zeta)n \rceil$-queried at some point, we have $\mathbf{x}_i \in S_i^{Q'}$ for all $i \in [n]$ since $S_i^j \subseteq S_i^{Q'}$ for all i, j. By the construction of $\mathcal{A}'$, we have $|S_i^{Q'}| = Q \leq 2^{\lambda^c}$. On the other hand, C is (ζ, ℓ, L)-list recoverable for $\ell = 2^{\lambda^c}$ and $L = 2^{\tilde{O}(\lambda^{c'})}$ as required in Item 1 of Lemma 4.2. Thus, the number of codewords that ever become $\lceil (1-\zeta)n \rceil$-queried is at most $L = 2^{\tilde{O}(\lambda^{c'})}$.

Suppose that we simulate oracles $H_1, ..., H_n$ for $\mathcal{A}'$ via lazy sampling, that is, instead of uniformly choosing random functions at first, we sample function values whenever they are queried by $\mathcal{A}'$. Suppose that a codeword $\mathbf{x}$ becomes $\lceil(1-\zeta)n\rceil$-queried at some point of the execution of $\mathcal{A}'$. Since the function values on the unqueried $\lfloor\zeta n\rfloor$ positions are not sampled yet, $\mathbf{x}$ can become a valid proof only if all those values happen to be 1, which occurs with probability $\left(\frac{1}{2}\right)^{\lfloor\zeta n\rfloor} = 2^{-\Omega(\lambda)}$ by $\zeta = \Omega(1)$ and $n = \Omega(\lambda)$. Since one of them is the output $\mathbf{x}^*$, by the union bound, the probability that $\mathbf{x}^*$ is a valid proof is at most $L \cdot \left(\frac{1}{2}\right)^{\lfloor\zeta n\rfloor} = 2^{-\Omega(\lambda)}$ by $L = 2^{\tilde{O}(\lambda^{c'})}$ and $c' < 1$. This completes the proof Lemma 6.9. □

Theorem 6.1 follows from Lemmas 6.3 and 6.9.

7 Separations for Cryptographic Primitives

In this section, we give constructions of cryptographic primitives that are secure in the CROM but insecure in the QROM. They are easy consequences of our proof of quantumness in the QROM constructed in Section 6.

7.1 Separation for One-Way Functions

We give a construction of a family of functions that is one-way in the CROM but not one-way in the QROM. It is easy to generically construct such a one-way function from proofs of quantumness. Indeed, we prove a stronger claim than that in Section 7.2. Nonetheless, we give a direct construction with a similar structure to the proof of quantumness presented in Section 6. An interesting feature of the direct construction which the generic construction does not have is that it is not even *distributionally one-way* in the QROM as explained in Remark 4.

Theorem 7.1 (Separation for one-way functions)**.** *There exists a family $\{f_\lambda\}_\lambda$ of efficiently computable oracle-aided functions that is one-way against non-uniform oracle-dependent adversaries in the CROM but not one-way against oracle-independent adversaries in the QROM.*

Proof. By Theorem 3.7, it suffices to give a construction of a family $\{f_\lambda\}_\lambda$ that is one-way against oracle-*independent* adversaries in the CROM but not one-way against oracle-independent adversaries in the QROM. We prove this below.

The construction of f_λ is very similar to that of the proof of quantumness constructed in Section 6. We rely on similar parameter settings as in Section 6, and use similar notations as in Section 6.

We define $f_\lambda^H : C \to \{0,1\}^n$ as follows:

$$f_\lambda^H(\mathbf{x}_1, ..., \mathbf{x}_n) = (H_1(\mathbf{x}_1), ..., H_n(\mathbf{x}_n)).$$

where $H_i : \Sigma \to \{0,1\}$ is the function that outputs the i-th bit of the output of $H : \Sigma \to \{0,1\}^n$.

The Prove algorithm in Section 6 can be understood as an oracle-independent algorithm to invert f_λ for the image 1^n in the QROM. This can be extended to find a preimage of any image $y \in \{0,1\}^n$ in a straightforward manner: We only need to modify the definition of T_i^H to the subset consisting of $\mathbf{e}_i \in \Sigma$ such that $H_i(\mathbf{e}_i) = y_i$ instead of $H_i(\mathbf{e}_i) = 1$ in the proof of Lemma 6.3. The rest of the proof works analogously. Thus, $\{f_\lambda\}_\lambda$ is not one-way against oracle-independent adversaries in the QROM.

The proof of one-wayness in the CROM is similar to that of soundness of the proof of quantumness in Section 6. By a straightforward extension of the proof of Lemma 6.9 where we replace 1^n with arbitrary $y \in \{0,1\}^n$, we obtain the following claim.

Claim 7.2. *For any oracle-independent adversaries $\mathcal{A}$ that makes $\mathsf{poly}(\lambda)$ queries and $y \in \{0,1\}^n$,*

$$\Pr[y = f_\lambda^H(\mathbf{x}') : \mathbf{x}' \xleftarrow{\$} \mathcal{A}^H(1^\lambda, y)] < \mathsf{negl}(\lambda).$$

The above claim does not immediately imply one-wayness since in the one-wayness game, y is chosen by first sampling $\mathbf{x} \xleftarrow{\$} C$ and then setting $y = f_\lambda^H(\mathbf{x})$ instead of fixing y independently of H. Fortunately, we can show that the distribution of y is almost independent of H as shown in the following claim.

