Scalable Pseudorandom Quantum States*

Zvika Brakerski[1] and Omri Shmueli[2]

[1] Weizmann Institute of Science**
[2] Tel-Aviv University***

Abstract. Efficiently sampling a quantum state that is hard to distinguish from a truly random quantum state is an elementary task in quantum information theory that has both computational and physical uses. This is often referred to as pseudorandom (quantum) state generator, or PRS generator for short.

In existing constructions of PRS generators, security scales with the number of qubits in the states, i.e. the (statistical) security parameter for an n-qubit PRS is roughly n. Perhaps counter-intuitively, n-qubit PRS are not known to imply k-qubit PRS even for $k < n$. Therefore the question of *scalability* for PRS was thus far open: is it possible to construct n-qubit PRS generators with security parameter λ for all n, λ. Indeed, we believe that PRS with tiny (even constant) n and large λ can be quite useful.

We resolve the problem in this work, showing that any quantum-secure one-way function implies scalable PRS. We follow the paradigm of first showing a *statistically* secure construction when given oracle access to a random function, and then replacing the random function with a quantum-secure (classical) pseudorandom function to achieve computational security. However, our methods deviate significantly from prior works since scalable pseudorandom states require randomizing the amplitudes of the quantum state, and not just the phase as in all prior works. We show how to achieve this using Gaussian sampling.

1 Introduction

Quantum mechanics asserts that the state of a physical system is characterized by a vector in complex Hilbert space, whose dimension corresponds to the number of degrees of freedom of the system. Specifically, a system with 2^n possible degrees of freedom (such as an n-qubit system, the quantum analogue to an n

* The full version of this paper is available at https://arxiv.org/abs/2004.01976.

** Email: zvika.brakerski@weizmann.ac.il. Supported by the Binational Science Foundation (Grant No. 2016726), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and via Project PROMETHEUS (Grant 780701).

*** Email: omrishmueli@mail.tau.ac.il. Supported by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482), by the Israel Science Foundation Grant No. 18/484, and by Len Blavatnik and the Blavatnik Family Foundation.

bit system) is represented as a unit vector over $\mathbb{C}^{2^n}$. The ability to sample a random state of a system is a fundamental task when attempting to provide a computational description of the physical world.

Since the description length of a quantum state is infinite (and very long even when taken to a finite precision), relaxed notions for random state sampling are considered in the literature. Most commonly (and in this work) we consider restricting the *number of copies* of the sampled state that are given to the adversary.[3] The notion of quantum t-designs [AE07] considers computationally unbounded adversaries that are given t copies of the sampled state, and the requirement is that this input is (statistically) indistinguishable from t copies of a true random state. The resources of generating t-designs scale at least linearly with t, and therefore if efficient generation is sought, t designs can only be constructed for polynomial t.[4] Recently, a computational variant known as Pseudorandom Quantum State (PRS) was proposed by Ji, Liu and Song [JLS18]. In a PRS, the adversary is allowed to request an a-priori unbounded polynomial number of samples t, but the guarantee of indistinguishability only holds against *computationally bounded* adversaries. PRS have applications in quantum-cryptography (e.g. quantum money [JLS18]) and computational physics (e.g. simulation of thermalized quantum states [PSW06]).

It was shown in [JLS18,BS19] that PRS can be constructed from any quantum-secure one-way function. The design paradigm in both works is as follows. First, assume you are given (quantum) oracle access to a (classical) random function, and show how to efficiently construct a PRS which is secure even against computationally unbounded adversaries, a notion that [BS19] calls Asymptotically Random State (ARS). Then, replace the random function with a post-quantum pseudorandom function (PRF) to obtain computational security. Since only a fixed number of calls to the PRF is required in order to generate each PRS copy, this paradigm also leads to new constructions of t-designs, as observed in [BS19].

The previous works [JLS18, BS19] showed how to construct an n-qubit PRS, which is secure against any $\mathrm{poly}(n)$ time adversary. To be more precise, they constructed ARS whose distinguishing advantage is bounded by $4t^2 \cdot 2^{-n}$, and converted it into a PRS using a PRF as described above. We can therefore say that the *statistical security parameter* of the scheme is (essentially) n, and there is an additional computational security parameter that comes from the hardness of the PRF. Indeed, a security parameter of n seems quite sufficient since the complexity of the construction is $\mathrm{poly}(n)$ so it is possible to choose n as large as needed in order to provide sufficient security. Alas it is not possible to convert an n-qubit state generator into one that produces a random state over a smaller

[3] Recall that in the quantum setting, due to the no-cloning property, providing additional copies of the same state allows to recover more information about it. In utmost generality, any additional copy provides additional information, and a complete recovery of a quantum state requires infinitely many copies.

[4] As usual, we use the notion of security parameter λ that indicates the power of honest parties and of adversaries. We assume that honest parties run in time $\mathrm{poly}(\lambda)$ for a *fixed* polynomial, whereas the advantage of the adversary needs to scale super-polynomially, and preferably exponentially, with λ.

number of qubits, say $k < n$. This may be quite surprising as one would imagine that we can simply generate an n-qubit state, and just take its k-qubit prefix. However, recall that the n-qubits are in superposition, and taking a prefix is equivalent to measurement of the remaining $(n-k)$ qubits. For each of the t copies, this measurement has a different outcome and therefore each of the t copies will produce a different k-qubit states, as opposed to t copies of the same state as we wanted.

This peculiar state of affairs means that prior to this work it was not known, for example, how to construct ARS/PRS of n qubits, but with adversarial advantage bounded by 2^{-2n}. This issue is also meaningful when considering the concrete (non-asymptotic) security guarantees of PRS, where we wish to obtain for example 128 bits of security against an adversary that obtains at most 2^{20} copies of a PRS over 70 qubits.

This Work: Scalable ARS/PRS. In this light, it is desirable to introduce ARS/PRS constructions where the security parameter is in fact a parameter which is tunable independently of the length of the generated state. We call this notion *scalable* ARS/PRS. We notice that the approaches of [JLS18, BS19] are inherently not scalable since they can only generate states in which all computational-basis elements have the same amplitude, and the randomness only effects the phase. Such vectors are inherently distinguishable from uniform unless the dimension is very large (hence their dependence between length and security). In this work, we present new techniques for constructing ARS/PRS and in particular present a scalable construction under the same cryptographic assumptions as previous works.

1.1 Our Results

Our main technical result, as in all previous works, is concerned with constructing an ARS generator which is efficient given oracle access to a random function.[5]

Lemma 1.1 (Main Technical Lemma). *There exists a scalable ARS generator.*

Furthermore, for every length n of a quantum state and security parameter λ, running the generator t times (for any t) produces an output distribution that is $O\left(\frac{t}{e^{\lambda}}\right)$-indistinguishable from t copies of a random quantum state of n qubits.

We note that in previous works that construct ARS generators [JLS18, BS19] the dependence on t in the bound on the trace distance is quadratic, that is, previous ARS generators are known to achieve a bound of $\frac{t^2}{2^n}$ on the trace distance between t-copies of the ARS and a random quantum (n-qubit) state, whereas in this work the trace distance bound only scales up linearly with t.

As immediate corollaries and similarly to [JLS18, BS19], we derive the existence of a scalable PRS generator (assuming post-quantum one-way functions)

[5] Note that this is not the quantum random oracle model since the random oracle is "private" and the adversary does not get access to it.

and scalable t-design generators (unconditionally). Unlike scalable PRS generators, scalable state t-design generators were known to exist before this work, however their depth was known to scale up linearly with t (and polynomially in n), and in our construction the depth scales logarithmically with t (and polynomially in n, λ).

Corollary 1.2. *If post-quantum one-way functions exist, then scalable PRS generators exist.*

Corollary 1.3. *For any polynomial $t(\cdot) : \mathbb{N} \to \mathbb{N}$, scalable state $t(\lambda)$-design generators exist where the circuit depth is* $\mathrm{poly}(n, \lambda, \log t)$.

Our ARS construction requires a random oracle with n bits of input (where n is the length of the generated state) and $\mathrm{poly}(\lambda)$ bits of output, it therefore follows that if $n = O(\log \lambda)$, then it is possible to instantiate the construction with a completely random string of length $2^n \cdot \mathrm{poly}(\lambda) = \mathrm{poly}(\lambda)$, and obtain statistically secure PRS. We view this consequence as not very surprising in hindsight.

Recently Alagic, Majenz and Russell [AMR19] proposed the notion of random state simulators. Simulators are stateful, and their local state grows with the number of copies t, however, there is no a-priori bound on the number of copies that the simulator can produce, and the guarantee is information-theoretic rather than computational. One can observe that a scalable ARS generator also implies efficient state simulators, by using the random-oracle simulation technique of Zhandry [Zha19]. The state simulators of [AMR19] follow a different approach, which is not known to imply ARS, and achieve simulators with perfect security (and thus straightforwardly scalable), but our ARS provides a different avenue for scalable random quantum state simulators as well.

