Pseudorandom (Function-Like) Quantum State Generators: New Definitions and Applications

Prabhanjan Ananth[1], Aditya Gulati[2], Luowen Qian[3⋆], and Henry Yuen[4⋆⋆]

[1] University of California, Santa Barbara, USA, prabhanjan@cs.ucsb.edu
[2] University of California Santa Barbara, USA, adityagulati@ucsb.edu
[3] Boston University, USA, luowenq@bu.edu
[4] Columbia University, USA, hyuen@cs.columbia.edu

Abstract. Pseudorandom quantum states (PRS) are efficiently constructible states that are computationally indistinguishable from being Haar-random, and have recently found cryptographic applications. We explore new definitions, new properties and applications of pseudorandom states, and present the following contributions:

1. **New Definitions**: We study variants of pseudorandom *function-like* state (PRFS) generators, introduced by Ananth, Qian, and Yuen (CRYPTO'22), where the pseudorandomness property holds even when the generator can be queried adaptively or in superposition. We show feasibility of these variants assuming the existence of post-quantum one-way functions.
2. **Classical Communication**: We show that PRS generators with logarithmic output length imply commitment and encryption schemes with *classical communication*. Previous constructions of such schemes from PRS generators required quantum communication.
3. **Simplified Proof**: We give a simpler proof of the Brakerski–Shmueli (TCC'19) result that polynomially-many copies of uniform superposition states with random binary phases are indistinguishable from Haar-random states.
4. **Necessity of Computational Assumptions**: We also show that a secure PRS with output length logarithmic, or larger, in the key length necessarily requires computational assumptions.

⋆ Supported by DARPA under Agreement No. HR00112020023.

⋆⋆ Supported by AFOSR award FA9550-21-1-0040 and NSF CAREER award CCF-2144219.

1 Introduction

The study of pseudorandom objects is central to the foundations of cryptography. After many decades, cryptographers have developed a deep understanding of the zoo of pseudorandom primitives such as one-way functions (OWF), pseudorandom generators (PRG), and pseudorandom functions (PRF) [8,9].

The study of pseudorandomness in the quantum setting, on the other hand, is just getting started. Objects such as state and unitary k-designs have been studied extensively, but these are best thought of as quantum analogues of k-wise independent hash functions [1,6]. There are unconditional constructions of state and unitary designs and they do not imply any computational assumptions [1,18].

Quantum pseudorandomness requiring computational assumptions, in contrast, has been studied much less. Ji, Liu, and Song introduced the notion of *pseudorandom quantum states (PRS)* and *pseudorandom quantum unitaries (PRU)* [11]. At a high level, these are efficiently sampleable distributions over states/unitaries that are computationally indistinguishable from being sampled from the Haar distribution (i.e., the uniform measure over the space of states/ unitaries). Ji, Liu, and Song as well as Brakerski and Shmueli have presented constructions of PRS that are based on quantum-secure OWFs [11,3,4]. Kretschmer showed, however, that PRS do not necessarily imply OWFs; there are oracles relative to which PRS exist but OWFs don't [12]. This was followed by recent works that demonstrated the cryptographic utility of PRS: basic cryptographic tasks such as bit commitment, symmetric-key encryption, and secure multiparty computation can be accomplished using only PRS as a primitive [2,16]. It is an intriguing research direction to find more cryptographic applications of PRS and PRU.

The key idea in [2] that unlocked the aforementioned applications was the notion of a *pseudorandom function-like state (PRFS) generator*. To explain this we first review the definition of PRS generators. A quantum polynomial-time (QPT) algorithm G is a PRS generator if for a uniformly random key $k \in \{0,1\}^\lambda$ (with λ being the security parameter), polynomially-many copies of the state $|\psi_k\rangle = G(k)$ is indistinguishable from polynomially-many copies of a state $|\vartheta\rangle$ sampled from the Haar measure by all QPT algorithms. One can view this as a quantum analogue of classical PRGs. Alternately, one could consider a version of PRS where the adversary only gets one copy of the state. However, as we will see later, the multi-copy security of PRS will play a crucial role in our applications.

The notion of PRFS generator introduced by [2] is a quantum analogue of PRF (hence the name *function-like*): in addition to taking in a key k, the generator G also takes an *input* x (just like a PRF takes a key k and an input x). Let $|\psi_{k,x}\rangle = G(k,x)$. The pseudorandomness property of G is that for all sequences of inputs $(x_1,\ldots,x_s)$ for polynomially large s, averaged over the key k, the collection of states $|\psi_{k,x_1}\rangle^{\otimes t},\ldots,|\psi_{k,x_s}\rangle^{\otimes t}$ for polynomially large t is computationally indistinguishable from $|\vartheta_1\rangle^{\otimes t},\ldots,|\vartheta_s\rangle^{\otimes t}$ where the $|\vartheta_i\rangle$'s are sampled independently from the Haar measure. In other words, while PRS generators look like (to a computationally bounded distinguisher) they are sampling a *single* state from the Haar measure, PRFS generators look like they are sampling *many* (as compared to the key length) states from the Haar measure. Importantly, this still holds true even when the distinguisher is given the inputs $x_1, \ldots, x_s$.

As mentioned, this (seemingly) stronger notion of quantum pseudorandomness provided a useful conceptual tool to perform cryptographic tasks (encryption, commitments, secure computation, etc) using pseudorandom states alone. Furthermore, [2] showed that for a number of applications, PRFS generators with logarithmic input length suffices and furthermore such objects can be constructed in a black-box way from PRS generators.[5]

Despite exciting progress in this area in the last few years, there is still much to understand about the properties, relationships, and applications of pseudorandom states. In this paper we explore a number of natural questions about pseudorandom states:

- *Feasibility of Stronger Definitions of PRFS*: In the PRFS definition of [2], it was assumed that the set of inputs on which the adversary obtains the outputs are determined ahead of time. Moreover, the adversary could obtain the output of PRFS on only classical inputs. This is often referred to as *selective security* in the cryptography literature. For many interesting applications, this definition is insufficient[6]. This leads us to ask: *is it feasible to obtain strengthened versions of PRFS that maintain security in the presence of adaptive and superposition queries?*
- *Necessity of Assumptions*: In the classical setting, essentially all cryptographic primitives require computational assumptions, at the very least $\mathsf{P} \neq \mathsf{NP}$. What computational assumptions are required by pseudorandom quantum states? The answer appears to depend on the output length of the PRS generator. Brakerski and Shmueli [4] constructed PRS generators with output length $c \log \lambda$ for some $c > 0$ satisfying statistical security (in other words, the outputs are statistically close to being Haar-random). On the other hand, Kretschmer showed that the existence of PRS generators with output length λ implies that $\mathsf{BQP} \neq \mathsf{PP}$ [12]. This leads to an intriguing question: *is it possible to unconditionally show the existence of $n(\lambda)$-length output PRS, for some $n(\lambda) \geq \log(\lambda)$?*
- *Necessity of Quantum Communication*: A common theme in all the different PRS-based cryptographic constructions of [2,16] is that the parties involved in the system perform quantum communication. Looking forward, it is conceivable that quantum communication will be a much more expensive resource than having access to a quantum computer. Achieving quantum cryptography with classical communication has been an important direction, dating back to Gavinsky [7]. We ask the following question: *is quantum communication inherent in the cryptographic constructions based on PRS*?

[5] However, unlike the equivalence between PRG and PRF in the classical setting [8], it is not known whether *every* PRFS generator can be constructed from PRS generators in a black-box way.

[6] For example, the application of private-key encryption from PRFS as described in [2] is only selectively secure. This is due to the fact that the underlying PRFS is selectively secure.

1.1 Our Results

We explore the aforementioned questions. Our results include the following.

Adaptive-Secure and Quantum-Accessible PRFS. As mentioned earlier, the notion of PRFS given by [2] has *selective security*, meaning that the inputs $x_1, \ldots, x_s$ are fixed ahead of time. Another way of putting it is, the adversary can only make non-adaptive, classical queries to the PRFS generator (where by query we mean, submit an input x to the generator and receive $|\psi_{k,x}\rangle = G(k, x)$ where k is the hidden, secret key).

We study the notion of *adaptively secure PRFS*, in which the security holds with respect to adversaries that can make queries to the generator adaptively. We consider two variants of this: one where the adversary is restricted to making classical queries to the generator (we call this a *classically-accessible adaptively secure PRFS*), and one where there are no restrictions at all; the adversary can even query the generator on a *quantum superposition of inputs* (we call this a *quantum-accessible adaptively secure PRFS*). These definitions can be found in Section 3.

We then show feasibility of these definitions by constructing classically- and quantum-accessible adaptively secure PRFS generators from the existence of post-quantum one-way functions. These constructions are given in the full version of the paper.

A Sharp Threshold For Computational Assumptions. In Section 4 we show that there is a sharp threshold between when computational assumptions are required for the existence of PRS generators: we give a simple argument that demonstrates that PRS generators with $\log \lambda$-length outputs require computational assumptions on the adversary[7]. This complements the aforementioned result of Brakerski and Shmueli [4] that shows $c \log \lambda$-length PRS for some $c > 0$ do not require computational assumptions. We also note that the calculations of [12] can be refined to show that the existence of $(1+\epsilon)\log \lambda$-length PRS for all $\epsilon > 0$ implies that $\mathsf{BQP} \neq \mathsf{PP}$.

PRS-Based Constructions With Classical Communication. We show that bit commitments and pseudo one-time pad schemes can be achieved using only classical communication based on the existence of PRS with λ-bit keys and $O(\log(\lambda))$-output length. This improves upon the previous result of [2] who achieved bit commitments and pseudo one-time pad schemes from PRS using quantum communication. However, we note that [2] worked with a wider range

[7] We also note that there is a much more roundabout argument for a quantitatively weaker result: [2] constructed bit commitment schemes from $O(\log \lambda)$-length PRS. If such PRS were possible to construct unconditionally, this would imply information-theoretically secure bit commitment schemes in the quantum setting. However, this contradicts the famous results of [13,15], which rules out this possibility. Our calculation, on the other hand, directly shows that $\log \lambda$ (without any constants in front) is a sharp threshold.

of parameters while our constructions are based on PRS with $O(\log(\lambda))$-output length.

En route, we use quantum state tomography (or tomography for short), a well studied concept in quantum information. Roughly speaking, tomography, allows for obtaining a classical string u that captures some properties of an unknown quantum state ρ, given many copies of this state.

We develop a new notion called *verifiable tomography* that might particularly be useful in cryptographic settings. Verifiable tomography allows for verifying whether a given string u is consistent (according to some prescribed verification procedure) with a quantum state ρ. We present the definition and instantiations of verifiable tomography in Section 5. In Section 6, we use verifiable tomography to achieve the aforementioned applications. At a high level, our constructions are similar to the ones in [2], except that verifiable tomography is additionally used to make the communication classical.

A Simpler Analysis of Binary-Phase PRS. Consider the following construction of PRS. Let $\{F_k : \{0,1\}^n \to \{0,1\}\}_{k\in\{0,1\}^\lambda}$ denote a (quantum-secure) pseudorandom function family. Then $\{|\psi_k\rangle\}_k$ forms a PRS, where $|\psi_k\rangle$ is defined as

$$|\psi_k\rangle = 2^{-n/2} \sum_{x\in\{0,1\}^n} (-1)^{F_k(x)} |x\rangle \ . \tag{1}$$

In other words, the pseudorandom states are *binary phase states* where the phases are given by a pseudorandom function. This is a simpler construction of PRS than the one originally given by [11], where the phases are pseudorandomly chosen N-th roots of unity with $N = 2^n$. Ji, Liu, and Song conjectured that the binary phase construction should also be pseudorandom, and this was confirmed by Brakerski and Shmueli [3].