Claim 7.3. *We have*

$$\Delta((H, y), (H, y')) = \mathsf{negl}(\lambda)$$

where $H \xleftarrow{\$} \mathsf{Func}(\Sigma, \{0,1\}^n)$,$\mathbf{x} \xleftarrow{\$} C$, $y = f_\lambda^H(\mathbf{x})$, and $y' \xleftarrow{\$} \{0,1\}^n$.

By combining Claims 7.2 and 7.3, one-wayness against oracle-independent adversaries immediately follows.

For proving Claim 7.3, we rely on the following well-known lemma that relates the collision probability and statistical distance from the uniform distribution.

Definition 7.4. *For a random variable X over a finite set $\mathcal{X}$, we define its collision probability as* $\mathrm{Col}(X) = \sum_{x \in \mathcal{X}} \Pr[X = x]^2$.

Lemma 7.5. *Let X be a random variable over a finite set $\mathcal{X}$. For $\epsilon > 0$, if* $\mathrm{Col}(X) \leq \frac{1}{|\mathcal{X}|}(1 + \epsilon)$, *then*

$$\Delta(X, U_{\mathcal{X}}) \leq \sqrt{\epsilon}/2$$

where $U_{\mathcal{X}}$ denotes the uniform distribution over $\mathcal{X}$.

See e.g., [MV08, Lemma 4.5] for the proof of Lemma 7.5.

Then, we prove Claim 7.3 below.

Proof of Claim 7.3. By Lemma 7.5, it suffices to prove $\mathrm{Col}(H, y) = 2^{-(|\Sigma|+1)n} \cdot (1 + \mathsf{negl}(\lambda))$ where $H \xleftarrow{\$} \mathsf{Func}(\Sigma, \{0,1\}^n), \mathbf{x} \xleftarrow{\$} C, y = f_\lambda^H(\mathbf{x})$. We prove this as follows where H and H' are uniformly

sampled from $\mathsf{Func}(\Sigma,\{0,1\}^n)$ and $\mathbf{x}$ and $\mathbf{x}'$ are uniformly sampled from C.

$$\begin{aligned}
\mathrm{Col}(H,y) &= \Pr_{H,H',\mathbf{x},\mathbf{x}'}[H=H' \ \wedge \ f_\lambda^H(\mathbf{x})=f_\lambda^{H'}(\mathbf{x}')]\\
&= 2^{-|\Sigma|n}\cdot\Pr_{H,\mathbf{x},\mathbf{x}'}[f_\lambda^H(\mathbf{x})=f_\lambda^H(\mathbf{x}')]\\
&= 2^{-|\Sigma|n}\cdot\sum_{j=0}^{n}\Pr_{\mathbf{x},\mathbf{x}'}[\mathsf{hw}(\mathbf{x}-\mathbf{x}')=n-j]\cdot 2^{-(n-j)}\\
&= 2^{-|\Sigma|n}\cdot\sum_{j=0}^{n}\Pr_{\mathbf{x}}[\mathsf{hw}(\mathbf{x})=n-j]\cdot 2^{-(n-j)}\\
&\le 2^{-(|\Sigma|+1)n}\cdot\left(1+\frac{2^n}{|C_\lambda|}+\sum_{j=1}^{n-1}\Pr_{\mathbf{x}}[\mathsf{hw}(\mathbf{x})=n-j]\cdot 2^j\right)\\
&\le 2^{-(|\Sigma|+1)n}\cdot\left(1+\frac{2^n}{|C_\lambda|}+\sum_{j=1}^{n-1}\left(\frac{2n}{|\Sigma|}\right)^j\right)\\
&\le 2^{-(|\Sigma|+1)n}\cdot\left(1+\frac{2^n}{|C_\lambda|}+\sum_{j=1}^{\infty}\left(\frac{2n}{|\Sigma|}\right)^j\right)\\
&= 2^{-(|\Sigma|+1)n}\cdot\left(1+\frac{2^n}{|C_\lambda|}+\frac{\left(\frac{2n}{|\Sigma|}\right)}{1-\left(\frac{2n}{|\Sigma|}\right)}\right)\\
&= 2^{-(|\Sigma|+1)n}\cdot(1+\mathsf{negl}(\lambda))
\end{aligned}$$

where we used $\Pr_{\mathbf{x}}[\mathsf{hw}(\mathbf{x})=n]\le 1$ and $\Pr_{\mathbf{x}}[\mathsf{hw}(\mathbf{x})=0]=\frac{1}{|C_\lambda|}$ for the fifth line, Item 3 of Lemma 4.2 for the sixth line, and $|\Sigma|=2^{\lambda^{\Theta(1)}}$, $n=\Theta(\lambda)$, and $|C_\lambda|\ge 2^{n+\lambda}$ for the final line. This completes the proof of Claim 7.3. □

This completes the proof of Theorem 7.1. □

Remark 4 (On distributional one-wayness)**.** *It is worth mentioning that $\{f_\lambda\}_\lambda$ is not even* distributionally *one-way in the QROM. That is, one can find an almost uniformly distributed preimage of y with quantum oracle access to H. This can be seen by observing that the proof of Lemma 6.3 actually shows that the proving algorithm outputs an almost uniformly distributed valid proof. This corresponds to finding an almost uniformly distributed preimage of y for the above f_λ.*

7.2 Separation for Collision-Resistant Hash Functions.

We give a construction of a family of compressing functions that is collision-resistant in the CROM but not even one-way in the QROM. It is a generic construction based on proofs of quantumness.