1.2 Paper Organization

We provide a detailed technical overview of our results in Section 2. Preliminaries appear in Section 3, and in particular we formally state the derivation of the corollaries from the main theorem (which were implicit in previous work) in Section 3.3. Our technical results are presented in the following two sections. In Section 4 we present quantum information-theoretic tools which are required for our construction but may also find other uses. Then Section 5 contains our actual construction.

Acknowledgments

We wish to thank the TCC 2019 reviewer of [BS19] who brought the scalability problem in existing PRS constructions to our attention. We thank Crypto 2020 reviewers for their insightful comments on a prior version of this manuscript.

2 Technical Overview

We now provide a technical outline of how we achieve our main result in Lemma 1.1. Deriving the corollaries is straightforward using known techniques.[6]

As Lemma 1.1 states, we design an algorithm that has oracle access to a random function f, takes as input a bit length n and a security parameter λ, runs in time $\mathrm{poly}(n, \lambda)$, and produces a quantum state over n-qubits $|\psi_{f,n,\lambda}\rangle$ (note that even though our algorithm is randomized, it can either output the state $|\psi_{f,n,\lambda}\rangle$ or $\perp$ and will never output the "wrong" state). It furthermore holds that the distribution that samples a random function f and outputs $|\psi_{f,n,\lambda}\rangle^{\otimes t}$ (i.e. t copies of the state $|\psi_{f,n,\lambda}\rangle$), is within trace distance at most $\mathrm{poly}(t)/2^{\lambda}$ from the distribution that produces t copies of a truly randomly sampled n-qubit state.

We recall the standard Dirac notation for vectors in Hilbert space. An n-qubit state is generically denoted by a unit vector in $\mathbb{C}^{2^n}$ of the form $|\alpha\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$. Throughout this overview we wish to refer to normalized as well as non-normalized vectors. We will use the convention that a vector $|\alpha\rangle$ is not necessarily normalized unless explicitly noted that it represents a quantum state (or a unit vector), and will denote its normalization

$$|\hat{\alpha}\rangle = \sum_{x\in\{0,1\}^n} \hat{\alpha}_x|x\rangle := \frac{1}{\sqrt{\langle\alpha|\alpha\rangle}} \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle \ ,$$

where $\langle\alpha|\alpha\rangle = \sum_x |\alpha_x|^2$.

As explained above, prior works generated quantum states where in the standard basis all coefficients had the same amplitude, i.e. their ARS could be represented by $|\alpha\rangle$ s.t. $|\alpha_x| = 1$ for all x. We abandon this approach, which as we explained cannot lead to a scalable ARS construction. Instead, we will show how to interpret a random function f as an implicit representation of a random unit vector in $\mathbb{C}^{2^n}$. Moreover, we want this interpretation to be *locally computable* in the sense that the value α_x only depends on $f(x)$. Our approach, therefore, is more direct and also more involved than the approach taken in previous works, since we will try to sample from a space that most closely resembles the uniform distribution over quantum states.

2.1 Our Approach: Implicit Random Gaussian Vector

Assume that we had an efficiently computable classical function $g(\cdot)$ s.t. if we set $v_x = g(f(x))$ and consider the vector $|v\rangle = \sum_x v_x|x\rangle$, then the distribution on $|v\rangle$ (induced by sampling the function f randomly) is *spherically symmetric*, i.e. invariant to unitary transformations ("rotations" in $\mathbb{C}^{2^n}$). In this case, the normalized vector $|\hat{v}\rangle$ is a uniform unit vector. In other words, we will show how

[6] We note that this standard transition from ARS with oracle to PRS and to t-designs was not formally stated in its generic form in previous works. In this work we also provide the generic derivations in Section 3.3.

to use the random function f as an implicit representation of a vector $|v\rangle$ such that for all x, v_x can be efficiently *locally* computed given x (and oracle access to f).

Our solution, therefore, needs to address two challenges. The first is to properly define a locally efficiently computable function g with the desirable properties. The second is to efficiently generate the quantum state $|\hat{v}\rangle$ given oracle access to the values v_x. Let us describe how we handle each one of these challenges at a high level, and then expand on the parts that contain the bulk of technical novelty.

First Technique: Multivariate Gaussian Sampling. For the first challenge, we use the multivariate Gaussian distribution, whose spherical symmetry has proven useful for many applications in the literature. Our function g will simply be a Gaussian sampler (or more accurately, a two-dimensional Gaussian sampler, for the real and imaginary parts of v_x). That is, we use the entries of the random function f as random tape for a Gaussian sampling procedure g. Since the Gaussian distribution is spherically symmetric, such a g has the properties that we need.

This approach indeed seems quite suitable but achieving (perfect) spherical symmetry is at odds with achieving computational efficiency, simply because the Gaussian distribution is continuous and has infinite support. Indeed, we will need to show a truncated discretized Gaussian distribution which on one hand can be sampled efficiently, and on the other hand provides approximate spherical symmetry. Note that the notion of approximation we are interested in here is with respect to the trace distance between the quantum state $|\hat{v}\rangle^{\otimes t}$ and a t-repetition of a random unit vector. This requires us to develop tools in order to relate this notion to standard notions such as Euclidean distance. These tools are not particularly complicated but we view them as fundamental and of potential to be used elsewhere.[7] We elaborate more on this in Section 2.2 below, and the full details appear in Section 4.

Second Technique: Rejection Sampling. The second challenge is addressed using a quantum analog of the *rejection sampling* technique. Recall that in standard probability theory, if it is possible to sample from a distribution p where $\Pr[x] = p_x$, then we can consider the experiment of first sampling from p, and then either outputting the sample received x with probability q_x, or aborting and restarting the process with probability $1 - q_x$. This process constitutes a sampler for the distribution $\frac{p_x q_x}{\sum_x p_x q_x}$. The probability of not aborting is $\sum_x p_x q_x$, and therefore the expected running time of the new sampler is $\frac{1}{\sum_x p_x q_x}$. In the quantum setting, a similar technique can be used for superpositions (Indeed, extensions of these technique were used e.g. in [ORR13]).

In this work we use quantum rejection sampling to generate quantum states from scratch. To create our state $|v\rangle$ we will start with the uniform superposition $|u\rangle = \sum_x |x\rangle$, and via a rejection process we can obtain (not necessarily with

[7] We will not be surprised if they were already discovered and used in the literature, but we were unable to find a relevant reference.

good pobability), any desired superposition $|v\rangle$. The probability of success in the quantum case is $\frac{1}{d^2} \cdot \frac{\langle v|v\rangle}{\langle u|u\rangle}$, where d is an *a-priori* bound on $\max_x |v_x|$ that needs to be given as a parameter to the rejection sampling procedure. (The algorithm and success probability are analogous to the classical version described above, when replacing q_x with $\frac{v_x}{d}$ and considering ℓ_2 norm instead of ℓ_1.)

On the face of it, the rejection sampling procedure can work to create any state $|v\rangle$ when a bound d is known. However, the probability of success can still be very small (e.g. negligible), so if we wish to use repetition to obtain $|v\rangle$, the expected running time will become very large (e.g. super-polynomial). Fortunately, our vectors $|v\rangle$ are (approximately) Gaussian, which means that they have strong concentration properties that guarantee that with high probability two properties are satisfied. The first is that all entries v_x have roughly the same magnitude, up to a factor of $\mathrm{poly}(n, \lambda)$.[8] This allows us to choose the value d in such a way that the rejection sampling algorithm will operate correctly. The second property is that $\langle v|v\rangle \approx 2^n$ (formally, $\langle v|v\rangle$ is a constant factor away from 2^n), this makes the probability of success noticeable (i.e. $1/\mathrm{poly}(n, \lambda)$). We informally call a vector that maintains the combination of these two properties "balanced". By running in time $\mathrm{poly}(n, \lambda)$ and repeating the process as needed we can amplify the success probability to $1 - 2^{-\lambda}$. We generalize these properties and provide a state generator for any oracle v_x which satisfied the balance property, see Section 4.

Lastly, we note that while the first property above (bound on d) can be made to hold for any n, the second one (lower bound on $\langle v|v\rangle$) might not hold with high enough probability. Special care needs to be taken in the case where n is very small, since in that case concentration properties are insufficient to imply that $\langle v|v\rangle$ does not fall far below its expected value with small yet significant probability (we wish to succeed with all but $2^{-\lambda}$ probability, so anything higher than that is already significant). In such a case, the success probability of the rejection sampler might become negligibly small, which will lead to failure in generating a state.[9] Luckily, since the dimension of the vector $|v\rangle$ is 2^n, good concentration kicks in already at $n \gtrsim \log(\lambda)$, so we only need to worry about this issue when $n < \log(\lambda)$. For such small n, the sampling algorithm can store the vector $|v\rangle$ in its entirety, and check whether the norm $\langle v|v\rangle$ is sufficiently close to its expectation (which happens with constant probability). If the norm is not in the required range, we sample a new Gaussian.[10] Repeating this roughly λ times

[8] Note that, e.g. tail bounds on the norm of a Gaussians asserts that the probability that its amplitude is beyond k times standard deviation is at most $e^{-c \cdot k^2}$ for some constant c. This means that if we want to find a tail bound that applies to all 2^n components of the vector $|v\rangle$ at the same time via union bound, it suffices to use $k \approx \sqrt{n + \lambda}$.