We give a simpler proof of this in the full version, which may be of independent interest.

1.2 Threshold For Computational Assumptions

We show that PRS generators with λ-bit keys and $\log\lambda$-length outputs cannot be statistically secure. To show this we construct an inefficient adversary, given polynomially many copies of a state, can distinguish whether the state was sampled from the output distribution of a $\log\lambda$-length PRS generator or sampled from the Haar distribution on $\log\lambda$-qubit states with constant probability.

Simple Case: PRS output is always pure. Let us start with a simple case when the PRS generator is such that each possible PRS state is pure. Consider the subspace spanned by all possible PRS outputs. The dimension of the subspace spanned by these states is atmost 2^λ: the reason being that there are at most 2^λ keys. Now, consider the subspace spanned by t-copies of PRS states. The dimension of this subspace is still at most 2^λ and in particular, independent of t. Define $P^{(t)}$ to be a projector (which could have an inefficient implementation) onto this subspace. By definition, the measurement of t copies of the output of a PRS generator with respect to $P^{(t)}$ always succeeds.

Recall that the subspace spanned by t-copies of states sampled from the Haar distribution (of length $\log \lambda$) is a symmetric subspace of dimension $\binom{2^\lambda+t-1}{t}$. By choosing t as an appropriate polynomial (in particular, set $t \gg \lambda$), we can make $\binom{2^\lambda+t-1}{t} \gg 2^\lambda$, such that a measurement with $P^{(t)}$ on t copies of states sampled from the Haar distribution fails with constant probability. Hence, an adversary, who just runs P, can successfully distinguish between t copies of the output of a $\log \lambda$-length PRS generator and t copies of a sample from a Haar distribution with constant probability.

General Case. Now let us focus on the case when the PRS generator can also output mixed states. Then we have 2 cases:

- *The majority of outputs of the PRS generator are negligibly close to a pure state:* In this case, we show that the previous approach still works. We replace the projector $P^{(t)}$ with a projection onto the space spanned by states closest to the output states of the PRS generator and we can show that modified projector still succeeds with constant probability.
- *The majority of outputs of the PRS generator are not negligibly close to a pure state:* In this case, most PRS outputs have purity[8] non-negligibly away from 1. Thus, we can violate the security of PRS as follows: run polynomially (in λ) many SWAP tests to check if the state is mixed or not. When the input state is from a Haar distribution, the test will always determine the input state to be pure. On the other hand, if the input state is the output of a PRS generator, the test will determine the input to be pure with probability that is non-negligibly bounded away from 1. Thus, this case cannot happen if the PRS generator is secure.

Details can be found in Section 4.

1.3 Cryptographic Applications With Classical Communication

We show how to construct bit commitments and pseudo one-time encryption schemes from $O(\log(\lambda))$-output PRS with classical communication. Previously, [2] achieved the same result for a wider range of parameters. In this overview, we mainly focus on bit commitments since the main techniques used in constructing commitments will be re-purposed for designing pseudo one-time encryption schemes.

We use the construction of bit commitments from [2] as a starting point. Let $d = O(\log \lambda)$, $n = O(\log \lambda)$ and G is a (d, n)-PRFS generator[9]. The commitment scheme from [2] is as follows:

[8] A density matrix ρ has purity p if $\mathrm{Tr}(\rho^2) = p$.

[9] This in turn can be built from $O(\log(\lambda))$-output PRS as shown in [2].

– In the commit phase, the receiver sends a random $2^d n$-qubit Pauli $P = P_1 \otimes P_2 \otimes \cdots \otimes P_{2^d-1}$ to the sender, where each P_i is an n-qubit Pauli. The sender on input bit b, samples a key k uniformly at random from $\{0,1\}^\lambda$. The sender then sends the state $\rho = \bigotimes_{x\in[2^d]} P_x^b \sigma_{k,x} P_x^b$, where $\sigma_{k,x} = G(k,x)$ to the receiver.
– In the reveal phase, the sender sends (k, b) to the receiver. The receiver accepts if $P^b \rho P^b$ is a tensor product of the PRFS evaluations of (k, x), for all $x = 0, \ldots, 2^d - 1$.

To convert this scheme into one that only has classical comunication, we need a mechanism to generate classical information c from ρ, where ρ is generated from (k, b) as above, that have the following properties:

1. *Classical Description*: c can be computed efficiently and does not leak any information about b.
2. *Correctness*: (k, b) is accepted as a valid opening for c,
3. *Binding*: (k', b'), for $b \neq b'$, is rejected as an opening for c

State Tomography. To design such a mechanism, we turn to quantum state tomography. Quantum state tomography is a process that takes as input multiple copies of a quantum state σ and outputs a string u that is close (according to some distance metric) to a classical description of the state σ. In general, tomography procedures require exponential in d number of copies of a state and also run in time exponential in d, where d is the dimension of the state. Since the states in question are $O(\log(\lambda))$-output length PRFS states, all the algorithms in the commitment scheme would still be efficient.

Since performing tomography on a PRFS state does not violate its pseudorandomness property, the hiding property is unaffected. For achieving correctness and binding properties, we need to also equip the tomography process with a verification algorithm, denoted by Verify. A natural verification algorithm that can be associated with the tomography procedure is the following: to check if u is a valid classical description of a state σ, simply run the above tomography procedure on many copies of σ and check if the output density matrix is close to u.

More formally, we introduce a new tomography called verifiable tomography and we present a generic transformation that converts a specific tomography procedure into one that is also verifiable. We will see how verifiable tomography helps us achieve both correctness and binding. Before we dive into the new notion and understand its properties, we will first discuss the specific tomography procedure that we consider.

Instantiation. We develop a tomography procedure based on [14] that outputs a denisity matrix close (constant distance away) to the input with $1 - \mathsf{negl}(\lambda)$ probability. This is an upgrade to the tomography procedure in [14], the expected distance of whose output was a constant. To achieve this, we make use of the fact that if we repeat [14]'s tomography procedure polynomially many times, most output states cluster around the input at a constant distance with $1 - \mathsf{negl}(\lambda)$

probability. We believe this procedure might be of independent interest. Details about this procedure can be found in Section 5.2.

Verifiable Tomography. Verifiable tomography is a pair of efficient algorithms $(\mathsf{Tomography}, \mathsf{Verify})$ associated with a family of channels Φ_λ such that the following holds:

– *Same-input correctness:* Let $u_1 = \mathsf{Tomography}(\Phi_\lambda(x))$ and $u_2 = \mathsf{Tomography}(\Phi_\lambda(x))$, then $\mathsf{Verify}(u_1, u_2)$ accepts with high probability.
– *Different-input correctness:* Let $u_1 = \mathsf{Tomography}(\Phi_\lambda(x_1))$ and $u_2 = \mathsf{Tomography}(\Phi_\lambda(x_2))$, and $x_1 \neq x_2$, then $\mathsf{Verify}(u_1, u_2)$ rejects with high probability.

The family of channels we consider corresponds to the PRFS state generation. That is, $\Phi_\lambda(x = (k, i))$ outputs $G(k, i)$. As mentioned earlier, we can generically convert the above instantiation into a verifiable tomography procedure. Let us see how the generic transformation works.

For simplicity, consider the case when the underlying PRFS has perfect state generation, i.e., the output of PRFS is always a pure state. In this case, the verification algorithm is the canonical one that we described earlier: on input u and PRFS key k, input i, it first performs tomography on many copies of $G(k, i)$ to recover u' and then checks if u is close to u' or not. The same-input correctness follows from the tomography guarantee of the instantiation. To prove the different-input correctness, we use the fact that PRFS outputs are close to uniformly distributed and the following fact [2, Fact 6.9]: for two arbitrary n-qubit states $|\psi\rangle$ and $|\phi\rangle$,

$$\mathop{\mathbb{E}}_{P \xleftarrow{\$} \mathcal{P}_n} \left[|\langle\psi| P |\phi\rangle|^2 \right] = 2^{-n}.$$

Thus, if $x_1 \neq x_2$ then u_1 and u_2 are most likely going to be far and thus, differing-input correctness property is satisfied as well.

The proofs get more involved when the underlying PRFS does not satisfy perfect state generation. We consider PRFS generators that satisfy recognisable abort; we note that this notion of PRFS can be instantiated from PRS, also with $O(\log(\lambda))$ outpout length, using [2]. A $(d(\lambda), n(\lambda))$-PRFS generator G has the *strongly recognizable abort property* if its output can be written as follows: $G_\lambda(k, x) = Tr_{\mathcal{A}} \left(\eta |0\rangle\langle 0| \otimes |\psi\rangle\langle\psi| + (1 - \eta) |\bot\rangle\langle\bot| \right)$, where $\mathcal{A}$ is the register with the first qubit. Moreover, $|\bot\rangle$ is of the form $|1\rangle |\widehat{\bot}\rangle$ for some $n(\lambda)$-qubit state state $|\widehat{\bot}\rangle$ so that, $(\langle 0| \otimes \langle\psi|)(|\bot\rangle) = 0$. The same-input correctness essentially follows as before; however arguing differing-input correctness property seems more challenging.

Arguing different-input correctness is more tricky. Consider the following degenerate case: suppose k be a key and x_1, x_2 be two inputs such that PRFS on input (k, x_1) and PRFS on (k, x_2) abort with very high probability (say, close to 1). Note that the recognizable abort property does not rule out this degenerate case. Then, it holds that the outputs $u_1 = \mathsf{Tomography}(\Phi_\lambda(x_1))$ and $u_2 = \mathsf{Tomography}(\Phi_\lambda(x_2))$ are close. $\mathsf{Verify}(u_1, u_2)$ accepts and thus, the different-input correctness is not satisfied. To handle such degenerate cases, we incorporate the following into the verification procedure: on input (u_1, u_2), reject if either u_1 or u_2 is close to an abort state. Checking whether a classical description of a state is close to an abort state can be done efficiently.

From Verifiable Tomography to Commitments. Incorporating verifiable tomography into the commitment scheme, we have the following:

- The correctness follows from the same-input correctness of the tomography procedure.
- The binding property follows from the different-input correctness of the tomography procedure.
- The hiding property follows from the fact that the output of a PRFS generator is indistinguishable from Haar random, even given polynomially many copies of the state.

2 Preliminaries

We present the preliminaries in this section. We use λ to denote the security parameter. We use the notation $\mathsf{negl}(\cdot)$ to denote a negligible function.

We refer the reader to [17] for a comprehensive reference on the basics of quantum information and quantum computation. We use I to denote the identity operator. We use $\mathcal{D}(\mathcal{H})$ to denote the set of density matrices on a Hilbert space $\mathcal{H}$.

Haar Measure. The Haar measure over $\mathbb{C}^d$, denoted by $\mathscr{H}(\mathbb{C}^d)$ is the uniform measure over all d-dimensional unit vectors. One useful property of the Haar measure is that for all d-dimensional unitary matrices U, if a random vector $|\psi\rangle$ is distributed according to the Haar measure $\mathscr{H}(\mathbb{C}^d)$, then the state $U|\psi\rangle$ is also distributed according to the Haar measure. For notational convenience we write $\mathscr{H}_m$ to denote the Haar measure over m-qubit space, or $\mathscr{H}((\mathbb{C}^2)^{\otimes m})$.