Theorem 7.6 (Separation for collision-resistant functions)**.** *There exists a family $\{f_\lambda\}_\lambda$ of efficiently computable oracle-aided compressing keyless (resp. keyed) functions that is collision-resistant against uniform (resp. non-uniform) oracle-dependent adversaries in the CROM but not even one-way against oracle-independent adversaries in the QROM.*

Proof. Since the keyed version immediately follows from the keyless version by Theorem 3.8, we prove the keyless version below.

Let $(\mathsf{Prove}, \mathsf{Verify})$ be a keyless proof of quantumness with soundness against uniform oracle-dependent adversaries that is shown to exist in Theorem 6.1, and let ℓ_π be its maximum proof length.

We assume that the proof of quantumness uses a random oracle $H : \{0,1\}^{\lambda+\ell_\pi} \to \{0,1\}^\lambda$ without loss of generality. We construct $f_\lambda^H : \{0,1\}^{\lambda+\ell_\pi} \to \{0,1\}^\lambda$ as follows:

$$f_\lambda^H(x,\pi) := \begin{cases} x & \text{if } \mathsf{Verify}^H(1^\lambda,\pi) = \top \\ H(x,\pi) & \text{otherwise} \end{cases}$$

where the input is parsed as $x \in \{0,1\}^\lambda$ and $\pi \in \{0,1\}^{\ell_\pi}$. Collision-resistance of $\{f_\lambda\}_\lambda$ against uniform oracle-dependent adversaries in the CROM is clear from the soundness of the proof of quantumness. Indeed, an adversary with a classical access to H can output (x,π) such that $\mathsf{Verify}(1^\lambda,\pi) = \top$ only with a negligible probability. Assuming that this does not happen, an adversary has to find a collision of H, which can be done only with probability at most $\frac{Q(Q+1)}{2} \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$ where $Q = \mathsf{poly}(\lambda)$ is the number of queries to H. On the other hand, the correctness of the proof of quantumness gives a trivial way to invert f_λ^H on any target $y \in \{0,1\}^\lambda$ with a quantum access to H: one can just run $\pi \xleftarrow{\$} \mathsf{Prove}^H(1^\lambda)$ and then output (y,π). We have $f_\lambda^H(y,\pi) = y$ except for a negligible probability by the correctness of the proof of quantumness. This means that $\{f_\lambda\}_\lambda$ is not one-way against oracle-independent adversaries in the QROM. □

7.3 Separations for Public Key Primitives

In [YZ21], they give separations between security in the CROM and QROM for public key encryption (PKE) and digital signatures. Since their constructions are generic based on proofs of quantumness, we can plug our proofs of quantumness given in Section 6 into their constructions to obtain the following theorems.

Theorem 7.7. *If there exists a PKE scheme that is IND-CPA secure in the CROM, then there exists a PKE scheme that is IND-CCA secure in the CROM but not IND-CPA secure in the QROM.*

Theorem 7.8. *There exists a digital signature scheme that is EUF-CMA secure in the CROM but not EUF-NMA secure in the QROM.*

See [YZ21] for the formal definitions of PKE and digital signatures and their security. Note that [YZ21] proved similar theorems relative to additional artificial classical oracles and weaker variants of them assuming the LWE assumption. We significantly improve them by removing the necessity of additional oracles or complexity assumptions.

7.4 A Remark on Pseudorandom Generators

One might think that we can also construct pseudorandom generators (PRGs) that are secure in the CROM but insecure in the QROM because Theorem 7.1 gives one-way functions (OWFs) that are secure in the CROM but insecure in the QROM and there is a black-box construction of PRGs from OWFs [HILL99]. However, we remark that this does not work. The reason is that PRGs constructed from OWFs may be secure in the QROM even if the building block OWF is insecure in the QROM. For example, there is no obvious attack against the PRG of [HILL99] even with an inverter for the building block OWF.

Indeed, we believe that we can show that *any* black-box construction of PRGs from OWFs may remain secure even if the building block OWF is insecure. We sketch the intuition below. Let

$f : \mathcal{X} \to \mathcal{X}$ be a OWF. We augment the domain to $\mathcal{X} \times \mathcal{R}$ where $\mathcal{R}$ is an exponentially large space by defining

$$f'(x, r) := f(x).$$

Then, it is clear that f' is also a OWF. Suppose that we construct a PRG G by making black-box use of f'. Since f' is a secure OWF, $G^{f'}$ is a secure PRG. For each $r^* \in \mathcal{R}$, we define f'_{r^*} as follows:

$$f'_{r^*}(x, r) := \begin{cases} f(x) & \text{if } r \neq r^* \\ x & \text{otherwise} \end{cases}.$$

Then, f'_{r^*} clearly does not satisfy the one-wayness: for inverting y, one can just output (y, r^*). On the other hand, when we run G with respect to f'_{r^*} instead of f' for a randomly chosen r^*, there would be a negligibly small chance of calling the second branch of f'_{r^*} if the number of G's queries is polynomial. This means that G remains secure even though the building block function f'_{r^*} is insecure as a OWF.

We observe that the (im)possibility of separating CROM and QROM for PRGs is closely related to the Aaronson-Ambainis conjecture [AA14] (Conjecture 8.1). Very roughly speaking, the conjecture claims that any single-bit output algorithm in the QROM can be simulated in the CROM with a polynomial blowup on the number of queries. Since a PRG distinguisher's output is a single-bit, it is reasonable to expect that we can prove the equivalence of QROM security and CROM security for PRGs under the Aaronson-Ambainis conjecture. Unfortunately, this does not work as it is because a distinguisher takes a PRG value as its input, which may be correlated with the random oracle, whereas the Aaronson-Ambainis conjecture only captures the case where no side information of the random oracle is given. Nonetheless, we conjecture that QROM security and CROM security for PRGs (against polynomial-query unbounded-time adversaries) are equivalent. It is a fascinating direction for future work to reduce it to the Aaronson-Ambainis conjecture or its reasonable variant.