[9] We stress again that if the success probability becomes negligible with only negligible probability, e.g. $2^{-\sqrt{\lambda}}$, this is still a problem since the state generator will simply fail with this probability and therefore we cannot hope to be $2^{-\lambda}$ close to uniform.

[10] Recall that we think of the values of the function $f(x)$ as the random tape of a Gaussian sampler g. We can consider a function f with output length which is λ

guarantees that we generate a "balanced" vector from a spherically symmetric distribution with all but $2^{-\lambda}$ probability.

2.2 Approximate Gaussians Under Tensored Trace Distance

We wish to do approximate sampling from the continuous Gaussian distribution using an efficiently locally sampleable distribution. If we wish to be fully precise, we need to consider Gaussian distributions over the complex regime. However, for the purpose of sampling, one can think of each complex coordinate just as two real-valued coordinates. For the purpose of this overview we will simplify things even further and assume that we wish to sample from a real-valued Gaussian, i.e. a vector in $\mathbb{R}^{2^n}$ instead of $\mathbb{C}^{2^n}$. Everything we discuss here be extended to the complex regime in a natural manner. From this point and on, our goal is to find an efficient sampler g s.t. when sampling v_x i.i.d from the distribution generated by g, and sampling w_x from a continuous Gaussian, it holds that the trace distance (quantum optimal distinguishing probability) between the quantum states $|\hat{v}\rangle^{\otimes t}$ and $|\hat{w}\rangle^{\otimes t}$, is at most $\mathrm{poly}(t) \cdot 2^{-\lambda}$ for all t. For any vectors $|v\rangle, |w\rangle$, we refer to the trace distance between $|\hat{v}\rangle^{\otimes t}$ and $|\hat{w}\rangle^{\otimes t}$ as the "t-tensored trace distance" between $|v\rangle$ and $|w\rangle$.

An efficiently sampleable distribution is necessarily discrete and supported over a finite segment, whereas the Gaussian distribution is continuous and supported over $(-\infty, \infty)$. Indeed, even in the classical setting Gaussian samplers need to handle this discrepancy. Usually, when one says that it is efficient to sample from the Gaussian distribution, they mean that it is possible to sample to within any polynomial precision and from a Gaussian truncated far enough away from the standard deviation that the probability mass that is chopped off is negligible.[11] We adopt a similar approach here. Formally, sampling to within a fixed precision is equivalent to sampling from a *rounded* Gaussian distribution, i.e. the distribution obtained by sampling from a continuous Gaussian and then rounding the result to the nearest multiple of ϵ, where ϵ indicates the required precision. Truncation means that we sample from the distribution obtained by sampling a Gaussian, and if the absolute value of the sampled value x is at most some bound B, then return x, otherwise return 0. Setting B to be sufficiently larger than the standard deviation, say by roughly a factor of k, would imply that the resulting distribution only distorts the Gaussian by e^{-k^2} in total variation distance. We set our sampler g therefore to be a sampler from the B-truncated ε-rounded Gaussian distribution. It is possible to sample from a distribution that's within ε statistical distance from this distribution in time $\mathrm{poly}(\log(1/\epsilon), \log(B))$ by standard Gaussian sampling techniques, and therefore we can set $1/\epsilon$ to be a sufficiently large exponential function in λ, n and maintain the efficient sampling property.

---

times the number of random bits used by the sampler g, so that we have sufficient randomness to re-run g as needed.

[11] An alternative to chopping the ends of the distribution is to construct a sampler that runs only in expected polynomial time and might run for a very long superpolynomial time with small probability. This approach is less suitable for our purposes.

The challenge, as already mentioned above, is to translate this intuitive notion of "approximate Gaussian" to one that is provable under tensored trace distance. In fact, we present a general analysis of the effects of truncation and rounding on tensored trace distance. We do this using a two-phase proof.

Part I: Tensored Trace Distance Respects Statistical Distance. We show that truncating a continuous Gaussian introduces negligible trace distance for *any* number of copies t. This follows quite straightforwardly from the classical total variation distance bound between the distributions. In fact, we show a more general claim (Lemma 4.3): Let $|v\rangle$ and $|w\rangle$ be distributions over n-qubit states, such that their classical distributions as 2^n-dimensional vectors are within classical statistical distance (total variation distance) δ. Then their t-tensored trace distance is at most δ for all t. The intuition here (which can also be translated to a formal proof), is that even given an infinite number of repetitions, a quantum state does not contain more information than its 2^n-dimensional coefficient vector. Therefore, a (computationally unbounded) adversary that attempts to distinguish $|\hat{v}\rangle^{\otimes t}$ and $|\hat{w}\rangle^{\otimes t}$ as quantum states cannot do better than a classical (computationally unbounded) adversary which receives $|v\rangle, |w\rangle$ as explicit vectors.

Part II: Tensored Trace Distance Respects Rounding. We say that a distribution $|v\rangle$ is a rounding of a distribution $|w\rangle$ if $|v\rangle$ can be described as first sampling an element from $|w\rangle$ and then applying some mapping φ s.t. for all w, $\|\varphi(w) - w\|$ is bounded (say be some value δ).[12] We wish to show that if $|v\rangle$ is a rounding of $|w\rangle$ then these vectors are close under tensored trace distance.

Let us start by considering the case $t = 1$, i.e. the distinguisher needs to distinguish between the quantum states $|\hat{v}\rangle$ and $|\hat{w}\rangle$. It is well established that if $|\hat{v}\rangle$ and $|\hat{w}\rangle$ are close in Euclidean distance, then they are also close in trace distance. However, this does not complete the proof since we only have a bound on the Euclidean distance between the unnormalized vectors $|v\rangle$ and $|w\rangle$. Indeed, the notion we care about is the Euclidean distance when projected onto the unit sphere, or in other words the *angular* distance induced by φ. In our case, our distribution $|w\rangle$ (the Gaussian) is such that the norm is quite regular with high probability, and this is preserved also for the rounded version (some straightforward yet fairly elaborate calculation is required in order to establish the exact parameters).[13]

Once we formalize the right notion of approximation (i.e. angular distance), it is possible to state a general lemma (Lemma 4.4) that shows that if φ is s.t. the angular distance between its input and output (over the support of $|v\rangle$) is bounded, then the t-tensored trace distance degrades moderately with t.

[12] Note that we call this "rounding" but in general this can be applied in other situations.

[13] This introduces an additional layer of complication into our proof, as we will need to apply the rounding tool to a restriction of the Gaussian distribution for which the norm is well behaved. Since the "regular norm" variant is close in statistical distance to the standard Gaussian, this can be handled by our first technique above.

Therefore, if we start with a short enough angular distance, our trace distance will indeed be bounded by $\text{poly}(t)/2^{\lambda}$.

3 Preliminaries

3.1 Standard Notions and Notations

During this paper we use standard notations from the literature. For $n \in \mathbb{N}$,

- We denote $[n] := \{1, \cdots, n\}$.
- We denote by $[n]_2$ the $\lceil \log_2(n) \rceil$-bit binary representation of n.
- We denote by ω_n the complex root of unity of order n: $\omega_n := e^{\frac{2\pi i}{n}}$.
- We denote by $\mathcal{S}(n)$ the set of n-qubit pure quantum states, by $\mathcal{D}(n)$ the set of n-qubit mixed quantum states and by $\mathcal{U}(n)$ the set of n-qubit quantum unitary circuits.
- We sometimes denote 2^n with N, when we do that, we explicitly note it.

Vectors and Quantum States. We use standard Dirac notation throughout this paper, vectors are not assumed to be normalized unless explicitly mentioned. Specifically, for a column vector $u \in \mathbb{C}^m$, we denote $|u\rangle := u$, $\langle u| := u^{\dagger}$, where $u^{\dagger}$ is the conjugate transposed of u. We usually let $\hat{u}$ denote the normalized version of the vector u, namely: $\hat{u} := \frac{1}{\|u\|} \cdot u$ (where u is a nonzero complex vector). Vectors that represent quantum states have unit norm and therefore are normalized by default.

We make a distinction between a *vector* in a Hilbert space, and the *quantum state* corresponding to this vector. The two objects are related as a complete characterization of a (pure) quantum state over n-qubits is characterized by a vector in a 2^n-dimensional Hilbert space (up to normalization and global phase). However, the vector is not necessarily (and almost always is not) recoverable given the n-qubit state, and quantum states that correspond to different vectors can be indistinguishable (even perfectly).[14] In terms of vector notation, the symbol $|u\rangle$ can refer either to the vector in the Hilbert space of to the quantum state that corresponds to this vector, we will explicitly mention which of the two we refer to when using this notation.