Fact 1. *We have*

$$\mathop{\mathbb{E}}_{|\psi\rangle \leftarrow \mathscr{H}(\mathbb{C}^d)} |\psi\rangle\langle\psi| = \frac{I}{d} \ .$$

2.1 Distance Metrics and Matrix Norms

Trace Distance. Let $\rho, \sigma \in \mathcal{D}(\mathcal{H})$ be density matrices. We write $\mathrm{TD}(\rho, \sigma)$ to denote the trace distance between them, i.e.,

$$\mathrm{TD}(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_1$$

where $\|X\|_1 = \mathrm{Tr}(\sqrt{X^\dagger X})$ denotes the trace norm.

We denote $\|X\| := \sup_{|\psi\rangle}\{\langle\psi|X|\psi\rangle\}$ to be the operator norm where the supremum is taken over all unit vectors. For a vector x, we denote its Euclidean norm to be $\|x\|_2$.

Frobenius Norm. The Frobenius norm of a matrix M is

$$\|M\|_F = \sqrt{\sum_{i,j} |M_{i,j}|^2} = \sqrt{\mathrm{Tr}\,(MM^\dagger)},$$

where $M_{i,j}$ denotes the $(i,j)^{th}$ entry of M.

We state some useful facts about Frobenius norm below.

Fact 2. *For all matrices A, B we have $\|A-B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 - 2\mathrm{Tr}(A^\dagger B)$.*

Fact 3. *Let M_0, M_1 be density matricies and $|\psi\rangle$ be a pure state such that $\langle\psi| M_0 |\psi\rangle \leq \alpha$ and $\|M_0 - M_1\|_F^2 \leq \beta$, where $\beta + 2\alpha < 1$ then*

$$\langle\psi| M_1 |\psi\rangle \leq \alpha + \sqrt{\beta} + \sqrt{(2-2\alpha)\,\beta}.$$

Proof. From fact 2, we have the following:

$$\begin{aligned}\|M_0 - |\psi\rangle\langle\psi|\,\|_F &= \sqrt{\|M_0\|_F^2 + \|\,|\psi\rangle\langle\psi|\,\|_F^2 - 2\mathrm{Tr}(M_0^\dagger\,|\psi\rangle\langle\psi|)} \\ &= \sqrt{\|M_0\|_F^2 + 1 - 2\,\langle\psi| M_0 |\psi\rangle} \\ &\geq \sqrt{\|M_0\|_F^2 + 1 - 2\alpha}.\end{aligned}$$

By triangle inequality, we know

$$\|M_1\|_F \leq \|M_0\|_F + \|M_0 - M_1\|_F \leq ||M_0||_F + \sqrt{\beta}.$$

Similarly by fact 2,

$$\begin{aligned}\|M_1 - |\psi\rangle\langle\psi|\,\|_F &= \sqrt{1 + \|M_1\|_F^2 - 2\,\langle\psi| M_1 |\psi\rangle} \\ &\leq \sqrt{1 + \left(\|M_0\|_F + \sqrt{\beta}\right)^2 - 2\,\langle\psi| M_1 |\psi\rangle}.\end{aligned}$$

By triangle inequality, we know $\|M_0 - |\psi\rangle\langle\psi|\,\|_F \leq \|M_1 - |\psi\rangle\langle\psi|\,\|_F + \|M_0 - M_1\|_F$. Hence,

$$\sqrt{1 + \|M_0\|_F^2 - 2\alpha} \leq \sqrt{1 + \left(\|M_0\|_F + \sqrt{\beta}\right)^2 - 2\,\langle\psi| M_1 |\psi\rangle} + \sqrt{\beta}.$$

By some easy manipulation, we get

$$\langle\psi| M_1 |\psi\rangle \leq \alpha + \|M_0\|_F^2\sqrt{\beta} + \sqrt{(1 + \|M_0\|_F^2 - 2\alpha)\,\beta} \leq \alpha + \sqrt{\beta} + \sqrt{(2-2\alpha)\,\beta}.$$

□

Fact 4. *For any $0 \leq \varepsilon \leq 1$,*

$$\mathsf{Pr}_{|\psi_1\rangle, |\psi_2\rangle \leftarrow \mathscr{H}_n}\left[\|\,|\psi_1\rangle\langle\psi_1| - |\psi_2\rangle\langle\psi_2|\,\|_F^2 \leq \varepsilon\right] \leq \frac{1}{e^{2^n(1-\frac{\varepsilon}{2})}}.$$

Proof. From Fact 2,

$$\begin{aligned}\| \, |\psi_1\rangle\langle\psi_1| - |\psi_2\rangle\langle\psi_2| \, \|_F^2 &= \| \, |\psi_1\rangle\langle\psi_1| \, \|_F^2 + \| \, |\psi_2\rangle\langle\psi_2| \, \|_F^2 - 2\mathrm{Tr}\left(|\psi_1\rangle\langle\psi_1| \, |\psi_2\rangle\langle\psi_2|\right) \\ &= 2 - 2|\langle\psi_1|\psi_2\rangle|^2\end{aligned}$$

Thus, we have the following:

$$\begin{aligned}\mathsf{Pr}_{|\psi_1\rangle,|\psi_2\rangle\leftarrow\mathscr{H}_n}\left[\| \, |\psi_1\rangle\langle\psi_1| - |\psi_2\rangle\langle\psi_2| \, \|_F^2 \leq \varepsilon\right] &= \mathsf{Pr}_{|\psi_1\rangle,|\psi_2\rangle\leftarrow\mathscr{H}_n}\left[|\langle\psi_1|\psi_2\rangle|^2 \geq 1 - \frac{\varepsilon}{2}\right] \\ &\leq \frac{1}{e^{2^n(1-\frac{\varepsilon}{2})}},\end{aligned}$$

where the last inequality was shown in [5] (Equation 14). □

2.2 Quantum Algorithms

A quantum algorithm A is a family of generalized quantum circuits $\{A_\lambda\}_{\lambda\in\mathbb{N}}$ over a discrete universal gate set (such as $\{CNOT, H, T\}$). By generalized, we mean that such circuits can have a subset of input qubits that are designated to be initialized in the zero state, and a subset of output qubits that are designated to be traced out at the end of the computation. Thus a generalized quantum circuit A_λ corresponds to a *quantum channel*, which is a is a completely positive trace-preserving (CPTP) map. When we write $A_\lambda(\rho)$ for some density matrix ρ, we mean the output of the generalized circuit A_λ on input ρ. If we only take the quantum gates of A_λ and ignore the subset of input/output qubits that are initialized to zeroes/traced out, then we get the *unitary part* of A_λ, which corresponds to a unitary operator which we denote by $\hat{A}_\lambda$. The *size* of a generalized quantum circuit is the number of gates in it, plus the number of input and output qubits.

We say that $A = \{A_\lambda\}_\lambda$ is a *quantum polynomial-time (QPT) algorithm* if there exists a polynomial p such that the size of each circuit A_λ is at most $p(\lambda)$. Furthermore we say that A is *uniform* if there exists a deterministic polynomial-time Turing machine M that on input 1^n outputs the description of A_λ.

We also define the notion of a *non-uniform* QPT algorithm A that consists of a family $\{(A_\lambda, \rho_\lambda)\}_\lambda$ where $\{A_\lambda\}_\lambda$ is a polynomial-size family of circuits (not necessarily uniformly generated), and for each λ there is additionally a subset of input qubits of A_λ that are designated to be initialized with the density matrix ρ_λ of polynomial length. This is intended to model non-uniform quantum adversaries who may receive quantum states as advice. Nevertheless, the reductions we show in this work are all uniform.

The notation we use to describe the inputs/outputs of quantum algorithms will largely mimic what is used in the classical cryptography literature. For example, for a state generator algorithm G, we write $G_\lambda(k)$ to denote running the generalized quantum circuit G_λ on input $|k\rangle\langle k|$, which outputs a state ρ_k.

Ultimately, all inputs to a quantum circuit are density matrices. However, we mix-and-match between classical, pure state, and density matrix notation; for example, we may write $A_\lambda(k, |\theta\rangle, \rho)$ to denote running the circuit A_λ on input

$|k\rangle\langle k| \otimes |\theta\rangle\langle\theta| \otimes \rho$. In general, we will not explain all the input and output sizes of every quantum circuit in excruciating detail; we will implicitly assume that a quantum circuit in question has the appropriate number of input and output qubits as required by context.

2.3 Pseudorandomness Notions

Next, we recall the different notions of pseudorandomness. First, in Section 2.3, we recall (classical) pseudorandom functions (prfs) and consider two notions of security associated with it. Then in Section 2.3, we define pseudorandom quantum state (PRS) generators, which are a quantum analogue of pseudorandom generators (PRGs). Finally in Section 2.3, we define pseudorandom function-like quantum state (PRFS) generators, which are a quantum analogue of pseudorandom functions. To make it less confusing to the reader, we use the abbreviation "prfs" (small letters) for classical pseudorandom functions and "PRFS" (all caps) for pseudorandom function-like states.

Pseudorandom Functions We present two security notions of pseudorandom functions. First, we consider the notion of post-quantum security, defined below.

Definition 1 (Post-quantum pseudorandom functions). *We say that a deterministic polynomial-time algorithm $F : \{0,1\}^\lambda \times \{0,1\}^{d(\lambda)} \to \{0,1\}^{n(\lambda)}$ is a* post-quantum secure pseudorandom function (pq-prf) *if for all QPT (non-uniform) distinguishers $A = (A_\lambda, \rho_\lambda)$ there exists a negligible function $\varepsilon(\cdot)$ such that the following holds:*

$$\left| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[A_\lambda^{\mathcal{O}_{\mathsf{prf}}(k,\cdot)}(\rho_\lambda) = 1 \right] - \Pr_{\mathcal{O}_{\mathsf{Rand}}} \left[A_\lambda^{\mathcal{O}_{\mathsf{Rand}}(\cdot)}(\rho_\lambda) = 1 \right] \right| \leq \varepsilon(\lambda),$$

where:

- *$\mathcal{O}_{\mathsf{prf}}(k,\cdot)$, modeled as a classical algorithm, on input $x \in \{0,1\}^{d(\lambda)}$, outputs $F(k,x)$.*
- *$\mathcal{O}_{\mathsf{Rand}}(\cdot)$, modeled as a classical algorithm, on input $x \in \{0,1\}^{d(\lambda)}$, outputs y_x, where $y_x \leftarrow \{0,1\}^{n(\lambda)}$.*

Moreover, the adversary A_λ only has classical access to $\mathcal{O}_{\mathsf{prf}}(k,\cdot)$ and $\mathcal{O}_{\mathsf{Rand}}(\cdot)$. That is, any query made to the oracle is measured in the computational basis.

We also say that F is a $(d(\lambda), n(\lambda))$-pq-prf to succinctly indicate that its input length is $d(\lambda)$ and its output length is $n(\lambda)$.

Next, we consider the quantum-query security, as considered by Zhandry [19]. In this security notion, the adversary has superposition access to either $\mathcal{O}_{\mathsf{prf}}$ or $\mathcal{O}_{\mathsf{Rand}}$. By definition, quantum-query security implies post-quantum security.