8 Proofs of Randomness

In this section, we construct proofs of randomness assuming the Aaronson-Ambainis conjecture [AA14].

Roughly speaking, the Aaronson-Ambainis conjecture claims that for any algorithm $\mathcal{A}$ with a *quantum* access to a random oracle, there is an algorithm $\mathcal{B}$ that approximates the probability that $\mathcal{A}$ outputs a particular output with a *classical* access to the random oracle, and the number of queries of $\mathcal{A}$ and $\mathcal{B}$ are polynomially related. A formal claim is stated below.

Conjecture 8.1 (Aaronson-Ambainis conjecture [AA14, Theorem 22])**.** *Let $\epsilon, \delta > 0$ be reals. Given any quantum algorithm $\mathcal{A}$ that makes Q quantum queries to a random oracle $H : \{0,1\}^n \to \{0,1\}^m$, there exists a deterministic classical algorithm $\mathcal{B}$ that makes $\mathsf{poly}(Q, m, \epsilon^{-1}, \delta^{-1})$ classical queries and satisfies*

$$\Pr_{H \xleftarrow{\$} \mathsf{Func}(\{0,1\}^n, \{0,1\}^m)} [\left|\Pr[\mathcal{A}^H() \to 1] - \mathcal{B}^H()\right| \leq \epsilon] \geq 1 - \delta.$$

Remark 5. *We remark that the way of stating the conjecture is slightly different from that in [AA14, Theorem 22], but they are equivalent. The difference is that [AA14] considers oracle access to Boolean inputs whereas we consider an oracle access to functions. They are equivalent by considering a function as a bit string concatenating outputs on all inputs. We remark that a straightforward rephrasing would result in an oracle with 1-bit outputs, but their conjecture is equivalent in*

the setting with m-bit output oracles since an m-bit output oracle can be seen as a concatenation of m 1-bit output oracles. We note that the number of $\mathcal{B}$'s queries in the above conjecture depends on m unlike theirs due to this difference.

We also remark that Aaronson and Ambainis [AA14] reduce the above conjecture to another seemingly unrelated conjecture in Fourier analysis. In the literature, the latter conjecture is often referred to as Aaronson-Ambainis conjecture. On the other hand, we call Conjecture 8.1 Aaronson-Ambainis conjecture since this is more relevant to our work.

The main theorem we prove in this section is the following.

Theorem 8.2. *If Conjecture 8.1 is true, there exists keyless (resp. keyed) proofs of randomness in the QROM that has true randomness against uniform (resp. non-uniform) oracle-dependent adversaries.*

By Theorems 3.6, 3.9 and 3.15, it suffices to prove the following theorem for proving Theorem 8.2.

Theorem 8.3. *If Conjecture 8.1 is true, there exists keyless proofs of min-entropy in the QROM that has min-entropy against oracle-independent adversaries.*

In the following, we prove Theorem 8.3.

From proofs of quantumness to proofs of min-entropy. Our proof of quantumness constructed in Section 6 has a large entropy in proofs. We can easily show that this is inherent assuming Aaronson-Ambainis conjecture. This is because if the proving algorithm is almost deterministic, it can be simulated by a polynomial-query classical algorithm, which breaks soundness. The following theorem gives a generalization of the above argument.

Theorem 8.4. *If Conjecture 8.1 is true, the following holds. Let $(\mathsf{Prove}, \mathsf{Verify})$ be a keyless proof of quantumness relative to a random oracle $H : \{0,1\}^n \to \{0,1\}^m$ that satisfies $(Q_{\mathsf{poq}}, \epsilon_{\mathsf{poq}})$-soundness. Let $\mathcal{A}$ be an oracle-independent adversary that makes $Q_{\mathcal{A}}$ quantum queries. Let $\epsilon_{\mathcal{A}}, \delta_{\mathcal{A}} > 0$ be reals (that may depend on the security parameter). There exists a polynomial p such that if we have*

$$Q_{\mathsf{poq}} \geq p(\lambda, Q_{\mathcal{A}}, \epsilon_{\mathcal{A}}^{-1}, \delta_{\mathcal{A}}^{-1})$$

and

$$\epsilon_{\mathsf{poq}} \leq \delta_{\mathcal{A}}/4,$$ [20]

then we have

$$\Pr_{H \xleftarrow{\$} \mathsf{Func}(\{0,1\}^n, \{0,1\}^m)} \left[\max_{\pi^* \text{ s.t. } \mathsf{Verify}^H(1^\lambda, \pi^*) = \top} \Pr[\mathcal{A}^H(1^\lambda) \to \pi^*] \leq \epsilon_{\mathcal{A}} \right] \geq 1 - \delta_{\mathcal{A}}.$$

We defer the proof of Theorem 8.4 to the end of this section. By plugging the proofs of quantumness in Section 6 into Theorem 8.4, we obtain proofs of min-entropy, which proves Theorem 8.3.