Distributions Over Quantum States as Density Matrices. Density matrices are a mathematical tool to describe mixed quantum states, that is, distributions over quantum states. Formally, let μ a (possibly continuous) probability distribution over n-qubit quantum states, $\mu : \mathcal{S}(2^n) \to [0,1]$, $\int_{|\psi\rangle \in \mathcal{S}(2^n)} 1 \mathrm{d}\mu(|\psi\rangle) = 1$, then the density matrix induced by μ is denoted ρ_μ and defined as:

$$\rho_\mu = \mathbb{E}_{|\psi\rangle \leftarrow \mu}\Big[\big(|\psi\rangle\langle\psi|\big)\Big] := \int_{|\psi\rangle \in \mathcal{S}(2^n)} \big(|\psi\rangle\langle\psi|\big) \mathrm{d}\mu(|\psi\rangle) \ . \tag{1}$$

[14] Information theoretically, in the general case, one requires an infinite number of copies of a quantum state in order to precisely recover the vector in the Hilbert space that characterizes this state.

Statistical Distance. We use basic properties of the statistical distance metric (also known as total variation distance). Statistical distance can be described in terms of operations, that is, for two (possibly continuous) distributions D_1, D_2 with corresponding supports S_1, S_2, the statistical distance between D_1, D_2 is the maximal advantage,

$$\left| \Pr_{x \leftarrow D_1} [\mathsf{A}(x) = 1] - \Pr_{x \leftarrow D_2} [\mathsf{A}(x) = 1] \right|$$

taken over all functions $\mathsf{A} : S_1 \cup S_2 \to \{0, 1\}$. We note that we can allow A to be randomized and obtain an equivalent definition. The statistical distance between two random variables is the statistical distance between their associated distributions.

Additionally, throughout the proof of Theorem 5.1 we will use the following fact about the statistical distance between a distribution and a conditional version of it.

Fact 3.1 *Let X be a random variable and E some probabilistic event. Denote $Y = X|\bar{E}$, i.e. the conditional variable of X conditioned on E* not *happening. Then*

$$\mathrm{SD}(X, Y) \leq \Pr[E] \ .$$

Trace Distance. The trace distance, defined below, is a generalization of statistical distance to the quantum setting and represents the maximal distinguishing probability between distributions over quantum states.

Definition 3.2 (Trace Distance). *Let $\rho_0, \rho_1 \in \mathcal{D}(2^n)$ be two density matrices of n-qubit mixed states. For a projective measurement A with output in $\{0, 1\}$ define*

$$\Delta_{\mathsf{A},\rho_0,\rho_1} := \left| \Pr\left[\mathsf{A}\big(\rho_0\big) = 0\right] - \Pr\left[\mathsf{A}\big(\rho_1\big) = 0\right] \right| \ .$$

The trace distance between ρ_0, ρ_1 is

$$\mathrm{TD}(\rho_0, \rho_1) := \max_{\{0,1\} \ \textit{projective measurement}\ \mathsf{A}} \Delta_{\mathsf{A},\rho_0,\rho_1} \ .$$

We note that the trace distance is often equivalently defined as $\frac{1}{2} \left\| \rho_0 - \rho_1 \right\|_1$, where $\left\| \cdot \right\|_1$ refers to the ℓ_1 norm of the vector of eigenvalues of the operand matrix.

A standard fact about trace distance is the following.

Fact 3.3 *Let D_0, D_1 be two distributions over n-qubit states and let $\rho_0, \rho_1 \in \mathcal{D}(2^n)$ be the corresponding density matrices. For a projective measurement A with output in $\{-1, 1\}$ define*

$$\tilde{\Delta}_{\mathsf{A},\rho_0,\rho_1} := \left| \mathbb{E}_{\substack{|\psi\rangle \leftarrow D_0, \\ \textit{Measurement}}} \left[\mathsf{A}\big(|\psi\rangle\big)\right] - \mathbb{E}_{\substack{|\psi\rangle \leftarrow D_1, \\ \textit{Measurement}}} \left[\mathsf{A}\big(|\psi\rangle\big)\right] \right| \ .$$

Then,

$$2 \cdot \mathrm{TD}(\rho_0, \rho_1) = \max_{\{-1,1\} \ \textit{projective measurement}\ \mathsf{A}} \tilde{\Delta}_{\mathsf{A},\rho_0,\rho_1} \ .$$

The trace distance between pure states is given by the following expression.

Fact 3.4 *For n-qubit pure quantum states $|\psi\rangle, |\phi\rangle$, the trace distance between them is:*

$$\mathrm{TD}\Big(|\psi\rangle\langle\psi|, |\phi\rangle\langle\phi|\Big) = \sqrt{1 - |\langle\psi|\phi\rangle|^2} \ .$$

Trace distance is an operator on density matrices. In this work we will sometimes use it directly on distributions, that is we denote $\mathrm{TD}(D_1, D_2)$, where D_1, D_2 are distributions over n-qubit quantum states. This notation refers to the trace distance between the two density matrices induced by D_1 and D_2 (as per Eq. (1)). That is,

$$\mathrm{TD}(D_1, D_2) := \mathrm{TD}(\rho_{D_1}, \rho_{D_2}) = \mathrm{TD}\Bigg(\mathbb{E}_{|\psi\rangle\leftarrow D_1}\Big[\big(|\psi\rangle\langle\psi|\big)\Big] \ , \ \mathbb{E}_{|\psi\rangle\leftarrow D_2}\Big[\big(|\psi\rangle\langle\psi|\big)\Big]\Bigg) \ .$$

Quantum Unitary for a Classical Function. Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function. The unitary of f is denoted by U_f, it is a unitary over $n+m$ qubits defined as

$$\forall x \in \{0,1\}^n, y \in \{0,1\}^m : U_f|x, y\rangle := |x, y \oplus f(x)\rangle \ .$$

Quantum Rejection Sampling. Quantum Rejection Sampling (QRS) is a known efficient procedure for taking one quantum state $|\alpha\rangle$ and outputting with some probability a different quantum state $|\beta\rangle$, given black box access to a circuit that describes their closeness. Formally, the algorithm QRS gets as input an n-qubit quantum state $|\alpha\rangle$ and quantum oracle access to a unitary U on $n+k$ qubits (where k is related to the binary description length for complex numbers that is being used) and have the following correctness and time complexity guarantees.

Theorem 3.5 (Quantum Rejection Sampling). *Let $|\alpha\rangle, |\beta\rangle$ be two n-qubit quantum states and let U be an $(n+k)$-qubit unitary. Assume there exists a positive real number d such that the following hold*

- $d \geq \max_{x\in\{0,1\}^n} \left|\frac{\beta_x}{\alpha_x}\right|$.
- *$\forall x \in \{0,1\}^n$, the complex number $\frac{(\beta_x/\alpha_x)}{d}$ can be described with full precision in k bits.*
- *U is the unitary of the classical function $f : \{0,1\}^n \to \{0,1\}^k$ such that $f(x) := \frac{(\beta_x/\alpha_x)}{d}$.*

Then $\mathsf{QRS}^U(|\alpha\rangle)$ outputs $(\mathtt{success}, |\beta\rangle)$ with probability at least $\frac{1}{d^2}$ and otherwise outputs $(\mathtt{fail}, |0^n\rangle)$.

The algorithm makes a single query to U, and assuming this query takes a single time step, the time complexity of $\mathsf{QRS}^U(|\alpha\rangle)$ is $\mathrm{poly}(n,k)$.

3.2 Pseudorandom Functions and m-Wise Independent Functions

We define pseudorandom functions with quantum security (QPRFs).

Definition 3.6 (Quantum-Secure Pseudorandom Function (QPRF)). *Let $\mathcal{K} = \{\mathcal{K}_n\}_{n\in\mathbb{N}}$ be an efficiently samplable key distribution, and let $\mathsf{PRF} = \{\mathsf{PRF}_n\}_{n\in\mathbb{N}}$, $\mathsf{PRF}_n : \mathcal{K}_n \times \{0,1\}^n \to \{0,1\}^{\mathrm{poly}(n)}$ be an efficiently computable function, where $\mathrm{poly}(\cdot)$ is some polynomial. We say that PRF is a quantum-secure pseudorandom function if for every efficient non-uniform quantum algorithm $A = \{A_n\}_{n\in\mathbb{N}}$ (with quantum advice) that can make quantum queries there exists a negligible function $\mathrm{negl}(\cdot)$ s.t. for every $n \in \mathbb{N}$,*

$$\left| \Pr_{k\leftarrow\mathcal{K}_n}[A_n^{\mathsf{PRF}_k} = 1] - \Pr_{f\leftarrow(\{0,1\}^n)^{(\{0,1\}^n)}}[A_n^f = 1] \right| \leq \mathrm{negl}(n) \ .$$

In [Zha12], QPRFs were proved to exist under the assumption that post-quantum one-way functions exist.

We define m-wise independent functions as keyed functions s.t. when the key is sampled from the key distribution, then any m different inputs to the function generate m-wise independent random variables.