Unlike all the other pseudorandom notions considered in this section, we are going to use a different convention and allow the key length to be a polynomial in λ, instead of it being just λ. We also parameterize the advantage of the adversary.

Definition 2 (Quantum-query secure pseudorandom functions). *We say that a deterministic polynomial-time algorithm* $F : \{0,1\}^{\ell(\lambda)} \times \{0,1\}^{d(\lambda)} \to \{0,1\}^{n(\lambda)}$ *is a* quantum-query ε-secure pseudorandom function (qprf) *if for all QPT (non-uniform) distinguishers* $A = (A_\lambda, \rho_\lambda)$ *there exists a function* $\varepsilon(\cdot)$ *such that the following holds:*

$$\left| \Pr_{k \leftarrow \{0,1\}^{\ell(\lambda)}} \left[A_\lambda^{|\mathcal{O}_{\mathsf{prf}}(k,\cdot)\rangle}(\rho_\lambda) = 1 \right] - \Pr_{\mathcal{O}_{\mathsf{Rand}}} \left[A_\lambda^{|\mathcal{O}_{\mathsf{Rand}}(\cdot)\rangle}(\rho_\lambda) = 1 \right] \right| \leq \varepsilon(\lambda),$$

where:

- $\mathcal{O}_{\mathsf{prf}}(k,\cdot)$ *on input a* $(d+n)$*-qubit state on registers* $\mathbf{X}$ *(first* d *qubits) and* $\mathbf{Y}$*, applies an* $(n+d)$*-qubit unitary* U *described as follows:* $U |x\rangle |a\rangle = |x\rangle |a \oplus F(k,x)\rangle$*. It sends back the registers* $\mathbf{X}$ *and* $\mathbf{Y}$*.*
- $\mathcal{O}_{\mathsf{Rand}}(\cdot)$ *on input a* $(d+n)$*-qubit state on registers* $\mathbf{X}$ *(first* d *qubits) and* $\mathbf{Y}$*, applies an* $(n+d)$*-qubit unitary* R *described as follows:* $R |x\rangle |a\rangle = |x\rangle |a \oplus y_x\rangle$*, where* $y_x \leftarrow \{0,1\}^{n(\lambda)}$*. It sends back the registers* $\mathbf{X}$ *and* $\mathbf{Y}$*.*

Moreover, A_λ *has superposition access to* $\mathcal{O}_{\mathsf{prf}}(k,\cdot)$ *and* $\mathcal{O}_{\mathsf{Rand}}(\cdot)$*. We denote the fact that* A_λ *has quantum access to an oracle* $\mathcal{O}$ *by* $A_\lambda^{|\mathcal{O}\rangle}$*.*

We also say that F *is a* $(\ell(\lambda), d(\lambda), n(\lambda), \varepsilon)$*-qprf to succinctly indicate that its input length is* $d(\lambda)$ *and its output length is* $n(\lambda)$*. When* $\ell(\lambda) = \lambda$*, we drop* $\ell(\lambda)$ *from the notation. Similarly, when* $\varepsilon(\lambda)$ *can be any negligible function, we drop* $\varepsilon(\lambda)$ *from the notation.*

Zhandry [19] presented a construction of quantum-query secure pseudorandom functions from one-way functions.

Lemma 1 (Zhandry [19]). *Assuming post-quantum one-way functions, there exists quantum-query secure pseudorandom functions.*

Useful Lemma. We will use the following lemma due to Zhandry [20]. The lemma states that any q-query algorithm cannot distinguish (quantum) oracle access to a random function versus a $2q$-wise independent hash function. We restate the lemma using our notation.

Lemma 2 ([20, Theorem 3.1]). *Let* A *be a* q*-query algorithm. Then, for any* $d, n \in \mathbb{N}$*, every* $2q$*-wise independent hash function* $H : \{0,1\}^{\ell(q)} \times \{0,1\}^d \to \{0,1\}^n$ *satisfies the following:*

$$\left| \Pr_{k \leftarrow \{0,1\}^{\ell(q)}} \left[A_\lambda^{|\mathcal{O}_{\mathsf{H}}(k,\cdot)\rangle}(\rho_\lambda) = 1 \right] - \Pr_{\mathcal{O}_{\mathsf{Rand}}} \left[A_\lambda^{|\mathcal{O}_{\mathsf{Rand}}(\cdot)\rangle}(\rho_\lambda) = 1 \right] \right| = 0,$$

where $\mathcal{O}_{\mathsf{Rand}}$ *is as defined in Definition 2 and* $\mathcal{O}_{\mathsf{H}}$ *is defined similarly to* $\mathcal{O}_{\mathsf{prf}}$ *except that the unitary* U *uses* H *instead of* F*.*

Pseudorandom Quantum State Generators We move onto the pseudorandom notions in the quantum world. The notion of pseudorandom states were first introduced by Ji, Liu, and Song in [11]. We reproduce their definition here:

Definition 3 (PRS Generator [11]). *We say that a QPT algorithm G is a* pseudorandom state (PRS) generator *if the following holds.*

1. ***State Generation**. For all λ and for all $k \in \{0,1\}^\lambda$, the algorithm G behaves as*
$$G_\lambda(k) = |\psi_k\rangle\langle\psi_k| \,.$$
for some $n(\lambda)$-qubit pure state $|\psi_k\rangle$.
2. ***Pseudorandomness**. For all polynomials $t(\cdot)$ and QPT (nonuniform) distinguisher A there exists a negligible function $\varepsilon(\cdot)$ such that for all λ, we have*
$$\left| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[A_\lambda(G_\lambda(k)^{\otimes t(\lambda)}) = 1 \right] - \Pr_{|\vartheta\rangle \leftarrow \mathscr{H}_{n(\lambda)}} \left[A_\lambda(|\vartheta\rangle^{\otimes t(\lambda)}) = 1 \right] \right| \le \varepsilon(\lambda) \ .$$

We also say that G is a $n(\lambda)$-PRS generator to succinctly indicate that the output length of G is $n(\lambda)$.

Ji, Liu, and Song showed that post-quantum one-way functions can be used to construct PRS generators.

Theorem 5 ([11,4]). *If post-quantum one-way functions exist, then there exist PRS generators for all polynomial output lengths.*

Pseudorandom Function-Like State (PRFS) Generators In this section, we recall the definition of pseudorandom function-like state (PRFS) generators by Ananth, Qian and Yuen [2]. PRFS generators generalize PRS generators in two ways: first, in addition to the secret key k, the PRFS generator additionally takes a (classical) input x. The second way in which this definition generalizes the definition of PRS generators is that the output of the generator need not be a pure state.

However, they considered the weaker selective security definition where the adversary needs to choose all the inputs to be queried to the PRFS ahead of time. Later we will introduce the stronger and the more useful definition of adaptive security.

Definition 4 (Selectively Secure PRFS generator). *We say that a QPT algorithm G is a (selectively secure)* pseudorandom function-like state (PRFS) generator *if for all polynomials $s(\cdot), t(\cdot)$, QPT (nonuniform) distinguishers A and a family of indices $\left(\{x_1, \dots, x_{s(\lambda)}\} \subseteq \{0,1\}^{d(\lambda)}\right)_\lambda$, there exists a negligible function $\varepsilon(\cdot)$ such that for all λ,*

$$\Big| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[A_\lambda(x_1, \dots, x_{s(\lambda)}, G_\lambda(k, x_1)^{\otimes t(\lambda)}, \dots, G_\lambda(k, x_{s(\lambda)})^{\otimes t(\lambda)}) = 1 \right]$$
$$- \Pr_{|\vartheta_1\rangle, \dots, |\vartheta_{s(\lambda)}\rangle \leftarrow \mathscr{H}_{n(\lambda)}} \left[A_\lambda(x_1, \dots, x_{s(\lambda)}, |\vartheta_1\rangle^{\otimes t(\lambda)}, \dots, |\vartheta_{s(\lambda)}\rangle^{\otimes t(\lambda)}) = 1 \right] \Big| \le \varepsilon(\lambda) \ .$$

We say that G is a $(d(\lambda), n(\lambda))$-PRFS generator to succinctly indicate that its input length is $d(\lambda)$ and its output length is $n(\lambda)$.

Our notion of security here can be seen as a version of *(classical) selective security*, where the queries to the PRFS generator are fixed before the key is sampled.

State Generation Guarantees. Towards capturing a natural class of PRFS generators, [2] introduced the concept of *recognizable abort*. At a high level, recognizable abort is the property that the output of PRFS can be written as a convex combination of a pure state and a known abort state, denoted by $|\perp\rangle$. In more detail, the PRFS generator works in two stages. In the first stage it either generates a valid PRFS state $|\psi\rangle$ or it aborts. If it outputs a valid PRFS state then the first qubit is set to $|0\rangle$ and if it aborts, the entire state is set to $|\perp\rangle$. We have the guarantee that $|0\rangle|\psi\rangle$ is orthogonal to $|\perp\rangle$. In the next stage, the PRFS generator traces out the first qubit and outputs the resulting state. Our definition could be useful to capture many generators that don't always succeed in generating the pseudorandom state; for example, Brakerski and Shmueli [4] design generators that doesn't always succeed in generating the state.

We formally define the notion of recognizable abort[10] below.

Definition 5 (Recognizable abort). *A $(d(\lambda), n(\lambda))$-PRFS generator G has the* strongly recognizable abort property *if there exists an algorithm $\widehat{G}$ and a special $(n(\lambda)+1)$-qubit state $|\perp\rangle$ such that $G_\lambda(k,x)$ has the following form: it takes as input $k \in \{0,1\}^\lambda$, $x \in \{0,1\}^{d(\lambda)}$ and does the following,*

- *Compute $\widehat{G}_\lambda(k,x)$ to obtain an output of the form $\eta\,|0\rangle\langle 0| \otimes |\psi\rangle\langle\psi| + (1-\eta)\,|\perp\rangle\langle\perp|$ and moreover, $|\perp\rangle$ is of the form $|1\rangle\,|\widehat{\perp}\rangle$ for some $n(\lambda)$-qubit state state $|\widehat{\perp}\rangle$. As a consequence, $(\langle 0| \otimes \langle\psi|)(|\perp\rangle) = 0$.*
- *Trace out the first bit of $\widehat{G}_\lambda(k,x)$ and output the resulting state.*

As observed by [2], the definition alone does not have any constraint on η being close to 1. The security guarantee of a PRFS generator implies that η will be negligibly close to 1 with overwhelming probability over the choice of k [2, Lemma 3.6].

3 Adaptive Security

The previous work by [2] only considers PRFS that is selectively secure. That is, the adversary needs to declare the input queries ahead of time. For many applications, selective security is insufficient. For example, in the application of PRFS to secret-key encryption (satisfying multi-message security), the resulting

[10] We note that [2] define a slightly weaker definition of recognizable abort. However, the definitions and results considered in [2] also work with our (stronger) definition of recognizable abort.

scheme was also only proven to be selectively secure, whereas one could ask for security against adversaries that can make *adaptive* queries to the PRFS generator. Another drawback of the notion considered by [2] is the assumption that the adversary can make classical queries to the challenger who either returns PRFS states or independent Haar random states, whereas one would ideally prefer security against adversaries that can make *quantum superposition* queries.