Proof of Theorem 8.3. For any polynomial $h(\lambda)$, there exists a constant C such that $Q_{\mathsf{poq}} = 2^{C(h(\lambda)+\lambda)}$ and $\epsilon_{\mathsf{poq}} = 2^{-\lambda-2}$ satisfy the requirements of Theorem 8.4 for $Q_{\mathcal{A}} = \mathsf{poly}(\lambda)$, $\epsilon_{\mathcal{A}} = 2^{-(h(\lambda)+\lambda)}$, and $\delta_{\mathcal{A}} = 2^{-\lambda}$. As shown in Lemma 6.9, our proof of quantumness constructed in Section 6, which we denote by $(\mathsf{Prove}_{\mathsf{poq}}, \mathsf{Verify}_{\mathsf{poq}})$, satisfies subexponential security. Thus, by

[20] In fact, it suffices to require $\epsilon_{\mathsf{poq}} \leq c\delta_{\mathcal{A}}$ for any constant $c < 1$.

standard complexity leveraging, there is a polynomial $h'(\lambda)$ such that if we replace the security parameter with $h'(\lambda)$ in $(\mathsf{Prove}_{\mathsf{poq}}, \mathsf{Verify}_{\mathsf{poq}})$, then it satisfies $(2^{C(h(\lambda)+\lambda)}, 2^{-\lambda-2})$-soundness. By Theorem 8.4, we have

$$\Pr_{H\xleftarrow{\$}\mathsf{Func}(\{0,1\}^n,\{0,1\}^m)}\left[\max_{\pi^* \text{ s.t. } \mathsf{Verify}^H_{\mathsf{poq}}(1^{h'(\lambda)},\pi^*)=\top} \Pr[\mathcal{A}^H(1^\lambda)\to\pi^*] \le 2^{-(h(\lambda)+\lambda)}\right] \ge 1-2^{-\lambda}. \tag{36}$$

Then, we construct proofs of min-entropy $(\mathsf{Prove}, \mathsf{Verify})$ as follows.

$\mathsf{Prove}^H(1^\lambda, h(\lambda)) := \mathsf{Prove}^H_{\mathsf{poq}}(1^{h'(\lambda)})$

$\mathsf{Verify}^H(1^\lambda, h(\lambda), \pi)$**:** If $\mathsf{Verify}^H_{\mathsf{poq}}(1^{h'(\lambda)}, \pi) = \bot$, it outputs $\bot$. Otherwise, it outputs π.

Suppose that $(\mathsf{Prove}, \mathsf{Verify})$ does not have min-entropy against oracle-independent adversaries. Then, there exist an oracle-independent adversary $\mathcal{A}$ that makes $\mathsf{poly}(\lambda)$ queries and a polynomial $h(\lambda)$ such that we have

$$\Pr[\mathsf{Verify}^H(1^\lambda, h(\lambda), \mathcal{A}^H(1^\lambda)) \neq \bot] \ge 1/\mathsf{poly}(\lambda) \wedge H_\infty\left(\mathcal{A}^H_\top(1^\lambda)\right) \le h(\lambda) \tag{37}$$

for a non-negligible fraction of H. It is easy to see that Equation (37) implies

$$\max_{\pi^* \text{ s.t. } \mathsf{Verify}^H_{\mathsf{poq}}(1^{h'(\lambda)},\pi^*)=\top} \Pr[\mathcal{A}^H(1^\lambda)\to\pi^*] \ge 2^{-h(\lambda)}/\mathsf{poly}(\lambda).$$

Since this holds for a non-negligible fraction of H, this contradicts Equation (36). Therefore, $(\mathsf{Prove}, \mathsf{Verify})$ has min-entropy against oracle-independent adversaries. □

Intuition for the proof of Theorem 8.4. Towards a contradiction, we assume that

$$\Pr_{H\xleftarrow{\$}\mathsf{Func}(\{0,1\}^n,\{0,1\}^m)}\left[\max_{\pi^* \text{ s.t. } \mathsf{Verify}^H(1^\lambda,\pi^*)=\top} \Pr[\mathcal{A}^H(1^\lambda)\to\pi^*] > \epsilon_{\mathcal{A}}\right] > \delta_{\mathcal{A}}.$$

We have to construct a classical adversary that breaks the soundness of the proof of quantumness. If $\epsilon_{\mathcal{A}} \approx 1$, it is easy: We consider an algorithm $\mathcal{A}_j$ that outputs the j-th bit of $\mathcal{A}$'s output for $j \in [\ell_\pi]$ where ℓ_π is the length of a proof in the proof of quantumness. For $\delta_{\mathcal{A}}$-fraction of H, $\mathcal{A}_j$'s output is almost deterministic for all j. Then, we can classically simulate $\mathcal{A}_j$ for all j by invoking Conjecture 8.1 for $\epsilon << 1$ and $\delta << \delta_{\mathcal{A}}/\ell_\pi$. This breaks the soundness of the proof of quantumness.

When $\epsilon_{\mathcal{A}} << 1$, such a simple bit-by-bit simulation attack does not work. The reason is that mixing up bits of multiple valid proofs does not result in a valid proof in general. To deal with such a case, we attempt to convert $\mathcal{A}$ into an almost deterministic attacker. If this is done, the same idea as the case of $\epsilon_{\mathcal{A}} \approx 1$ works. For making $\mathcal{A}$ almost deterministic, our first idea is to consider an modified adversary $\mathcal{A}'$ that outputs the smallest valid proof π in the lexicographical order such that $\mathcal{A}$ outputs π with probability at least $\epsilon_{\mathcal{A}}$. If we can efficiently construct such $\mathcal{A}'$, then this idea works. However, the problem is that $\mathcal{A}'$ cannot exactly compute the probabilities that $\mathcal{A}$ outputs each π with a limited number of queries. What $\mathcal{A}'$ can do is to run $\mathcal{A}$ many times to approximate the probabilities up to a $1/\mathsf{poly}$ error.[21] Now, a problem occurs if there are multiple π such that the probability that $\mathcal{A}$ outputs π is within $\epsilon_{\mathcal{A}} \pm 1/\mathsf{poly}$.

