Cryptography from Pseudorandom Quantum States*

Prabhanjan Ananth[1][0000−0001−5387−5730], Luowen Qian[2][0000−0002−1112−8822], and Henry Yuen[3][0000−0002−2684−1129]

[1] UCSB, Santa Barbara, USA prabhanjan@cs.ucsb.edu
[2] Boston University, Boston, USA luowenq@bu.edu
[3] Columbia University, New York City, USA hyuen@cs.columbia.edu

Abstract. Pseudorandom states, introduced by Ji, Liu and Song (Crypto'18), are efficiently-computable quantum states that are computationally indistinguishable from Haar-random states. One-way functions imply the existence of pseudorandom states, but Kretschmer (TQC'20) recently constructed an oracle relative to which there are no one-way functions but pseudorandom states still exist. Motivated by this, we study the intriguing possibility of basing interesting cryptographic tasks on pseudorandom states.

We construct, assuming the existence of pseudorandom state generators that map a λ-bit seed to a $\omega(\log \lambda)$-qubit state, (a) statistically binding and computationally hiding commitments and (b) pseudo one-time encryption schemes. A consequence of (a) is that pseudorandom states are sufficient to construct maliciously secure multiparty computation protocols in the dishonest majority setting.

Our constructions are derived via a new notion called *pseudorandom function-like states* (PRFS), a generalization of pseudorandom states that parallels the classical notion of pseudorandom functions. Beyond the above two applications, we believe our notion can effectively replace pseudorandom functions in many other cryptographic applications.

1 Introduction

Assumptions are the bedrock of designing provably secure cryptographic constructions. Over the years, theoretical cryptographers have pondered over the precise assumptions needed to achieve cryptographic tasks, often losing sleep over this [24]. The celebrated work of Goldreich [18] shows that most interesting cryptographic tasks (encryption, commitments, pseudorandom generators, etc.) imply the existence of one-way functions, i.e., functions that can be efficiently computed in the forward direction but cannot be efficiently inverted. Thus it appears that the existence of one-way functions is a *minimal* and *necessary* assumption in cryptography.

* HY is supported by AFOSR award FA9550-21-1-0040 and NSF CAREER award CCF-2144219. LQ is supported by DARPA under Agreement No. HR00112020023.

Quantum information processing presents new opportunities for cryptography. Specifically, in many contexts the assumptions necessary for cryptographic tasks can be weakened with the help of quantum resources. To illustrate, the seminal work of Bennett and Brassard [7] showed that key exchange can be achieved unconditionally (i.e. without any computational assumptions) using quantum communication. In contrast, key exchange is known to require computational assumptions if the parties are restricted to classical communication. More recently, the work of Bartusek, Coladangelo, Khurana, and Ma [4] and that of Grilo, Lin, Song and Vaikuntanathan [19] demonstrate that quantum protocols for secure multiparty computation can be constructed from post-quantum one-way functions. On the other hand classical protocols for secure computation cannot be based (in a black-box fashion) on one-way functions alone [22].

These examples suggest that we revisit our belief about the necessity of certain cryptographic assumptions for quantum cryptographic tasks (tasks that make use of quantum computation and/or quantum communication). Specifically, it is not even clear whether one-way functions are even necessary in the quantum setting — Goldreich's result [18] only applies to classical cryptographic primitives and protocols.

Our work continues the research agenda carried out by our predecessors [29,7,8,19,4]: *can we achieve cryptographic tasks using quantum communication in a world without one-way functions*[4]*?*

Pseudorandom Quantum States. Motivated by the question above, we turn to the notion of pseudorandom quantum states (abbreviated PRS) introduced by Ji, Liu and Song [23]. A *PRS generator* G is a quantum polynomial-time (QPT) algorithm that, given input a λ-bit key, outputs an n-qubit quantum state with the guarantee that it is computationally indistinguishable from an n-qubit Haar random state (i.e. the uniform distribution over n-qubit pure states), even with many copies. Ji, Liu and Song (and subsequently improved by Brakerski and Shmueli [11,12]) show the existence of PRS assuming post-quantum one-way functions.