In this work, we consider stronger notions of security for PRFS. We strengthen the definitions of [2] in two ways. First, we allow the the adversary to make adaptive queries to the PRFS oracle, and second, we allow the adversary to make *quantum* queries to the oracle. The oracle model we consider here is slightly different from the usual quantum query model. In the usual model, there is an underlying function f and the oracle is modelled as a unitary acting on two registers, a *query* register $\mathbf{X}$ and an *answer* register $\mathbf{Y}$ mapping basis states $|x\rangle_{\mathbf{X}} \otimes |y\rangle_{\mathbf{Y}}$ to $|x\rangle_{\mathbf{X}} \otimes |y \oplus f(x)\rangle_{\mathbf{Y}}$ (in other words, the function output is XORed with answer register in the standard basis). The query algorithm also acts on the query and answer registers; indeed, it is often useful in quantum algorithms to initialize the answer register to something other than all zeroes.

In the PRS/PRFS setting, however, there is no underlying classical function: the output of the PRFS generator G could be an entangled pseudorandom state far from any standard basis state; it seems unnatural to XOR the pseudorandom the state with a standard basis state. Instead we consider a model where the query algorithm submits a query register $\mathbf{X}$ to the oracle, and the oracle returns the query register $\mathbf{X}$ as well as an answer register $\mathbf{Y}$. If the algorithm submits query $|x\rangle_{\mathbf{X}}$, then the joint state register $\mathbf{XY}$ after the query is $|x\rangle_{\mathbf{X}} \otimes |\psi_x\rangle_{\mathbf{Y}}$ for some pure state $|\psi_x\rangle$. Each time the algorithm makes a query, the oracle returns a fresh answer register. Thus, the number of qubits that the query algorithm acts on grows with the number of queries.[11]

How the oracle behaves when the query algorithm submits a superposition $\sum_x \alpha_x |x\rangle_{\mathbf{X}}$ in the query register is a further modeling choice. In the most general setting, the oracle behaves as a unitary on registers $\mathbf{XY}$,[12] and the resulting state of the query and answer registers is $\sum_x \alpha_x |x\rangle_{\mathbf{X}} \otimes |\psi_x\rangle_{\mathbf{Y}}$. That is, queries are answered in superposition. We call such an oracle *quantum-accessible*.

We also consider the case where the queries are forced to be *classical*, which may already be useful for some applications. Here, the oracle is modeled as a channel (instead of a unitary) that first measures the query register in the standard basis before returning the corresponding state $|\psi_x\rangle$. In other words, if the query is $\sum_x \alpha_x |x\rangle_{\mathbf{X}}$, then the resulting state becomes the mixed state $\sum_x |\alpha_x|^2 |x\rangle\langle x|_{\mathbf{X}} \otimes |\psi_x\rangle\langle\psi_x|_{\mathbf{Y}}$. This way, the algorithm cannot take advantage of quantum queries – but it can still make queries adaptively. We call such an oracle *classically-accessible*.

[11] Alternatively, one can think of answer registers $\mathbf{Y}_1, \mathbf{Y}_2, \ldots$ as being initialized in the zeroes state at the beginning, and the query algorithm is only allowed to act nontrivially on $\mathbf{Y}_i$ after the i'th query.

[12] Alternatively, one can think of the oracle as an *isometry* mapping register $\mathbf{X}$ to registers $\mathbf{XY}$.

To distinguish between classical and quantum access to oracles, we write $A^{\mathcal{O}}$ to denote a quantum algorithm that has classical access to the oracle $\mathcal{O}$, and $A^{|\mathcal{O}\rangle}$ to denote a quantum algorithm that has quantum access to the oracle $\mathcal{O}$.

3.1 Classical Access

We define adaptively secure PRFS, where the adversary is given *classical access* to the PRFS/Haar-random oracle.

Definition 6 (Adaptively-Secure PRFS). *We say that a QPT algorithm G is an* adaptively secure pseudorandom function-like state (APRFS) *generator if for all QPT (non-uniform) distinguishers A, there exists a negligible function ε, such that for all λ, the following holds:*

$$\left| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[A_\lambda^{\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)}(\rho_\lambda) = 1 \right] - \Pr_{\mathcal{O}_{\mathsf{Haar}}} \left[A_\lambda^{\mathcal{O}_{\mathsf{Haar}}(\cdot)}(\rho_\lambda) = 1 \right] \right| \leq \varepsilon(\lambda),$$

where:

- $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$*, on input* $x \in \{0,1\}^{d(\lambda)}$*, outputs* $G_\lambda(k,x)$*.*
- $\mathcal{O}_{\mathsf{Haar}}(\cdot)$*, on input* $x \in \{0,1\}^{d(\lambda)}$*, outputs* $|\vartheta_x\rangle$*, where, for every* $y \in \{0,1\}^{d(\lambda)}$*,* $|\vartheta_y\rangle \leftarrow \mathscr{H}_{n(\lambda)}$*.*

Moreover, the adversary A_λ has classical access to $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$ and $\mathcal{O}_{\mathsf{Haar}}(\cdot)$. That is, we can assume without loss of generality that any query made to either oracle is measured in the computational basis.

We say that G is a $(d(\lambda), n(\lambda))$-APRFS generator to succinctly indicate that its input length is $d(\lambda)$ and its output length is $n(\lambda)$.

Some remarks are in order.

Instantiation. For the case when $d(\lambda) = O(\log(\lambda))$, selectively secure PRFS is equivalent to adaptively secure PRFS. The reason being that we can assume without loss of generality, the selective adversary can query on all possible inputs (there are only polynomially many) and use the outputs to simulate the adaptive adversary. As a consequence of the result that log-input selectively-secure PRFS can be built from PRS [2], we obtain the following.

Lemma 3. *For $d = O(\log(\lambda))$ and $n = d + \omega(\log\log\lambda)$, assuming the existence of $(d+n)$-PRS, there exists a (d,n)-APRFS.*

In the case when $d(\lambda)$ is an arbitrary polynomial in λ, we present a construction of APRFS from post-quantum one-way functions in the full version of the paper.

Test procedure. It was shown by [2] that a PRFS admits a Test procedure (See Section 3.3 in [2]). The goal of a Test procedure is to determine whether the given state is a valid PRFS state or not. Having a Test procedure is useful in applications. For example, [2] used a Test procedure in the construction of a bit commitment scheme. We note that the same Test procedure also works for adaptively secure PRFS.

Multiple copies. In the definition of PRS (Definition 3) and selectively-secure PRFS (Definition 4), the adversary is allowed to obtain multiple copies of the same pseudorandom (or haar random) quantum state. While we do not explicitly state it, even in Definition 6, the adversary can indeed obtain multiple copies of a (pseudorandom or haar random) quantum state. To obtain t copies of the output of $G_\lambda(k,x)$ (or $|\vartheta_x\rangle$), the adversary can query the same input x, t times, to the oracle $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$ (or $\mathcal{O}_{\mathsf{Haar}}(\cdot)$).

3.2 Quantum Access

We further strengthen our notion of adaptively secure PRFS by allowing the adversary to make superposition queries to either $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$ or $\mathcal{O}_{\mathsf{Haar}}(\cdot)$. Providing superposition access to the adversary not only makes the definition stronger[13] than Definition 6 but is also arguably more useful for a larger class of applications. To indicate quantum query access, we put the oracle inside the ket notation: $A^{|\mathcal{O}\rangle}$ (whereas for classical query access we write $A^{\mathcal{O}}$).

We provide the formal definition below.

Definition 7 (Quantum-accessible Adaptively-secure PRFS). *We say that a QPT algorithm G is a quantum-accessible adaptively secure pseudorandom function-like state (QAPRFS) generator if for all QPT (non-uniform) distinguishers A if there exists a negligible function ε, such that for all λ, the following holds:*

$$\left| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[A_\lambda^{|\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)\rangle}(\rho_\lambda) = 1 \right] - \Pr_{\mathcal{O}_{\mathsf{Haar}}} \left[A_\lambda^{|\mathcal{O}_{\mathsf{Haar}}(\cdot)\rangle}(\rho_\lambda) = 1 \right] \right| \leq \varepsilon(\lambda),$$

where:

- $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$*, on input a d-qubit register* $\mathbf{X}$*, does the following: it applies a channel that controlled on the register* $\mathbf{X}$ *containing x, it creates and stores* $G_\lambda(k,x)$ *in a new register* $\mathbf{Y}$*. It outputs the state on the registers* $\mathbf{X}$ *and* $\mathbf{Y}$*.*
- $\mathcal{O}_{\mathsf{Haar}}(\cdot)$*, modeled as a channel, on input a d-qubit register* $\mathbf{X}$*, does the following: it applies a channel that controlled on the register* $\mathbf{X}$ *containing x, stores* $|\vartheta_x\rangle\langle\vartheta_x|$ *in a new register* $\mathbf{Y}$*, where* $|\vartheta_x\rangle$ *is sampled from the Haar distribution. It outputs the state on the registers* $\mathbf{X}$ *and* $\mathbf{Y}$*.*

Moreover, A_λ has superposition access to $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$ and $\mathcal{O}_{\mathsf{Haar}}(\cdot)$.

We say that G is a $(d(\lambda), n(\lambda))$-QAPRFS generator to succinctly indicate that its input length is $d(\lambda)$ and its output length is $n(\lambda)$.

We present a construction satisfying the above definition in the full version of the paper.

Unlike Definition 6, it is not without loss of generality that A_λ can get multiple copies of a quantum state. To illustrate, consider an adversary that submits

[13] It is stronger in the sense that an algorithm that has quantum query access to the oracle can simulate an algorithm that only has classical query access.

a state of the form $\sum_x \alpha_x |x\rangle$ to the oracle. It then gets back $\sum_x \alpha_x |x\rangle |\psi_x\rangle$ (where $|\psi_x\rangle$ is either the output of PRFS[14] or it is Haar random) instead of $\sum_x \alpha_x |x\rangle |\psi_x\rangle^{\otimes t}$, for some polynomial t. On the other hand, if the adversary can create multiple copies of $\sum_x \alpha_x |x\rangle$, the above definition allows the adversary to obtain $(\sum_x \alpha_x |x\rangle |\psi_x\rangle)^{\otimes t}$ for any polynomial $t(\cdot)$ of its choice.

4 On the Necessity of Computational Assumptions

The following lemma shows that the security guarantee of a PRS generator (and thus of PRFS generators) can only hold with respect to computationally bounded distinguishers, provided that the output length is at least $\log \lambda$.

Lemma 4. *Let G be a PRS generator with output length $n(\lambda) \geq \log \lambda$. Then there exists a polynomial $t(\lambda)$ and a quantum algorithm A (not efficient in general) such that*

$$\left| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[A_\lambda \left(G_\lambda(k)^{\otimes t(\lambda)} \right) = 1 \right] - \Pr_{|\vartheta\rangle \leftarrow \mathscr{H}_{n(\lambda)}} \left[A_\lambda \left(|\vartheta\rangle\langle\vartheta|^{\otimes t(\lambda)} \right) = 1 \right] \right| \geq \frac{1}{3}$$

for all sufficiently large λ.

Proof. For notational convenience we abbreviate $n = n(\lambda)$ and $t = t(\lambda)$. We split the proof into two cases.