[21] poly means a polynomial in the number of repetition of $\mathcal{A}$ run by $\mathcal{A}'$.

To deal with this issue, we rely on an idea to randomly decide the threshold.[22] That is, $\mathcal{A}'$ outputs the lexigographically smallest valid proof π such that the approximated probability that $\mathcal{A}$ outputs π is at least t for some randomly chosen threshold $t \in (\epsilon_{\mathcal{A}}/2, \epsilon_{\mathcal{A}})$. If we choose t from a sufficiently large set and set the approximation error to be sufficiently small, we can show that it is impossible that there are multiple π such that the probability that $\mathcal{A}$ outputs π is within $t \pm 1/\mathsf{poly}$ for a large fraction of t by a simple counting argument. This resolves the above problem.

Proof of Theorem 8.4. In the rest of this section, we give a formal proof of Theorem 8.4. We first show the following simple lemma.

Lemma 8.5. *Let $\mathcal{A}$ be a (possibly quantum) algorithm that outputs an ℓ-bit string z. For any $\epsilon, \delta > 0$, there is an algorithm $\mathsf{Approx}(\mathcal{A}, \epsilon, \delta)$ that runs $\mathcal{A}$ $O(\ell \log(\delta^{-1})\epsilon^{-2})$ times and outputs a tuple $\{P_z\}_{z\in\{0,1\}^\ell}$ such that*

$$\Pr\left[\forall z \in \{0,1\}^\ell \ \ |P_z - \Pr[\mathcal{A}() \to z]| \leq \epsilon\right] \geq 1 - \delta$$

where $\{P_z\}_{z\in\{0,1\}^\ell} \xleftarrow{\$} \mathsf{Approx}(\mathcal{A}, \epsilon, \delta)$. We say that $\mathsf{Approx}(\mathcal{A}, \epsilon, \delta)$ succeeds if the event in the above probability occurs.

Proof. $\mathsf{Approx}(\mathcal{A}, \epsilon, \delta)$ works as follows. It runs $\mathcal{A}()$ N times where N is an integer specified later. For each z, let K_z be the number of executions where $\mathcal{A}$ outputs z. Then it outputs $\{P_z := \frac{K_z}{N}\}_{z\in\{0,1\}^\ell}$.

If we set $N \geq C\ell \log(\delta^{-1})\epsilon^{-2}$ for a sufficiently large constant C, by Chernoff bound (Lemma 2.4), for each z, we have

$$\Pr\left[|P_z - \Pr[\mathcal{A}() \to z]| \leq \epsilon\right] \geq 1 - \frac{\delta}{2^\ell}.$$

By the union bound, we obtain Lemma 8.5. □

Then, we prove Theorem 8.4.

Proof of Theorem 8.4. Towards a contradiction, we assume that

$$\Pr_{H\xleftarrow{\$}\mathsf{Func}(\{0,1\}^n,\{0,1\}^m)}\left[\max_{\pi^* \text{ s.t. } \mathsf{Verify}^H(1^\lambda,\pi^*)=\top} \Pr[\mathcal{A}^H(1^\lambda) \to \pi^*] > \epsilon_{\mathcal{A}}\right] > \delta_{\mathcal{A}}. \tag{38}$$

It suffices to prove that there exists a classical adversary $\mathcal{B}$ that makes $p(Q_{\mathcal{A}}, m, \epsilon_{\mathcal{A}}^{-1}, \delta_{\mathcal{A}}^{-1})$ quantum queries and satisfies

$$\Pr_{H\xleftarrow{\$}\mathsf{Func}(\{0,1\}^n,\{0,1\}^m)}[\mathsf{Verify}^H(1^\lambda, \pi) = \top : \pi \xleftarrow{\$} \mathcal{B}^H(1^\lambda)] \geq \delta_{\mathcal{A}}/4$$

for some polynomial p. Let $M := \lceil \frac{4}{\epsilon_{\mathcal{A}}} \rceil$. For $i \in [M]$, we consider a quantum adversary $\mathcal{A}_i$ that works as follows.

[22] This idea is inspired by [CCY20].

$\mathcal{A}_i^H(1^\lambda)$**:** It runs $\{P_\pi\}_{\pi\in\{0,1\}^{\ell_\pi}} \xleftarrow{\$} \mathsf{Approx}(\mathcal{A}, \frac{\epsilon_{\mathcal{A}}}{4M}, \frac{1}{5})$ where ℓ_π is the length of a proof. Then it outputs the smallest π in the lexicographical order that satisfies

$$\mathsf{Verify}^H(1^\lambda, \pi) = \top$$

and

$$P_\pi > \frac{\epsilon}{2}\left(1 + \frac{2i-1}{2M}\right).$$

The number of queries by $\mathcal{A}_i$ is $Q_{\mathcal{A}_i} = \mathsf{poly}(\lambda, Q_{\mathcal{A}}, \epsilon_{\mathcal{A}}^{-1})$ since $\ell_\pi = \mathsf{poly}(\lambda)$. For each H, let π_i^H be the most lilkely output of $\mathcal{A}_i^H(1^\lambda)$.[23] We prove the following claim.