Definition 3.7 (m-Wise Independent Function). *Let $n, m, p \in \mathbb{N}$, let $\mathcal{K}$ be a key distribution, and let f, $f : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^p$ a function. $(f, \mathcal{K})$ is an m-wise independent function if for every distinct m input values $x_1, \cdots, x_m \in \{0,1\}^n$,*

$$\forall y_1, \cdots, y_m \in \{0,1\}^p : \Pr_{k\leftarrow\mathcal{K}}[f(k, x_1) = y_1 \wedge \cdots \wedge f(k, x_m) = y_m] = 2^{-p\cdot m} \ .$$

Based on m-wise independent functions we define efficiently samplable m-wise independent function families.

Definition 3.8 (Efficient $m(n)$-Wise Independent Function). *Let $m(n), p(n) : \mathbb{N} \to \mathbb{N}$ be functions, let $\mathcal{K} = \{\mathcal{K}_n\}_{n\in\mathbb{N}}$ be an efficiently samplable key distribution, and let $f = \{f_n\}_{n\in\mathbb{N}}$, $f_n : \mathcal{K}_n \times \{0,1\}^n \to \{0,1\}^{p(n)}$ be an efficiently computable function. Then, if for every $n \in \mathbb{N}$, $(f_n, \mathcal{K}_n)$ is an $m(n)$-independent function, then $(f, \mathcal{K})$ is an efficient $m(n)$-wise independent function.*

3.3 Quantum Randomness and Pseudorandomness

The Haar Measure The Haar measure on quantum states is the quantum analogue of the classical uniform distribution over classical bit strings. That is, it is the uniform (continuous) probability distribution on quantum states. Recall that an n-qubit quantum state can be viewed as a unit vector in $\mathbb{C}^{2^n}$, thus the Haar measure on n qubits is the uniform distribution over all unit vectors in $\mathbb{C}^{2^n}$. In this work we denote the n-qubit Haar distribution with μ_n. From this point forward we refer to the uniform distribution over quantum states simply as "random", and don't mention specifically that it is with respect to the Haar distribution.

Scalable Asymptotically Random State Generators We propose a scalable variant to the notion of Asymptotically Random State (ARS) generators which was implicitly defined in [JLS18] and explicitly in [BS19]. Previous works consider an ARS generator to be an efficient quantum algorithm Gen that gets quantum oracle access to $U_f : |x, y\rangle \to |x, y \oplus f(x)\rangle$ for a random classical function f, along with a parameter $n \in \mathbb{N}$ that denotes the number of desired output qubits. The guarantee of the ARS generator is that for any polynomial $t(n)$ in n, $t(n)$ outputs from Gen^{U_f} (executed with the same function f) have negligible trace distance (*in* n) from $t(n)$-copies of a random n-qubit state This means that n plays two roles, it denotes the number of qubits in the output state but also the security parameter that determines the quality of randomness (i.e. how indistinguishable it is from random).

A *Scalable* ARS generator is one that gets two parameters n, λ instead of one. n, as before, denotes the number of wanted output qubits, and λ is a security parameter, thus a scalable ARS generator eliminates the dependence between state size and security.

Definition 3.9 (Asymptotically Random State (ARS) Generator). *A quantum polynomial-time algorithm* Gen *with input* $(1^n, 1^\lambda)$ *for* $n, \lambda \in \mathbb{N}$ *and quantum oracle access to* $U_f : |x, y\rangle \to |x, y \oplus f(x)\rangle$ *for* $f : \{0,1\}^n \to \{0,1\}^{\mathrm{poly}(n,\lambda)}$*, is an ARS generator if there exists a negligible function* $\mathrm{negl}(\cdot)$ *s.t. for every polynomial* $t : \mathbb{N} \to \mathbb{N}$*, for all natural numbers* n, λ*,*

$$\mathrm{TD}\big(D_1, D_2\big) \leq \mathrm{negl}(\lambda) \ ,$$

where the distributions D_1, D_2 *are defined as follows.*

- D_1 : *Sample* $f \leftarrow \big(\{0,1\}^{\mathrm{poly}(n,\lambda)}\big)^{\{0,1\}^n}$*, perform* $t(\lambda)$ *independent executions of* $\mathsf{Gen}^{U_f}(1^n, 1^\lambda)$ *and output the* $t(\lambda)$ *output quantum states.*
- D_2 : *Sample* $|\psi\rangle \leftarrow \mu_n$ *a random* n*-qubit quantum state, and output* $t(\lambda)$ *copies of it:* $|\psi\rangle^{\otimes t(\lambda)}$*. Recall that* μ_n *is the Haar measure on* n *qubits.*

We next define (scalable) quantum state t-design generators and (scalable) pseudorandom quantum state (PRS) generators. After defining these, we briefly describe a general and simple reduction structure that shows how to construct t-designs and PRS generators from any ARS generator.

Approximate Quantum State t-Designs A quantum state t-design [AE07] is a distribution over quantum states that mimics the uniform distribution over quantum states when the number of output copies is restricted to t. A (scalable, approximate) quantum state t-design generator consists of two quantum algorithms K, G. The key sampler algorithm K samples a classical key k given two parameters $1^n, 1^\lambda$ where n denotes the number of qubits and λ denotes the security parameter. The state generation algorithm G gets a key k and outputs an n-qubit state $|\psi\rangle$. Informally, the randomness gaurantee of a t-design generator is that if we sample a key k once from $K(1^n, 1^\lambda)$ and then execute $G(k)$ t times

and output the t outputs, then this output distribution is going to be indistinguishable from t copies of an n-qubit quantum state, for unbounded quantum distinguishers. The formal definition follows.

Definition 3.10 ($\varepsilon(\lambda)$-Approximate State $t(\lambda)$-Design Generator). *Let $\varepsilon(\lambda) : \mathbb{N} \to [0,1]$, $t(\lambda) : \mathbb{N} \to \mathbb{N}$ be functions. We say that a pair of quantum algorithms (K, G) is an $\varepsilon(\lambda)$-approximate state $t(\lambda)$-design generator if the following holds:*

- ***Key Generation.*** *For all $n, \lambda \in \mathbb{N}$, $K(1^n, 1^\lambda)$ always outputs a classical key k.*
- ***State Generation.*** *Given k in the support of $K(1^n, 1^\lambda)$ the algorithm $G(1^n, 1^\lambda, k)$ will always output an n-qubit quantum state.*
- ***Approximate Quantum Randomness.*** *For all $n, \lambda \in \mathbb{N}$,*

$$\mathrm{TD}\big(D_1, D_2\big) \leq \varepsilon(\lambda) \ ,$$

where the distributions D_1, D_2 are defined as follows.

  - *D_1 : Sample $k \leftarrow K(1^n, 1^\lambda)$, perform $t(\lambda)$ independent executions of $G(1^n, 1^\lambda, k)$ and output the $t(\lambda)$ output quantum states.*
  - *D_2 : Sample $|\psi\rangle \leftarrow \mu_n$ a random n-qubit quantum state, and output $t(\lambda)$ copies of it: $|\psi\rangle^{\otimes t(\lambda)}$.*

It is not part of the standard definition, but it is usually the case that the algorithms K, G execute in time $\mathrm{poly}(n, \lambda)$, which is going to be the case in this work as well.

Pseudorandom Quantum States We define scalable Pseudorandom State (PRS) generators. Compared to t-designs, Quantum Pseudorandom State Generators have a slight difference, and formally incomparable randomness guarantee. Mainly, with a PRS we are guaranteed that the output state is going to be indistinguishable for any polynomial number of copies $t(\lambda)$ *without* knowing in advance $t(\lambda)$, however this indistinguishability is only *computational*. That is, it is only guaranteed that computationally bounded distinguishers will be unable to tell the difference between $t(\lambda)$ executions of the generator and $t(\lambda)$ copies of a random quantum state. The scalability property maintains the ability to increase security without increasing the state size n. We remind that the notion of scalability in PRS generators was not considered in previous works [JLS18,BS19] and thus the following definition differs a bit from the previous definition of a PRS, we view this as the more proper definition.

Definition 3.11 (Scalable Pseudorandom Quantum State (PRS) Generator). *We say that a pair of polynomial-time quantum algorithms (K, G) is a Pseudorandom State (PRS) Generator if the following holds:*

- ***Key Generation.*** *For all $n, \lambda \in \mathbb{N}$, $K(1^n, 1^\lambda)$ always outputs a classical key k.*

– ***State Generation.*** *Given k in the support of $K(1^n, 1^\lambda)$ the algorithm $G(1^n, 1^\lambda, k)$ will always output an n-qubit quantum state.*
– ***Quantum Pseudorandomness.*** *For any polynomial $t(\cdot)$ and a non-uniform polynomial-time quantum algorithm $A = \{A_\lambda\}_{\lambda\in\mathbb{N}}$ (with quantum advice) there exists a negligible function $\mathrm{negl}(\cdot)$ such that for all $n, \lambda \in \mathbb{N}$,*

$$\left|\Pr[A_\lambda\big(D_1\big) = 1] - \Pr[A_\lambda\big(D_2\big) = 1]\right| \leq \mathrm{negl}(\lambda) \ ,$$

where the distributions D_1, D_2 are defined as follows.
- *D_1 : Sample $k \leftarrow K(1^n, 1^\lambda)$, perform $t(\lambda)$ independent executions of $G(1^n, 1^\lambda, k)$ and output the $t(\lambda)$ output quantum states.*
- *D_2 : Sample $|\psi\rangle \leftarrow \mu_n$ a random n-qubit quantum state, and output $t(\lambda)$ copies of it: $|\psi\rangle^{\otimes t(\lambda)}$.*

Scalable PRS and Quantum State t-Design Generators from Scalable ARS Generators We recall a generic transformation from previous works that explain how to construct PRS generators and quantum state t-designs from any ARS generator. We start with the paradigm from [JLS18, BS19] that explains a simple way to turn any ARS generator into a PRS generator.