Case 1: if there does not exist a negligible function $\nu(\cdot)$ such that

$$\Pr_k \left[\min_{|\theta\rangle} \mathrm{TD}(G_\lambda(k), |\theta\rangle\langle\theta|) \leq \nu(\lambda) \right] \geq \frac{1}{2}. \tag{2}$$

Then there exists some non-negligible function $\kappa(\cdot)$ such that with probability at least $\frac{1}{2}$ over the choice of k, $\min_{|\theta\rangle} \mathrm{TD}(G_\lambda(k), |\theta\rangle\langle\theta|) \geq \kappa(\lambda)$. Let $\nu_{k,1} \geq ... \geq \nu_{k,2^n}$ and $|\alpha_{k,1}\rangle, ..., |\alpha_{k,2^n}\rangle$ be eigenvalues and eigenvectors for $G_\lambda(k)$. Then $\kappa \leq \mathrm{TD}(G_\lambda(k), |\alpha_{k,1}\rangle\langle\alpha_{k,1}|) = \frac{1}{2}(1 - \nu_{k,1} + \nu_{k,2} + \cdots + \nu_{k,2^n}) = 1 - \nu_{k,1}$. Thus by Hölder's inequality, $\mathrm{Tr}(G_\lambda(k)^2) \leq 1 - \kappa$. Therefore, a purity test using $t = O(1/\kappa(\lambda))$ copies will correctly reject PRS states with probability at least $\frac{1}{3}$ but never incorrectly reject any Haar random state.

Case 2: if there exists a negligible function $\nu(\cdot)$ such that (2) holds. There exists a polynomial $t(\lambda)$ such that

$$2^\lambda \leq \frac{1}{6} \cdot \dim \mathit{\Pi}_{\mathsf{sym}}^{2^n,t} = \frac{1}{6} \cdot \binom{2^n + t - 1}{t}$$

for all sufficiently large λ. This is because by setting $t = \lambda + 1$, we can lower bound the dimension of $\mathit{\Pi}_{\mathsf{sym}}^{2^n,t}$ by $\binom{2\lambda}{\lambda+1}$ and

$$\binom{2\lambda}{\lambda} \geq \frac{\lambda}{\lambda+1} \frac{4^\lambda}{\sqrt{\pi\lambda}} \left(1 - \frac{1}{8\lambda}\right)$$

[14] In this illustration, we are pretending that the PRFS satisfies perfect state generation property. That is, the output of PRFS is always a pure state.

which is much larger than $6 \cdot 2^\lambda$ for all sufficiently large λ.

Let $g \subseteq \{0,1\}^\lambda$ be the set of k's such that $\min_{|\theta\rangle} \mathrm{TD}(G_\lambda(k), |\theta\rangle\langle\theta|) \leq \nu(\lambda)$. Note that 2^λ is an upper bound on the rank of the density matrix

$$\mathop{\mathbb{E}}_{k \leftarrow g} |\psi_k\rangle\langle\psi_k|^{\otimes t}, \tag{3}$$

where $|\psi_k\rangle = \arg\min_{|\theta\rangle} \mathrm{TD}(G_\lambda(k), |\theta\rangle\langle\theta|)$. Note that the rank of the density matrix

$$\mathop{\mathbb{E}}_{|\vartheta\rangle \leftarrow \mathscr{H}_{n(\lambda)}} |\vartheta\rangle\langle\vartheta|^{\otimes t} = \frac{\Pi_{\mathsf{sym}}^{2^n,t}}{\dim \Pi_{\mathsf{sym}}^{2^n,t}} \tag{4}$$

is equal to $\dim \Pi_{\mathsf{sym}}^{2^n,t}$.

For all λ, define the quantum circuit A_λ that, given a state on tn qubits, performs the two-outcome measurement $\{P, I-P\}$ where P is the projector onto the support of $\mathbb{E}_{k\leftarrow g} |\psi_k\rangle\langle\psi_k|^{\otimes t}$, and accepts if the P outcome occurs.

By assumption of case 2, given the density matrix (3) the circuit A_λ will accept with probability at least $\frac{1}{2}$. On the other hand, given the density matrix (4) the circuit A_λ will accept with probability

$$\mathrm{Tr}\left(P \cdot \frac{\Pi_{\mathsf{sym}}^{2^n,t}}{\dim \Pi_{\mathsf{sym}}^{2^n,t}}\right) \leq \mathrm{Tr}\left(\frac{P}{\dim \Pi_{\mathsf{sym}}^{2^n,t}}\right) = \frac{\mathrm{rank}(P)}{\dim \Pi_{\mathsf{sym}}^{2^n,t}} \leq \frac{1}{6}.$$

Letting $A = \{A_\lambda\}_\lambda$ we obtained the desired Lemma statement. $\square$

We remark that the attack given in Lemma 4 cannot be used on smaller output length, up to additive factors of superpolynomially smaller order in the output length. Suppose $n = \log\lambda - \omega(\log\log\lambda)$ and for any $t = \lambda^{O(1)}$,

$$\begin{aligned}\log \binom{2^n+t-1}{t} &\leq 2^n \cdot \log \frac{e(2^n+t-1)}{2^n-1} \\ &= \frac{\lambda}{\omega(\log\lambda)} \cdot O(\log\lambda).\end{aligned}$$

This means that $\binom{2^n+t-1}{t} = 2^{\lambda/\omega(\log\lambda)} \ll 2^\lambda$ and therefore the attack above does not necessarily apply. Indeed, Brakerski and Shmueli [4] have shown that PRS generators with output length $n(\lambda) \leq c\log\lambda$ for some $c > 0$ can be achieved with statistical security.

We conclude the section by remarking that the result of Kretschmer [12] can be easily generalized so that PRS generators with output length at least $\log\lambda + c$ (for some small constant $0 < c < 2$) imply $\mathsf{BQP} \neq \mathsf{PP}$ as well[15].

[15] For readers familiar with [12], it can be verified that a sufficient condition for that proof to go through is if $2^\lambda \cdot e^{-2^n/3}$ is negligible, which is satisfied if $n \geq \log\lambda + 2$.

5 Tomography with Verification

Quantum state tomography (or just *tomography* for short) is a process that takes as input multiple copies of a quantum state ρ and outputs a string u that is a classical description of the state ρ; for example, u can describe an approximation of the density matrix ρ, or it could be a a more succinct description such as a *classical shadow* in the sense of [10]. In this paper, we use tomography as a tool to construct protocols based on pseudorandom states with only *classical* communication.

For our applications, we require tomography procedures satisfying a useful property called verification. Suppose we execute a tomography algorithm on multiple copies of a state to obtain a classical string u. The verification algorithm, given u and the algorithm to create this state, checks if u is consistent with this state or not. Verification comes in handy when tomography is used in cryptographic settings, where we would like to make sure that the adversary has generated the classical description associated with a quantum state according to some prescribed condition (this will be implictly incorporated in the verification algorithm).

Verifiable Tomography. Let $\mathcal{C} = \{\Phi_\lambda \ : \ \lambda \in \mathbb{N}\}$ be a family of channels where each channel Φ_λ takes as input $\ell(\lambda)$ qubits for some polynomial $\ell(\cdot)$. A *verifiable tomography scheme* associated with $\mathcal{C}$ is a pair $(\mathsf{Tomography}, \mathsf{Verify})$ of QPT algorithms, which have the following input/output behavior:

- $\mathsf{Tomography}$: given as input a quantum state $\rho^{\otimes L}$ for some density matrix ρ and some number L, output a classical string u (called a *tomograph* of ρ).
- Verify: given as input a pair of classical strings $(\mathbf{x}, u)$ where $\mathbf{x}$ has length $\ell(\lambda)$, output Valid or $\mathsf{Invalid}$.

We would like $(\mathsf{Tomography}, \mathsf{Verify})$ to satisfy correctness which we describe next.

5.1 Correctness Notions for Verifiable Tomography

We can consider two types of correctness. The first type of correctness, referred to as *same-input correctness*, states that $\mathsf{Verify}(\mathbf{x}, u)$ outputs Valid if u is obtained by running the $\mathsf{Tomography}$ procedure on copies of the output of $\Phi_\lambda(\mathbf{x})$. The second type of correctness, referred to as *different-input correctness*, states that $\mathsf{Verify}(\mathbf{x}', u)$ outputs $\mathsf{Invalid}$ if u is obtained by applying tomography to $\Phi_\lambda(\mathbf{x})$, where $(\mathbf{x}', \mathbf{x})$ do not satisfy a predicate Π.

Same-Input Correctness. Consider the following definition.

Definition 8 (Same-Input Correctness). *We say that* $(\mathsf{Tomography}, \mathsf{Verify})$ *satisfies* L-same-input correctness*, for some polynomial* $L(\cdot)$*, such that for every* $\mathbf{x} \in \{0,1\}^{\ell(\lambda)}$*, if the following holds:*

$$\Pr\left[\mathsf{Valid} \leftarrow \mathsf{Verify}\left(\mathbf{x}, \mathsf{Tomography}\left((\Phi_\lambda(\mathbf{x}))^{\otimes L(\lambda)}\right)\right)\right] \geq 1 - \mathsf{negl}(\lambda),$$

For some applications, it suffices to consider a weaker definition. Instead of requiring the correctness guarantee to hold for every input, we instead require that it holds over some input distribution.

Definition 9 (Distributional Same-Input Correctness). *We say that* $(\mathsf{Tomography}, \mathsf{Verify})$ *satisfies* $(L, \mathcal{D})$-distributional same-input correctness*, for some polynomial* $L(\cdot)$ *and distribution* $\mathcal{D}$ *on* $\ell(\lambda)$*-length strings, if the following holds:*

$$\Pr\left[\mathsf{Valid} \leftarrow \mathsf{Verify}\left(\mathbf{x}, \mathsf{Tomography}\left((\Phi_\lambda(\mathbf{x}))^{\otimes L(\lambda)}\right)\right) \ : \ \mathbf{x} \leftarrow \mathcal{D}\right] \geq 1 - \mathsf{negl}(\lambda)$$

Different-Input Correctness. Ideally, we would require that $\mathsf{Verify}(\mathbf{x}, u)$ outputs $\mathsf{Invalid}$ if u is produced by tomographing $\Phi_\lambda(\mathbf{x}')$, and $\mathbf{x}'$ is any string such that $\mathbf{x}' \neq \mathbf{x}$. However, for applications, we only require that this be the case when the pair $(\mathbf{x}, \mathbf{x}')$ satisfy a relation defined by a predicate Π. In other words, we require $\mathsf{Verify}(\mathbf{x}, u)$ outputs $\mathsf{Invalid}$ only when u is a tomograph of $\Phi_\lambda(\mathbf{x}')$ and $\Pi(\mathbf{x}', \mathbf{x}) = 0$.

We define this formally below.

Definition 10 (Different-Input Correctness). *We say that* $(\mathsf{Tomography}, \mathsf{Verify})$ *satisfies* (L, Π)-different-input correctness*, for some polynomial* $L(\cdot)$ *and predicate* $\Pi : \{0,1\}^{\ell(\lambda)} \times \{0,1\}^{\ell(\lambda)} \to \{0,1\}$*, such that for every* $\mathbf{x}, \mathbf{x}' \in \{0,1\}^{\ell(\lambda)}$ *satisfying* $\Pi(\mathbf{x}, \mathbf{x}') = 0$*, if the following holds:*

$$\Pr\left[\mathsf{Invalid} \leftarrow \mathsf{Verify}\left(\mathbf{x}', \mathsf{Tomography}\left((\Phi_\lambda(\mathbf{x}))^{\otimes L(\lambda)}\right)\right)\right] \geq 1 - \mathsf{negl}(\lambda)$$

Analogous to Definition 9, we correspondingly define below the notion of $(L, \mathcal{D}, \Pi)$-different-input correctness.