Claim 8.6. *For at least* $\left(\frac{\delta_{\mathcal{A}}}{2}\right)$*-fraction of* $H \in \mathsf{Func}(\{0,1\}^n, \{0,1\}^m)$ *and* $i \in [M]$*, it holds that*

$$\Pr[\mathcal{A}_i^H(1^\lambda) \to \pi_i^H] > 4/5.$$

Proof of Claim 8.6. By Equation (38), at least $\delta_{\mathcal{A}}$-fraction of H satisfies

$$\max_{\pi^* \text{ s.t. } \mathsf{Verify}^H(1^\lambda,\pi^*)=\top} \Pr[\mathcal{A}^H(1^\lambda) \to \pi^*] > \epsilon_{\mathcal{A}}. \tag{39}$$

Fix such H. Then, for at least $\frac{1}{2}$-fraction of $i \in [M]$, there does not exist π satisfying

$$\left|\Pr[\mathcal{A}^H(1^\lambda) \to \pi] - \frac{\epsilon_{\mathcal{A}}}{2}\left(1 + \frac{2i-1}{2M}\right)\right| < \frac{\epsilon_{\mathcal{A}}}{4M}. \tag{40}$$

This can be seen by a simple counting argument. First, we remark that if π satisfies Equation (40) for some $i \in [M]$, then we have $\Pr[\mathcal{A}^H(1^\lambda) \to \pi] > \epsilon_{\mathcal{A}}/2$. Therefore, the number of such π is at most $2/\epsilon_{\mathcal{A}}$. Second, we remark that each π can satisfy Equation (40) for at most one i. Therefore, the fraction of $i \in [M]$ such that there is π that satisfies Equation (40) is at most $2/(\epsilon_{\mathcal{A}} M) \le 1/2$.

Therefore, for at least $\left(\frac{\delta_{\mathcal{A}}}{2}\right)$-fraction of H and i, Equation (39) holds and there does not exist π satisfying Equation (40). For such H and i, if $\mathsf{Approx}(\mathcal{A}, \frac{\epsilon}{4M}, \frac{1}{5})$ succeeds, which occurs with probability at least $\frac{4}{5}$, then $\mathcal{A}_i$ outputs the smallest π in the lexicographical order that satisfies

$$\mathsf{Verify}^H(1^\lambda, \pi) = \top$$

and

$$\Pr[\mathcal{A}^H(1^\lambda) \to \pi] > \frac{\epsilon_{\mathcal{A}}}{2}\left(1 + \frac{2i-1}{2M}\right).$$

Since the above π is output with probability larger than $4/5$, this is the most likely output π_i^H. Thus, for at least $\left(\frac{\delta_{\mathcal{A}}}{2}\right)$-fraction of H and i, $\mathcal{A}_i^H$ returns π_i^H with probability larger than $4/5$. This completes the proof of Claim 8.6. □

For $j \in [\ell_\pi]$, let $\mathcal{A}_{i,j}$ be the algorithm that runs $\mathcal{A}_i$ and outputs the j-th bit of the output of $\mathcal{A}_i$. Since $\mathcal{A}_{i,j}$ makes the same number of queries as $\mathcal{A}_i$, its number of queries is $Q_{\mathcal{A}_{i,j}} = Q_{\mathcal{A}_i} = \mathsf{poly}(\lambda, Q_{\mathcal{A}}, \epsilon_{\mathcal{A}}^{-1})$. We apply Conjecture 8.1 to $\mathcal{A}_{i,j}$ where $\epsilon := 1/5$ and $\delta := \frac{\delta_{\mathcal{A}}}{4\ell_\pi}$. Then, Conjecture 8.1 ensures that there exists a deterministic classical algorithm $\mathcal{B}_{i,j}$ that makes $\mathsf{poly}(Q_{\mathcal{A}_{i,j}}, m, \epsilon^{-1}, \delta^{-1}) = \mathsf{poly}(\lambda, Q_{\mathcal{A}}, \epsilon_{\mathcal{A}}^{-1}, \delta_{\mathcal{A}}^{-1})$ classical queries and satisfies

$$\Pr_{H\xleftarrow{\$}\mathsf{Func}(\{0,1\}^n,\{0,1\}^m)}\left[\left|\Pr[\mathcal{A}_{i,j}^H(1^\lambda) \to 1] - \mathcal{B}_{i,j}^H(1^\lambda)\right| \le 1/5\right] \ge 1 - \frac{\delta_{\mathcal{A}}}{4\ell_\pi}.$$

[23] If there is a tie, we choose the smallest one in the lexicographical order.

By the union bound, we have

$$\Pr_{H\xleftarrow{\$}\mathsf{Func}(\{0,1\}^n,\{0,1\}^m)}\left[\forall j\in[\ell_\pi]\ \left|\Pr[\mathcal{A}^H_{i,j}(1^\lambda)\to 1]-\mathcal{B}^H_{i,j}(1^\lambda)\right|\le 1/5\right]\ge 1-\frac{\delta_{\mathcal{A}}}{4}. \tag{41}$$

Now, we give the classical adversary $\mathcal{B}$.

$\mathcal{B}^H(1^\lambda)$**:** It randomly chooses $i\xleftarrow{\$}[M]$. For $j=1,2,...,\ell_\pi$, it runs $\mathcal{B}^H_{i,j}(1^\lambda)$ and sets $\pi_j:=1$ if the output is larger than $1/2$ and $\pi_j:=0$ otherwise. Then it outputs $\pi:=\pi_1||\pi_2||...||\pi_{\ell_\pi}$.