Lemma 3.12. *If there exists a scalable ARS generator and post-quantum one-way functions exist, then there exists a scalable PRS generator.*

Proof (Proof Sketch.). The proof follows the same lines as the proof of [BS19, Claim 4, Section 3.1], with the additional scalability property. The key generator $K(1^n, 1^\lambda)$ of the PRS is the key generator of some quantum-secure pseudorandom function PRF with security parameter $n + \lambda$. For a sampled PRF key k, the state generator algorithm G simply executes the ARS generator with the pseudorandom function instead of the truly random function, $G(1^n, 1^\lambda, k) := \mathsf{Gen}^{U_{\mathsf{PRF}_k}}(1^n, 1^\lambda)$. For a polynomial $t(\cdot)$, $t(\lambda)$ copies of the generated distribution are computationally indistinguishable (by quantum adversaries) from $t(\lambda)$ copies of the standard output distribution of the ARS generator, by the security guarantee of the PRF. Additionally, $t(\lambda)$ copies of the output distribution of the ARS is already known to be indistinguishable (by unbounded distinguishers) from $t(\lambda)$ copies of a random quantum state, and our proof is concluded.

Also, we follow the observation from [BS19] that explains how an ARS generator implies the existence of t-designs (with depth that has logarithmic dependence on t).

Lemma 3.13. *Assume there exists a scalable ARS generator with the following properties:*

– *The generator is implemented by a circuit of depth $T(n, \lambda)$.*
– *For all n, λ, t its output is $\varepsilon(n, \lambda, t)$-indistinguishable from a t-tensor of a random n-qubit state.*

Then there exists an $\varepsilon(n,\lambda,t)$-approximate scalable t-design generator, which is implementable by circuits of depth

$$T(n,\lambda)\cdot\log(n)\cdot\log(2\cdot t\cdot T(n,\lambda))\ .$$

Proof (Proof Sketch.). The proof is similar to the explanation in [BS19, Section 3.2], with slight differences and an additional consideration of the scalability property. The key generator $K(1^n,1^\lambda)$ of the t-design samples an efficient m-wise independent function $\tilde{f}$, where $m := 2t\cdot T(n,\lambda)$. The state generator algorithm G executes the ARS generator with the function $\tilde{f}$ instead with the truly random function, $G(1^n,1^\lambda,\tilde{f}) := \mathsf{Gen}^{U_{\tilde{f}}}(1^n,1^\lambda)$. By [Zha12, Fact 2], The behavior of any quantum algorithm making at most m quantum queries to a $2m$-wise independent function is identical to its behavior when the queries are made to a random function. Therefore if we make t executions of $G(1^n,1^\lambda,\tilde{f})$, each of which makes at most $T(n,\lambda)$ queries to $U_{\tilde{f}}$, then the output distribution of the algorithm $G(1^n,1^\lambda,\tilde{f})$ is the same as that produced by the ARS generator (when it uses a truly random function). Since the classical depth of an m-wise independent function on n bits is $\log(n)\cdot\log(m)$, the proof follows (see elaboration on the classical depth of m-wise independent functions in [BS19, Section 3.2]).

3.4 The Continuous Gaussian and Rounded Gaussian Distributions

In this work we will work with distributions related to the Gaussian distribution over $\mathbb{R}$ denoted $\mathcal{N}(0,1)$, also known as the normal distribution having a mean of 0 and variance of 1. More specifically we will consider the complex Gaussian distribution over $\mathbb{C}$, denoted $\mathcal{N}^{\mathbb{C}}(0,1)$, where both real and imaginary parts of a complex number are sampled independently from $\mathcal{N}(0,1)$.

Rounded Gaussian Distribution. The true Gaussian distribution is continuous and we cannot *exactly* sample from it. Instead, we will use a discrete distribution that we can efficiently sample from. There are quite a few versions of distributions that are discretezations of the Gaussian distribution. In this work we use the rounded Gaussian distribution, which we denote by $\mathcal{N}^{\mathbb{C}}_{\mathrm{R}(\varepsilon,B)}(0,1)$. This distribution is parameterized by $\varepsilon = 2^{-m} > 0$ (for some $m\in\mathbb{N}$) and by $B\in\mathbb{N}$, where B is some integer multiple of ε.

To define the distribution $\mathcal{N}^{\mathbb{C}}_{\mathrm{R}(\varepsilon,B)}(0,1)$ we first define the rounding function $R_{(\varepsilon,B)}(\cdot)$. For a number $x\in\mathbb{R}$, if $|x|>B$ then $R_{(\varepsilon,B)}(x) := 0$, and otherwise $R_{(\varepsilon,B)}(x)$ rounds x up (in absolute value) to the nearest multiple of ε. Formally, if $|x|\le B$ then $R_{(\varepsilon,B)}(x)$ is the number $y\in\mathbb{R}$ that has minimal absolute value and s.t. both $|x|\le|y|$, $\exists k\in\mathbb{Z} : y = k\cdot\varepsilon$. For a complex number $z\in\mathbb{C}$, $R_{(\varepsilon,B)}(z)$ is just applying $R_{(\varepsilon,B)}(\cdot)$ to both real and imaginary parts of z.

We define $\mathcal{N}^{\mathbb{C}}_{\mathrm{R}(\varepsilon,B)}(0,1)$ to be the output distribution of the following process: Sample $z\leftarrow\mathcal{N}^{\mathbb{C}}(0,1)$ and output $R_{(\varepsilon,B)}(z)$. The output of $\mathcal{N}^{\mathbb{C}}_{\mathrm{R}(\varepsilon,B)}(0,1)$ is specified by a number between 0 and B with precision $\varepsilon = 2^{-m}$, thus the output length in bits is bounded by $m+\lceil\log_2(B)\rceil$.

We use the following standard fact about (classical) Gaussian sampling.

Fact 3.14 (Efficient Rounded Gaussian Sampling) *There is a sampling algorithm $G^{\mathbb{C}}_{R()}$ that takes $1^m, B$ (and random tape) as input, runs in polynomial time, i.e.* $\mathrm{poly}(m, \log B)$*, and samples from a distribution that has statistical distance at most 2^{-m} from the rounded Gaussian distribution $\mathcal{N}^{\mathbb{C}}_{R(2^{-m},B)}(0,1)$.*

4 General Tools for Quantum Information

4.1 State Generation of Balanced Vectors

In this subsection we describe a simple procedure that given quantum oracle access to the entries of some general, not necessarily normalized vector $v \in \mathbb{C}^{2^n}$, generates the n-qubit quantum state $|v\rangle$ that corresponds to (the normalization of) v. More formally, the procedure gets two pieces of information about v:

- Quantum oracle access to U_v, the unitary of the classical function $v : \{0,1\}^n \to \{0,1\}^k$ (where k is the description size in bits of each entry of v) that describes the vector v and maps $v(x) := v_x$.
- An upper bound $M \in \mathbb{N}$ on any entry of v, that is, $\max_{x\in[N]} |v_x| \leq M$.

The procedure runs in time $\mathrm{poly}(n, k, \log M)$ and outputs the quantum state $|v\rangle$ as follows.

$\mathsf{BVS}^{U_v}(M)$:

1. Define the quantum unitary $U_{\tilde{v}}$ which is the unitary of the classical function $\tilde{v} : \{0,1\}^n \to \{0,1\}^{k+\log(M)}$ that maps $\tilde{v}(x) := v_x \cdot \frac{1}{M}$.
   It's trivial to simulate $U_{\tilde{v}}$ given U_v: Given a query $|x, y\rangle$, we concatenate an ancilla of zeros and apply U_v to get $|x, y, v_x\rangle$, then apply a simple unitary that multiplies by $\frac{1}{M}$ (on the last register as input and on the second register as output) to obtain $|x, y \oplus v_x \cdot \frac{1}{M}, v_x\rangle$, and then use U_v again to uncompute the last register.
2. Execute quantum rejection sampling, $(b\,, |\hat{v}\rangle) \leftarrow \mathsf{QRS}^{U_{\tilde{v}}}(|+\rangle^{\otimes n})$ (see specification of QRS in Theorem 3.5). If $b =$ `fail` then output `fail`, otherwise output $|\hat{v}\rangle$.