Definition 11 (Distributional Different-Input Correctness). *We say that* $(\mathsf{Tomography}, \mathsf{Verify})$ *satisfies* $(L, \Pi, \mathcal{D})$-distributional different-input correctness*, for some polynomial* $L(\cdot)$*, predicate* $\Pi : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}$ *and distribution* $\mathcal{D}$ *supported on* $(\mathbf{x}, \mathbf{x}') \in \{0,1\}^{\ell(\lambda)} \times \{0,1\}^{\ell(\lambda)}$ *satisfying* $\Pi(\mathbf{x}, \mathbf{x}') = 0$*, if the following holds:*

$$\Pr_{(\mathbf{x},\mathbf{x}') \leftarrow \mathcal{D}}\left[\mathsf{Invalid} \leftarrow \mathsf{Verify}\left(\mathbf{x}', \mathsf{Tomography}\left((\Phi_\lambda(\mathbf{x}))^{\otimes L(\lambda)}\right)\right)\right] \geq 1 - \mathsf{negl}(\lambda)$$

Sometimes we will use the more general $(\varepsilon, L, \Pi, \mathcal{D})$*-distributional different-input correctness* definition. In this case, the probability of Verify outputting $\mathsf{Invalid}$ is bounded below by $1 - \varepsilon$ instead of $1 - \mathsf{negl}(\lambda)$.

5.2 Verifiable Tomography Procedures

We will consider two different instantiations of $(\mathsf{Tomography}, \mathsf{Verify})$ where the first instantiation will be useful for bit commitments and the second instantiation will be useful for pseudo one-time pad schemes.

In both the instantiations, we use an existing tomography procedure stated in the lemma below.

Lemma 5 (Section 1.5.3, [14]). *There exists a tomography procedure $\mathcal{T}$ that given sN^2 copies of an N-dimensional density matrix ρ, outputs a matrix M such that $\mathbb{E}\|M-\rho\|_F^2 \leq \frac{N}{s}$ where the expectation is over the randomness of the tomography procedure. Moreover, the running time of $\mathcal{T}$ is polynomial in s and N.*

We state and prove a useful corollary of the above lemma.

Corollary 1. *There exists a tomography procedure $\mathcal{T}_{\mathsf{imp}}$ that given $4sN^2\lambda$ copies of an N-dimensional density matrix ρ, outputs a matrix M such that the following holds:*

$$\mathsf{Pr}\left[\|M-\rho\|_F^2 \leq \frac{9N}{s}\right] \geq 1-\mathsf{negl}(\lambda)$$

Moreover, the running time of $\mathcal{T}_{\mathsf{imp}}$ is polynomial in s, N and λ.

The proof of this corollary can be found in the full version.

First Instantiation We will work with a verifiable tomography procedure that will be closely associated with a PRFS. In particular, we will use a $(d(\lambda), n(\lambda))$-PRFS $\{G_\lambda(\cdot,\cdot)\}$ satisfying recognizable abort property (Definition 5). Let $\widehat{G}$ be the QPT algorithm associated with G according to Definition 5. Note that the output length of $\widehat{G}$ is $n+1$. We set $d(\lambda) = \lceil \frac{\log(\lambda)}{\log(\log(\lambda))} \rceil$ and $n(\lambda) = \lceil 3\log(\lambda) \rceil$.

We will describe the algorithms $(\mathsf{Tomography}, \mathsf{Verify})$ in Figure 1. The set of channels $\mathcal{C} = \{\Phi_\lambda : \lambda \in \mathbb{N}\}$ is associated with $(\mathsf{Tomography}, \mathsf{Verify})$, where Φ_λ is defined as follows:

- Let the input be initialized on register $\mathbf{A}$.
- Controlled on the first register containing the value (P_x, k, x, b), where P_x is an n-qubit Pauli, $k \in \{0,1\}^\lambda, b \in \{0,1\}$, do the following: compute $\left(I \otimes P_x^b\right) \widehat{G}_\lambda(k,x) \left(I \otimes P_x^b\right)$ and store it in the register $\mathbf{B}$.
- Trace out $\mathbf{A}$ and output $\mathbf{B}$.

The channel Φ_λ can be represented as a quantum circuit of size polynomial in λ as the PRFS generator $\hat{G}$ runs in time polynomial in λ.

Distributional Same-Input Correctness. We prove below that $(\mathsf{Tomography}, \mathsf{Verify})$ satisfies distributional same-input correctness. For every $x \in \{0,1\}^{d(\lambda)}$, for every n-qubit Pauli P_x and $b \in \{0,1\}$, define the distribution $\mathcal{D}_{P_x,x,b}$ as follows: sample $k \xleftarrow{\$} \{0,1\}^\lambda$ and output $\mathbf{x} = (P_x, k, x, b)$.

Lemma 6. *Let $L = O(2^{3n}\lambda)$. The verifiable tomography scheme $(\mathsf{Tomography}, \mathsf{Verify})$ described in Figure 1 satisfies $(L, \mathcal{D}_{P_x,x,b})$-distributional same-input correctness for all P_x, x, b.*

The proof of this lemma can be found in the full version.

$\underline{\mathsf{Tomography}(\rho^{\otimes L})}$: On input L copies of an $2^{(n+1)}$-dimensional density matrix ρ, compute $\mathcal{T}_{\mathsf{imp}}(\rho^{\otimes L})$ to obtain M, where $\mathcal{T}_{\mathsf{imp}}$ is given in Corollary 1. Output M.

$\underline{\mathsf{Verify}(\mathbf{x}, M)}$:

1. Run $\rho^{\otimes L} \leftarrow (\Phi_\lambda(\mathbf{x}))^{\otimes L}$, where $L = 3^8 2^{3(n+1)+2}\lambda$.
2. Compute $\widehat{M} \leftarrow \mathsf{Tomography}\left(\rho^{\otimes L}\right)$.
3. If $\langle \perp | M | \perp \rangle > \frac{1}{9}$ for any $x \in \{0,1\}^d$, output Invalid.
4. If $\|M - \widehat{M}\|_F^2 \leq \frac{4}{729}$ output Valid. Output Invalid otherwise.

Fig. 1. First instantiation of Tomography

Distributional Different-Input Correctness. We prove below that (Tomography, Verify) satisfies $(\varepsilon, L, \Pi, \mathcal{D}_x)$-different-input correctness, where Π and $\mathcal{D}_x$ are defined as follows:

$$\Pi\left(\left(P_0, k_0, x_0, b_0\right), \left(P_1, k_1, x_1, b_1\right)\right) = \begin{cases} 0 & P_0 = P_1, x_0 = x_1 \text{ and } b_0 \neq b_1, \\ 1 & \text{otherwise.} \end{cases}$$

The sampler for $\mathcal{D}_x$ is defined as follows: sample $P_x \xleftarrow{\$} \mathcal{P}_n$, $k_0, k_1 \xleftarrow{\$} \{0,1\}^\lambda$ and output $\left(\left(P_x, k_0, x, 0\right), \left(\left(P_x, k_1, x, 1\right)\right)\right)$. We first prove an intermediate lemma that will be useful for proving distributional different-input correctness. Later on, this lemma will also be useful in the application of bit commitments.

Lemma 7. *Let $P_x \in \mathcal{P}_n$ and there exists a density matrix M such that* $\mathsf{Verify}(P_x \| k_0 \| x \| 0, M) = \mathsf{Valid}$ *and* $\mathsf{Verify}(P_x \| k_1 \| x \| 1, M) = \mathsf{Valid}$*, for some $k_0, k_1 \in \{0,1\}^\lambda$. Then*

$$\mathrm{Tr}\left(P_x \left|\psi_{k_1,x}\right\rangle\!\left\langle\psi_{k_1,x}\right| P_x \left|\psi_{k_0,x}\right\rangle\!\left\langle\psi_{k_0,x}\right|\right) \geq \frac{542}{729}.$$

The proof of this lemma can be found in the full version. With the above lemma in mind, we can prove the different-input correctness.

Lemma 8. (Tomography, Verify) *in Figure 1 satisfies $(O(2^{-n}), L, \Pi, \mathcal{D}_x)$-different-input correctness, where $L = O(2^{3n}\lambda)$.*

The proof of this lemma can be found in the full version.

We give a second instantiation in the full version that is used to achieve a psuedo-random one time pad.

6 Applications

In this section, we show how to use PRFS to constrtuct a variety of applications:

1. Bit commitments with classical communication and,
2. Pseudo one-time pad schemes with classical communication.

To accomplish the above applications, we use verifiable tomography from Section 5. The construction and proofs of the pseudo one-time pad schemes can be found in the full version of the paper.

6.1 Commitment scheme

We construct bit commitments with classical communication from pseudorandom function-like quantum states. We recall the definition by [2].

A (bit) commitment scheme is given by a pair of (uniform) QPT algorithms (C, R), where $C = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ is called the *committer* and $R = \{R_\lambda\}_{\lambda\in\mathbb{N}}$ is called the *receiver*. There are two phases in a commitment scheme: a commit phase and a reveal phase.

- In the (possibly interactive) commit phase between C_λ and R_λ, the committer C_λ commits to a bit, say b. We denote the execution of the commit phase to be $\sigma_{CR} \leftarrow \mathsf{Commit}\langle C_\lambda(b), R_\lambda\rangle$, where σ_{CR} is a joint state of C_λ and R_λ after the commit phase.
- In the reveal phase C_λ interacts with R_λ and the output is a trit $\mu \in \{0, 1, \perp\}$ indicating the receiver's output bit or a rejection flag. We denote an execution of the reveal phase where the committer and receiver start with the joint state σ_{CR} by $\mu \leftarrow \mathsf{Reveal}\langle C_\lambda, R_\lambda, \sigma_{CR}\rangle$.

We require that the above commitment scheme satisfies the correctness, computational hiding, and statistical binding properties below.

Definition 12 (Correctness). *We say that a commitment scheme (C, R) satisfies correctness if*

$$\mathsf{Pr}\left[b^* = b \ : \ \substack{\sigma_{CR}\leftarrow\mathsf{Commit}\langle C_\lambda(b),R_\lambda\rangle,\\ b^*\leftarrow\mathsf{Reveal}\langle C_\lambda,R_\lambda,\sigma_{CR}\rangle}\right] \geq 1 - \nu(\lambda),$$

where $\nu(\cdot)$ is a negligible function.

Definition 13 (Computational Hiding). *We say that a commitment scheme (C, R) satisfies computationally hiding if for any malicious QPT receiver $\{R^*_\lambda\}_{\lambda\in\mathbb{N}}$, for any QPT distinguisher $\{D_\lambda\}_{\lambda\in\mathbb{N}}$, the following holds:*

$$\Big| \Pr_{(\tau,\sigma_{CR^*})\leftarrow\mathsf{Commit}\langle C_\lambda(0),R^*_\lambda\rangle}[D_\lambda(\sigma_{R^*}) = 1] - \Pr_{(\tau,\sigma_{CR^*})\leftarrow\mathsf{Commit}\langle C_\lambda(1),R^*_\lambda\rangle}[D_\lambda(\sigma_{R^*}) = 1] \Big| \leq \varepsilon(\lambda),$$

for some negligible $\varepsilon(\cdot)$.