By the construction, we can see that $\mathcal{B}$ makes $\mathsf{poly}(\lambda,Q_{\mathcal{A}},\epsilon_{\mathcal{A}}^{-1},\delta_{\mathcal{A}}^{-1})$ queries. By combining Claim 8.6 and Equation (41), for at least $\left(\frac{\delta_{\mathcal{A}}}{4}\right)$-fraction of $H\in\mathsf{Func}(\{0,1\}^n,\{0,1\}^m)$ and $i\in[M]$, if the j-th bit of π_i^H is 1, we have

$$\forall\ j\in[\ell_\pi],\ \mathcal{B}^H_{i,j}(1^\lambda)>3/5$$

and otherwise

$$\forall\ j\in[\ell_\pi],\ \mathcal{B}^H_{i,j}(1^\lambda)<2/5.$$

Since we have $\mathsf{Verify}^H(1^\lambda,\pi_i^H)=\top$ for all $i\in[M]$, we have

$$\Pr_{H\xleftarrow{\$}\mathsf{Func}(\{0,1\}^n,\{0,1\}^m)}[\mathsf{Verify}^H(1^\lambda,\pi)=\top:\pi\xleftarrow{\$}\mathcal{B}^H(1^\lambda)]\ge\frac{\delta_{\mathcal{A}}}{4}.$$

This contradicts the soundness of the proof of quantumness. This completes the proof of Theorem 8.4. □

References

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 333–342. ACM Press, June 2011.

[AA14] Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. *Theory Comput.*, 10:133–166, 2014.

[Adl79] Leonard Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 55–60, 1979.

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 255–268. ACM Press, June 2020.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997.

[BBC+98] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *39th FOCS*, pages 352–361. IEEE Computer Society Press, November 1998.

[BBS09] László Babai, Robert Beals, and Ákos Seress. Polynomial-time theory of matrix groups. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 55–64. ACM Press, May / June 2009.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.

[BJS10] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467:459 – 472, 2010.

[BKVV20] Zvika Brakerski, Venkata Koppula, Umesh V. Vazirani, and Thomas Vidick. Simpler proofs of quantumness. In *TQC 2020*, volume 158 of *LIPIcs*, pages 8:1–8:14, 2020.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[BV93] Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. In *25th ACM STOC*, pages 11–20. ACM Press, May 1993.

[CCD+03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *35th ACM STOC*, pages 59–68. ACM Press, June 2003.

[CCY20] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 181–206. Springer, Heidelberg, November 2020.

[CDGS18] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. Random oracles and non-uniformity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 227–258. Springer, Heidelberg, April / May 2018.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CGLQ20] Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. Tight quantum time-space tradeoffs for function inversion. In *61st FOCS*, pages 673–684. IEEE Computer Society Press, November 2020.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 1–29. Springer, Heidelberg, December 2019.

[dBCW02] J. Niel de Beaudrap, Richard Cleve, and John Watrous. Sharp quantum versus classical query complexity separations. *Algorithmica*, 34(4):449–461, 2002.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 356–383. Springer, Heidelberg, August 2019.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Inf. Theory*, 54(1):135–150, 2008.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of reed-solomon and algebraic-geometry codes. *IEEE Trans. Inf. Theory*, 45(6):1757–1767, 1999.

[Hal02] Sean Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. In *34th ACM STOC*, pages 653–658. ACM Press, May 2002.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HIOS15] Iftach Haitner, Yuval Ishai, Eran Omri, and Ronen Shaltiel. Parallel hashing via list recoverability. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 173–190. Springer, Heidelberg, August 2015.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147, 1995.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, Heidelberg, April / May 2018.

[Kra03] Victor Yu. Krachkovsky. Reed-solomon codes for correcting phased error bursts. *IEEE Trans. Inf. Theory*, 49(11):2975–2984, 2003.

[KYY18] Shuichi Katsumata, Shota Yamada, and Takashi Yamakawa. Tighter security proofs for GPV-IBE in the quantum random oracle model. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 253–282. Springer, Heidelberg, December 2018.

[Lin10] Yehuda Lindell. Introduction to coding theory lecture notes, 2010. https://u.cs.biu.ac.il/~lindell/89-662/coding_theory-lecture-notes.pdf.

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 326–355. Springer, Heidelberg, August 2019.

[MV08] Michael Mitzenmacher and Salil P. Vadhan. Why simple hash functions work: exploiting the entropy in a data stream. In Shang-Teng Huang, editor, *19th SODA*, pages 746–755. ACM-SIAM, January 2008.

[Pad06] Sahadeo Padhye. A Public Key Cryptosystem Based On Pell Equation. Cryptology ePrint Archive, Report 2006/191, 2006. https://eprint.iacr.org/2006/191.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Rud07] Atri Rudra. *List Decoding and Property Testing of Error Correcting Codes*. PhD thesis, University of Washington, 8 2007.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, oct 1997.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 520–551. Springer, Heidelberg, April / May 2018.

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216. Springer, Heidelberg, October / November 2016.

[vDHI06] Wim van Dam, Sean Hallgren, and Lawrence Ip. Quantum algorithms for some hidden shift problems. *SIAM J. Comput.*, 36(3):763–778, 2006.

[Yue14] Henry Yuen. A quantum lower bound for distinguishing random functions from random permutations. *Quantum Info. Comput.*, 14(13–14):1089–1097, oct 2014.

[YZ21] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 568–597. Springer, Heidelberg, October 2021.

[Zha12] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, August 2012.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Info. Comput.*, 15(7–8):557–567, may 2015.