Claim 4.1 (BVS Success Probability) *If $\max_{i\in[N]} |v_i| \leq M$ then the execution $\mathsf{BVS}^{U_v}(M)$ always outputs either* `fail` *or the quantum state $v \cdot \frac{1}{\|v\|} = |\hat{v}\rangle$, furthermore the execution succeeds and outputs the quantum state $|\hat{v}\rangle$ with probability at least $\frac{\|v\|^2}{M^2 \cdot N}$.*

Proof. We need to make sure that we execute the quantum rejection sampling algorithm QRS with correct parameters (specified in Theorem 3.5), and also understand what exactly are the parameters for QRS. As the starting state $|\alpha\rangle$ we input $|+\rangle^{\otimes n}$, our target state $|\beta\rangle$ is $|\hat{v}\rangle = \frac{|v\rangle}{\|v\|}$. As the state transformation unitary U we use $U_{\tilde{v}}$, that is, the unitary of the classical function $f(x) := v_x \cdot \frac{1}{M}$.

It follows that there exists an upper bound $d \geq \max_{x\in\{0,1\}^n} \left|\frac{\beta_x}{\alpha_x}\right|$ s.t. $\forall x \in \{0,1\}^n : f(x) := f(x) := v_x \cdot \frac{1}{M} = \frac{\beta_x/\alpha_x}{d}$, by taking $d := \frac{M\cdot\sqrt{N}}{\|v\|}$.

– d is indeed an upper bound:

$$\forall x \in \{0,1\}^n : \left|\frac{\beta_x}{\alpha_x}\right| = \left|\frac{v_x/\left\|v\right\|}{1/\sqrt{N}}\right| = \left|v_x \cdot \frac{\sqrt{N}}{\|v\|}\right| \leq \frac{M \cdot \sqrt{N}}{\|v\|} \ .$$

– $f(\cdot)$ indeed computes $\frac{\beta_x}{\alpha_x}/d$:

$$\begin{aligned}
\forall x \in \{0,1\}^n : f(x) &:= v_x \cdot \frac{1}{M} \\
&= v_x \cdot \frac{1}{M} \cdot \frac{\sqrt{N}}{\sqrt{N}} \cdot \frac{\|v\|}{\|v\|} \\
&= \left(\frac{v_x}{\|v\|} \cdot \sqrt{N}\right) \cdot \left(\frac{1}{M} \cdot \frac{\|v\|}{\sqrt{N}}\right) \\
&= \frac{\left(\frac{v_x/\|v\|}{1/\sqrt{N}}\right)}{\left(\frac{M \cdot \sqrt{N}}{\|v\|}\right)} \\
&= \frac{\beta_x/\alpha_x}{d} \ .
\end{aligned}$$

The conditions for QRS hold, and thus from the correctness guarantee of quantum rejection sampling we can be sure that the algorithm BVS will always output either `fail` or $|\beta\rangle := |\hat{v}\rangle$. As for the probability of success in outputting $|\hat{v}\rangle$, again from the success guarantees of QRS this probability is at least $\frac{1}{d^2} = \frac{\|v\|^2}{M^2 \cdot N}$.

The above procedure tries to generate $|\hat{v}\rangle$ once and it will be convenient to have an amplified version of this algorithm as a black box, this is an option because we can always re-generate the state $|+\rangle^{\otimes n}$ efficiently and retry. The amplified version of the algorithm is with the same name and have one more parameter $k \in \mathbb{N}$ (amplification parameter), that is, $\mathsf{BVS}^{U_v}(M,k)$.

The amplified version of BVS executes k (parallel) repetitions of $\mathsf{BVS}^{U_v}(M)$, if all fail it outputs `fail`, and if either succeeds it outputs the generated state $|\hat{v}\rangle$. The probability of $\mathsf{BVS}^{U_v}(M,k)$ to succeed in generating the state $|\hat{v}\rangle$ follows.

Claim 4.2 (Amplified BVS Success Probability) *If $\max_{i\in[N]} |v_i| \leq M$ then the algorithm $\mathsf{BVS}^{U_v}(M,k)$ always outputs either `fail` or the quantum state $v \cdot \frac{1}{\|v\|} = |\hat{v}\rangle$, furthermore the algorithm succeeds and outputs the quantum state $|\hat{v}\rangle$ with probability at least $1 - e^{-\frac{k \cdot \|v\|^2}{M^2 \cdot N}}$.*

Proof.

$$\begin{aligned}\Pr\left[\mathsf{BVS}^{U_v}(M,k)=\mathtt{fail}\right] &= \Pr\left[\mathsf{BVS}^{U_v}(M) \text{ failed } k \text{ times in a row}\right]\\ &= \left(\Pr\left[\mathsf{BVS}^{U_v}(M)=\mathtt{fail}\right]\right)^k\\ &\leq \left(1-\frac{\|v\|^2}{M^2\cdot N}\right)^k\\ &\leq e^{-\frac{k\cdot\|v\|^2}{M^2\cdot N}} \ .\end{aligned}$$

4.2 Analytic Tools for Distributions

In this subsection we describe some analytic tools for bounding trace norm between two distributions, for multiple output copies. We start with an elementary property of trace distance and classical statistical distance that we will use in our construction.

Lemma 4.3 (Classical Statistical Distance Implies Trace Distance). *Let $n\in\mathbb{N}$ and let D_1, D_2 be two distributions over unit vectors in $\mathbb{C}^{2^n}$. Let $\tilde{D}_1$, $\tilde{D}_2$ be the quantum-state distributions of D_1, D_2, that is, for $b\in\{0,1\}$, a sample from $\tilde{D}_b$ is generated by sampling a vector v from D_b, and outputting an n-qubit register in the state described by v.*

Then, if $\mathrm{SD}(D_1,D_2)\leq\varepsilon$, then for every number of copies $t\in\mathbb{N}$,

$$\mathrm{TD}\left(\mathbb{E}_{|v\rangle\leftarrow\tilde{D}_1}\left[(|v\rangle\langle v|)^{\otimes t}\right],\mathbb{E}_{|v\rangle\leftarrow\tilde{D}_2}\left[(|v\rangle\langle v|)^{\otimes t}\right]\right)\leq\varepsilon \ .$$

Proof. Intuitively, the proof follows from the fact that a computationally unbounded mapping can always capture the computation of an (even unbounded) quantum process, along with the fact that when the classical description of a state is available then there is no advantage in having more than a single copy. Formally, we assume towards contradiction there is a projective measurement A (with output in $\{0,1\}$) that distinguishes between a t-tensor of $\tilde{D}_1$ and a t-tensor of $\tilde{D}_2$ with advantage bigger than ε, and describe a (randomized) distinguisher $\mathsf{A}':\mathbb{C}^{2^n}\to\{0,1\}$ that distinguishes between D_1,D_2 with advantage bigger than ε. Let A denote the Hermitian matrix that corresponds to the projective measurement A.

The distinguisher A' is defined as follows. Given an input $v\in\mathbb{C}^{2^n}$, consider the vector $|v'\rangle=|v\rangle^{\otimes t}$, and compute the value $p=\langle v'|A|v'\rangle$. We note that this value is exactly the probability that A outputs 1 when input the quantum state $|v\rangle^{\otimes t}$. The distinguisher A' then outputs 1 with probability p and 0 with probability $1-p$. By definition, the advantage of A' in distinguishing D_1 and D_2 is identical to the advantage of A in distinguishing the t-tensored $\tilde{D}_1$ and $\tilde{D}_2$.

Robustness to Small Shifts. Lemma 4.3 asserts that distributions on quantum states are indistinguishable if they are induced by indistinguishable distributions

over vectors in the respective Hilbert space. This is a very strict condition and in fact in many cases distributions on quantum states can be indistinguishable even if the the respective distributions over vectors are highly distinguishable.

This will be useful in the context of this work since we wish to show indistinguishability between the Haar random distribution, which corresponds to a continuous distribution over the sphere, and an efficiently samplable distribution (with oracle access to a random function), which necessarily produces a discrete distribution over vectors. Hence, the two distributions over vectors are necessarily distinguishable (with advantage 1), and yet we will be able to bound the distinguishing gap between the quantum states.

Technically, we rely on the well known property that quantum states that correspond to vectors with inner product close to 1 are indistinguishable. This is formalized in Lemma 4.4 below, which considers a distribution over vectors, and a small perturbation of this distribution, that does not shift the vector by too much. We show that such perturbation, which in particular captures the case of rounding a continuous distribution into some discrete domain, would be indistinguishable in terms of the resulting quantum state.

Lemma 4.4 (Angular Indistinguishability). *Let $n \in \mathbb{N}$, $\varepsilon \in [0,1]$, let D be a distribution over (not necessarily normalized) vectors in $V \subseteq \mathbb{C}^{2^n}$, let $\varphi : V \to \mathbb{C}^{2^n}$ be a function and let $\hat{\varphi} : V \to \mathbb{C}^{2^n}$ be the normalized version of φ, $\hat{\varphi}(v) := \frac{\varphi(v)}{\|\varphi(v)\|}$. Assume that for every $v \in V$, the normalization of v and its $\hat{\varphi}$-image are close on the unit sphere, that is,*

$$|\langle \hat{v} | \hat{\varphi}(v) \rangle| \geq 1 - \varepsilon,$$

then for all $t \in \mathbb{N}$,

$$\mathrm{TD}\Big(\mathbb{E}_{v \leftarrow D} \big[\left(|\hat{v}\rangle\langle\hat{v}|\right)^{\otimes t} \big] , \mathbb{E}_{v \leftarrow D} \big[\left(|\hat{\varphi}(v)\rangle\langle\hat{\varphi}(v)|\right)^{\otimes t} \big] \Big) \leq \sqrt{2t\varepsilon} \ . \tag{2}$$

In the original version of this work, the above lemma was proven using a proof different from the one below. We thank the CRYPTO reviewer for suggesting the simplified proof presented below.