Definition 14 (Statistical Binding). *We say that a commitment scheme (C, R) satisfies statistical binding if for every QPT sender $\{C^*_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a (possibly inefficient) extractor $\mathcal{E}$ such that the following holds:*

$$\mathsf{Pr}\left[\mu \neq b^* \wedge \mu \neq \bot \ : \ \begin{array}{c} (\tau,\sigma_{C^*R})\leftarrow\mathsf{Commit}\langle C^*_\lambda, R_\lambda\rangle, \\ b^*\leftarrow\mathcal{E}(\tau), \\ \mu\leftarrow\mathsf{Reveal}\langle C^*_\lambda, R_\lambda, \sigma_{C^*R}\rangle \end{array}\right] \leq \nu(\lambda),$$

where $\nu(\cdot)$ is a negligible function and τ is the transcript of the Commit *phase.*

Remark 1 (Comparison with [2]). In the binding definition of [2], given the fact that the sender's and the receiver's state could potentially be entangled with each other, care had to be taken to ensure that after the extractor was applied on the receiver's state, the sender's state along with the decision bit remains (indistinguishable) to the real world. In the above definition, however, since the communication is entirely classical, any operations performed on the receiver's end has no consequence to the sender's state. As a result, our definition is much simpler than [2].

Construction Towards constructing a commitment scheme with classical communication, we use a verifiable tomography from Figure 1.

Construction. We present the construction in Figure 2. In the construction, we require $d(\lambda) = \lceil \log \frac{3\lambda}{n} \rceil \geq 1$.

$\mathsf{Commit}(b)$:

- The reciever R_λ samples an m-qubit Pauli $P = \bigotimes_{x\in\{0,1\}^d} P_x$ where $m = 2^d n$. It sends P to the commiter.
- The committer C_λ on intput $b \in \{0, 1\}$ does the following:
  - Sample $k \xleftarrow{\$} \{0, 1\}^\lambda$.
  - For all $x \in \{0, 1\}^d$
    - Generate $\sigma_x^{\otimes L} \leftarrow (\Phi_\lambda (P_x||k||x||b))^{\otimes L}$, where $L = 3^8 2^{3n+5}\lambda$.
    - $M_x \leftarrow \mathsf{Tomography}\left(\sigma_x^{\otimes L}\right)$.
  - Send $M = (M_x)_{x\in\{0,1\}^d}$ to the reciever.

Reveal:

- The commiter sends (k, b) as the decommitment. If $b \notin \{0, 1\}$, the reciever outputs $\bot$. Output b if for each $x \in \{0, 1\}^d$, $\mathsf{Verify}(P_x||k||x||b, M) = \mathsf{Valid}$, output $\bot$ otherwise.

Fig. 2. Commitment scheme

We prove that the construction in Figure 2 satisfies correctness, computational hiding and statistical binding properties.

Lemma 9 (Correctness). *The commitment scheme in Figure 2 satisfies correctness.*

Proof. This follows from Lemma 6. □

Lemma 10 (Computational Hiding). *The commitment scheme in Figure 2 satisfies computational hiding.*

Proof. We prove the security via a hybrid argument. Fix $\lambda \in \mathbb{N}$. Consider a QPT adversary R^*_λ.

Hybrid $H_{1,b}$, for all $b \in \{0,1\}$. This corresponds to C commiting to the bit b.

Hybrid $H_{2,b}$, for all $b \in \{0,1\}$. This hybrid is the same as before except that for all $x \in \{0,1\}^d$, $\Phi_\lambda(P||k||x||b)$ replaced with $\left(|0\rangle\langle 0| \otimes \left(P_x^b\right)\left(|\vartheta_x\rangle\langle\vartheta_x|\right)\left(P_x^b\right)\right)$ where $|\vartheta_1\rangle, ..., |\vartheta_{2^d}\rangle \leftarrow \mathscr{H}_n$.

The hybrids $H_{1,b}$ and $H_{2,b}$ are computationally indistinguishable because of the security of $PRFS$. $H_{2,0}$ and $H_{2,1}$ are identical by the unitary invariance property of Haar distribution. Hence, $H_{1,0}$ and $H_{1,1}$ are computationally indistinguishable.

□

Lemma 11 (Statistical Binding). *The commitment scheme in Figure 2 satisfies $O(2^{-0.5\lambda})$-statistical binding.*

Proof of Lemma 11. Let $C^* = \{C^*_\lambda\}_{\lambda\in\mathbb{N}}$ be a malicous committer. Execute the commit phase between C^*_λ and R_λ. Let τ be the classical transcript and let σ_{C^*R} be the joint state of C^*R. We first provide the description of an extractor.

Description of $\mathcal{E}$. On the input $\tau = (P, M)$, the extractor does the following:

1. For all $k'||b' \in \{0,1\}^\lambda \times \{0,1\}$, run for all $x \in \{0,1\}^d$, $\mathsf{Verify}(P_x||k'||x||b', M)$.
2. If for all $x \in \{0,1\}^d$, $\mathsf{Verify}(P||k'||x||b', M) = \mathsf{Valid}$, output b'.
3. Else output $\perp$.

Fact 6. *Let $\mathcal{P}_m$ be the m-qubit Pauli group. Then,*

$$\Pr_{P \xleftarrow{\$} \mathcal{P}_m}\left[\exists k_0, k_1 : \forall x \in \{0,1\}^d, |\langle\psi_{k_0,x}| P_x |\psi_{k_1,x}\rangle|^2 \geq \delta\right] \leq \delta^{-2^d} 2^{2\lambda - m}.$$

Proof. We use the following fact [2, Fact 6.9]: Let $|\psi\rangle$ and $|\phi\rangle$ be two arbitrary n-qubit states. Then,

$$\mathop{\mathbb{E}}_{P_x \xleftarrow{\$} \mathcal{P}_n}\left[|\langle\psi| P_x |\phi\rangle|^2\right] = 2^{-n}.$$

For any k_0, k_1, x by the above fact, $\mathbb{E}_{P_x \xleftarrow{\$} \mathcal{P}_n}\left[|\langle\psi_{k_0,x}| P_x |\psi_{k_1,x}\rangle|^2\right] = 2^{-n}$. Using Markov's inequality we get that for all $\delta > 0$,

$$\Pr_{P_x \xleftarrow{\$} \mathcal{P}_n}\left[|\langle\psi_{k_0,x}| P_x |\psi_{k_1,x}\rangle|^2 \geq \delta\right] \leq \delta^{-1} 2^{-n}.$$

Since, all P_x's are independent,

$$\Pr_{P \xleftarrow{\$} \mathcal{P}_m}\left[\forall x \in \{0,1\}^d, |\langle\psi_{k_0,x}| P_x |\psi_{k_1,x}\rangle|^2 \geq \delta\right] \leq \left(\delta^{-1}2^{-n}\right)^{2^d}.$$

Using a union bound over all k_0, k_1,

$$\Pr_{P \xleftarrow{\$} \mathcal{P}_m}\left[\exists k_0, k_1 : \forall x \in \{0,1\}^d, |\langle\psi_{k_0,x}| P_x |\psi_{k_1,x}\rangle|^2 \geq \delta\right] \leq \delta^{-2^d} 2^{2\lambda - m}.$$

□

Let the transcript be (P, M) where P is chosen uniformly at random. Let

$$p = \mathsf{Pr}\left[\mu \neq b^* \wedge \mu \neq \perp \; : \; \begin{array}{c}(\tau, \sigma_{C^*R}) \leftarrow \mathsf{Commit}\langle C^*_\lambda, R_\lambda\rangle, \\ b^* \leftarrow \mathcal{E}(\tau), \\ \mu \leftarrow \mathsf{Reveal}\langle \tau, \sigma_{C^*R}\rangle\end{array}\right]$$

Then

$$p = \Pr_{P \xleftarrow{\$} \mathcal{P}_m}\left[\exists k_0, k_1, b_0, b_1 \; : \forall x \in \{0,1\}^d \; \begin{array}{c}\mathsf{Verify}(P_x||k_0||x||b_0, M_x) = \mathsf{Valid}, \\ \mathsf{Verify}(P_x||k_1||x||b_1, M_x) = \mathsf{Valid}, \\ b_0 \neq b_1\end{array}\right].$$

Without loss of generality we can assume $b_0 = 0$ and $b_1 = 1$,

$$p = \Pr_{P \xleftarrow{\$} \mathcal{P}_m}\left[\exists k_0, k_1 \; : \forall x \in \{0,1\}^d \; \begin{array}{c}\mathsf{Verify}(P_x||k_0||x||0, M_x) = \mathsf{Valid}, \\ \mathsf{Verify}(P_x||k_1||x||1, M_x) = \mathsf{Valid}\end{array}\right].$$

By Lemma 7,

$$p \leq \Pr_{P \xleftarrow{\$} \mathcal{P}_m}\Big[\exists k_0, k_1 \; : \forall x \in \{0,1\}^d, \\ Tr(P_x |\psi_{k_1,x}\rangle\langle\psi_{k_1,x}| P_x |\psi_{k_0,x}\rangle\langle\psi_{k_0,x}|) \geq 542/729\Big]$$

By Fact 6,

$$p \leq \frac{729}{542}^{2^d}\left(2^{2\lambda - m}\right).$$

For $m \geq 3\lambda$, the protocol satisfies $O(2^{-0.5\lambda})$-statistical binding.

□

Acknowledgements

The authors would like to thank the anonymous TCC 2022 reviewers for their helpful comments. The authors would also like to thank Fermi Ma for his suggestions that improved the bounds and the analysis in the proof of binary phase PRS.

References

1. Andris Ambainis and Joseph Emerson. Quantum t-designs: t-wise independence in the quantum world. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 129–140. IEEE Computer Society, 2007.
2. Prabhanjan Ananth, Luowen Qian, and Henry Yuen. In *CRYPTO*, 2022.
3. 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.
4. Zvika Brakerski and Omri Shmueli. Scalable pseudorandom quantum states. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 417–440. Springer, 2020.
5. Fernando G. S. L. Brandão, Aram W. Harrow, and Michał Horodecki. Local random quantum circuits are approximate polynomial-designs. *Communications in Mathematical Physics*, 346(2):397–434, 9 2016.
6. Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Phys. Rev. A*, 80:012304, Jul 2009.
7. Dmitry Gavinsky. Quantum money with classical verification. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012*, pages 42–52. IEEE Computer Society, 2012.
8. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 8 1986.
9. 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.
10. Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 10 2020.
11. 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.
12. William Kretschmer. Quantum pseudorandomness and classical complexity. In Min-Hsiu Hsieh, editor, *16th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2021, July 5-8, 2021, Virtual Conference*, volume 197 of *LIPIcs*, pages 2:1–2:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
13. Hoi-Kwong Lo and H. F. Chau. Is quantum bit commitment really possible? *Phys. Rev. Lett.*, 78:3410–3413, Apr 1997.
14. Lowe, Angus. Learning quantum states without entangled measurements. Master's thesis, 2021.
15. Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Phys. Rev. Lett.*, 78:3414–3417, Apr 1997.

16. Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In *CRYPTO*, 2022.
17. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
18. Aidan Roy and A. J. Scott. Unitary designs and codes. *Designs, Codes and Cryptography*, 53(1):13–31, Oct 2009.
19. Mark Zhandry. How to construct quantum random functions. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 679–687. IEEE Computer Society, 2012.
20. Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 758–775. Springer, 2012.