Proof. The lemma follows since

$$\begin{aligned}
&\mathrm{TD}\Big(\mathbb{E}_{v \leftarrow D} \big[\left(|\hat{v}\rangle\langle\hat{v}|\right)^{\otimes t} \big] , \mathbb{E}_{v \leftarrow D} \big[\left(|\hat{\varphi}(v)\rangle\langle\hat{\varphi}(v)|\right)^{\otimes t} \big] \Big) \\
&\qquad \leq \mathbb{E}_{v \leftarrow D} \Big[\mathrm{TD}\big(\left(|\hat{v}\rangle\langle\hat{v}|\right)^{\otimes t} , \left(|\hat{\varphi}(v)\rangle\langle\hat{\varphi}(v)|\right)^{\otimes t} \big) \Big] \\
&\qquad \leq \mathbb{E}_{v \leftarrow D} \Big[\sqrt{1 - |\langle \hat{v} | \hat{\varphi}(v) \rangle|^{2t}} \Big] \\
&\qquad \leq \mathbb{E}_{v \leftarrow D} \Big[\sqrt{1 - (1-\varepsilon)^{2t}} \Big] \\
&\qquad \leq \sqrt{2t\varepsilon} \ ,
\end{aligned}$$

where the first inequality follows from the convexity of trace distance, the second follows from the relation between trace distance and fidelity $\mathrm{TD}(\rho, \sigma) \leq$

$\sqrt{1-F(\rho,\sigma)}$ and the fact that for pure states $|u\rangle, |v\rangle$ the Fidelity is $|\langle u|v\rangle|^2$, and the last inequality follows from Bernoulli's inequality which implies $(1-\varepsilon)^{2\cdot t} \geq 1-2t\varepsilon$.

5 Scalable Asymptotically Random State (ARS) Generator

In this section we describe a procedure that given quantum oracle access to a random classical function, efficiently samples random quantum states that are arbitrarily random (i.e. we can scale up the randomness of our sampled state and make it increasingly harder to distinguish from a random quantum state, for an increasing number of output copies, without increasing the number of qubits in the state) and can generate multiple copies of a state when executed multiple times with oracle access to the same function. More formally, we describe a sampling procedure with the following inputs:

- 1^n: Number of wanted qubits in the output state.
- 1^λ: Security parameter that measures "how random" the output state is going to be (i.e. how hard will it be to distinguish t copies of the sampled state from t copies of a random quantum state, as a function of λ, t).
- Quantum oracle access to U_f: For a function $f : \{0,1\}^n \to \{0,1\}^{\mathrm{poly}(n,\lambda)}$ (for some polynomial $\mathrm{poly}(\cdot)$, specified later), the sampling procedure gets oracle access to the unitary mapping U_f of f.

The formal statement that explains how to construct a scalable ARS generator follows.

Theorem 5.1 (Scalable ARS Generator Construction). *There exists a scalable ARS generator* Gen *that for every* $n \in \mathbb{N}$ *number of qubits,* $5 \leq \lambda \in \mathbb{N}$ *security parameter and* $t \in \mathbb{N}$ *number of copies, satisfies the following trace distance bound,*

$$\mathrm{TD}\Big(D_1, D_2\Big) \leq (t+8)\cdot e^{-\lambda} + \Big(5\sqrt{t}+\lambda+1\Big)\cdot 2^{-\lambda} + 2\cdot\left(\frac{8}{10}\right)^{\lambda} \quad ,$$

where the distributions D_1, D_2 *are defined as follows:*

- D_1 : *Sample* $\tilde{f} \leftarrow \left(\{0,1\}^{\mathrm{poly}(n,\lambda)}\right)^{\{0,1\}^n}$*, execute* t *times the generation algorithm* $\mathsf{Gen}^{U_{\tilde{f}}}(1^n, 1^\lambda)$ *and output the* t *output states.*
- D_2 : *Sample* $|\psi\rangle$ *a random* n*-qubit state and output* $|\psi\rangle^{\otimes t}$*.*

Proof. We start with describing the procedure of $\mathsf{Gen}^{U_{\tilde{f}}}(1^n, 1^\lambda)$. First, we denote $\varepsilon := 2^{-n-\lambda}$, $B := \lceil 2\sqrt{n+\lambda} \rceil$ and set the polynomial $\mathrm{poly}(n,\lambda)$ that denotes the output size of $\tilde{f}$ to be $\lambda \cdot r(\varepsilon, B)$, where $r(\varepsilon, B)$ is the randomness complexity of the rounded Gaussian sampler $G^{\mathbb{C}}_{\mathrm{R}(\varepsilon,B)}$. Given the oracle access

to $\tilde{f} \in \left(\{0,1\}^{\lambda \cdot r(\varepsilon,B)}\right)^{\{0,1\}^n}$, the algorithm starts with deciding on a different function $f \in \left(\{0,1\}^{r(\varepsilon,B)}\right)^{\{0,1\}^n}$ that it is going to use.

In what follows, denote $N := 2^n$, for a function $h \in \left(\{0,1\}^{r(\varepsilon,B)}\right)^{\{0,1\}^n}$ denote by v^h the vector that is created by rounded Gaussian sampling with h, that is, $\forall x \in \{0,1\}^n, v^h_x := G^{\mathbb{C}}_{\mathrm{R}(\varepsilon,B)}(h(x))$. We think of $\tilde{f}$, that has an output length of $\lambda \cdot r(\varepsilon, B)$, as λ different functions, each having an output length of $r(\varepsilon, B)$. Specifically, for $i \in [\lambda]$ define the function $f_i \in \left(\{0,1\}^{r(\varepsilon,B)}\right)^{\{0,1\}^n}$ as the function that for input $x \in \{0,1\}^n$ outputs the i-th packet of $r(\varepsilon, B)$ bits from $\tilde{f}(x)$.
The procedure of Gen follows.

1. Decide on a function $f \in \left(\{0,1\}^{r(\varepsilon,B)}\right)^{\{0,1\}^n}$:
   - If $N > \lambda$, we actually use only the first $r(\varepsilon, B)$ bits of the output of $\tilde{f}$. That is, f is simply f_1.
   - If $N \leq \lambda$, iterate for $i \in [\lambda]$:
     - Compute the vector v^{f_i} by applying $G^{\mathbb{C}}_{\mathrm{R}(\varepsilon,B)}$ to each of the N outputs of f_i. If $\left\|v^{f_i}\right\| \geq \frac{\sqrt{N}}{2}$, denote $f := f_i$ and halt the loop.[15]

     If you executed all iterations and did not get a function f_i s.t. $\left\|v^{f_i}\right\| \geq \frac{\sqrt{N}}{2}$, halt and output $|0^n\rangle$ (as a sign of failure).
2. Given f execute $\mathsf{BVS}^{U_{vf}}(\sqrt{2} \cdot B, 8 \cdot \lambda \cdot B^2)$ and output the n-qubit quantum state generated by BVS.

The full analysis and the rest of the proof, showing why a t-tensor of the output of the generator (i.e. the distribution D_1) and a t-tensor of a random quantum state (i.e. the distribution D_2) are indistinguishable, is in the full version of this work[16].

References

AE07. Andris Ambainis and Joseph Emerson. Quantum t-designs: t-wise independence in the quantum world. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 129–140. IEEE, 2007.

AMR19. Gorjan Alagic, Christian Majenz, and Alexander Russell. Efficient simulation of random states and random unitaries. *CoRR*, abs/1910.05729, 2019.

BS19. Zvika Brakerski and Omri Shmueli. (pseudo) random quantum states with binary phase. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I*, volume 11891 of *Lecture Notes in Computer Science*, pages 229–250. Springer, 2019.

JLS18. Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 126–152. Springer, 2018.

[15] Note that all steps here can be done efficiently in λ, because $\lambda \geq N = 2^n$.
[16] The full version of this paper is available at https://arxiv.org/abs/2004.01976.

ORR13. Maris Ozols, Martin Roetteler, and Jérémie Roland. Quantum rejection sampling. *ACM Transactions on Computation Theory (TOCT)*, 5(3):1–33, 2013.

PSW06. Sandu Popescu, Anthony J Short, and Andreas Winter. Entanglement and the foundations of statistical mechanics. *Nature Physics*, 2(11):754–758, 2006.

Zha12. Mark Zhandry. How to construct quantum random functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687. IEEE, 2012.

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268. Springer, 2019.