This notion is analogous to pseudorandom generators (PRGs) from classical cryptography which take as input a random seed of length λ, and deterministically outputs a larger string of length $n > \lambda$ that is computationally indistinguishable from a string sampled from the uniform distribution. Despite the analogy, it has not been obvious whether pseudorandom quantum states have much cryptographic utility outside of quantum money [23] (unlike PRGs, which are ubiquitous in cryptography). Understanding the consequences of pseudorandom quantum states is particularly important in light of a recent result by Kretschmer [25], who showed that there is a relativized world where $BQP = QMA$ (and thus post-quantum one-way functions do not exist) while pseudorandom states exist. Kretschmer's result motivates us to focus the afore-

[4] Both the works [19,4] explicitly raised the question of basing secure computation on assumptions weaker than one-way functions.

mentioned research agenda on the following question: *what cryptographic tasks can be based solely on pseudorandom quantum states?*

1.1 Our Results

Our contributions in a nutshell are as follows:

- We propose a new notion called *pseudorandom function-like quantum states (PRFS).*
- Using PRFS, we show how to build (a) statistically binding commitments and (b) pseudo one-time encryption schemes. As a consequence of (a), we obtain maliciously secure computation in the dishonest majority setting.
- Finally, we show that for a certain range of parameters – the same as what is needed for the above applications – we can construct PRFS from a PRS.

Before we present the definition of PRFS, we first highlight the need for defining a new notion by describing the challenges for constructing primitives directly from PRS.

Challenges for Basing Primitives On PRS Although the closest classical analogue of a PRS generator is a PRG, the analogy breaks down in several critical ways. This makes it challenging to use PRS generators in the same way that PRGs are used throughout cryptography.

Specifically, PRS generators appear very *rigid*, meaning that it seems challenging to take an existing PRS generator and generically increase or decrease its output length. Moreover, it is difficult to use output qubits of a PRS generator independently.

Inability to Stretch the Output. A fundamental result about PRGs is that their *stretch* (the output length as a function of the key length) can be amplified arbitrarily. In other words, given a PRG G that maps λ random bits to at least $(\lambda+1)$ pseudorandom bits, one can construct a PRG with any polynomial output length. This fact is implicitly used everywhere in cryptography; specifically, it gives us the flexibility to choose the appropriate stretch of PRG relevant for the application without having to worry about the underlying hardness assumptions.

If PRS generators are analogous to PRGs, then one would expect that a similar amplification result to hold: the existence of PRS with nontrivial output length would (hopefully) imply the existence of PRS with arbitrarily large output length. The natural approach to amplify the stretch of a PRG by iteratively composing it with itself does not immediately work with PRS for a number of reasons; for one, a PRS generator takes as input a classical key while its output is a quantum state!

Inability to Shrink the Output. To add insult to injury, it is not even obvious how to *shrink* the output length of a PRS generator; this was also observed by Brakerski and Shmueli [9]. Classically, one can always discard bits from the

output of a PRG, and the result is still obviously a PRG. However, discarding a single qubit of an n-qubit pseudorandom state will leave a mixed state that is easily distinguishable from an $(n-1)$-qubit Haar-random state.

Inability to Separate the Output. Since the PRS output is highly entangled, it seems difficult to use the individual output qubits. As an example, suppose we want to encrypt a message of length ℓ. In the classical setting, an ℓ-bit output PRG can be used to encrypt a message of length ℓ by xor-ing the i^{th} PRG output bit with the i^{th} bit of the message. Implicitly, we are using the fact that the output of a PRG can be viewed a tensor product of bits and this feature of classical PRGs is mirrored by our notion of PRFS (explained next). On the other hand, if we have a single (entangled) PRS state (irrespective of the number of qubits it represents), it is unclear how to use each qubit to encode a bit; any operations performed on a single qubit could affect the other qubits that are entangled with this qubit.

New Notion: Pseudorandom Function-Like States Pseudorandom function-like states (abbreviated PRFS) is a generalization of PRS, where the same key k can be used to generate many pseudorandom states. In more details, a (d, n)-*PRFS generator* G is a QPT algorithm that, given as input a key $k \in \{0,1\}^\lambda$ and an input $x \in \{0,1\}^d$, outputs a n-qubit quantum state $|\psi_{k,x}\rangle$, satisfying the following pseudorandomness property: no efficient adversaries can distinguish between multiple copies of the output states $(|\psi_{k,x_1}\rangle, \ldots, |\psi_{k,x_s}\rangle)$ from a collection of states $(|\vartheta_1\rangle, \ldots, |\vartheta_s\rangle)$ where each $|\vartheta_i\rangle$ is sampled independently from the Haar distribution; furthermore, the indistinguishability holds even if the inputs $x_1, \ldots, x_s$ are chosen by the adversary. This is formalized in Definition 2.

An Alternate Perspective: Tensor Product PRS generators. If PRS generators are analogous to classical pseudorandom generators, then PRFS generators are analogous to classical pseudorandom *functions* (hence the name pseudorandom *function-like*). A PRS generator outputs a single state per key k. On the other hand, we can think of PRFS as a *relaxed* notion of PRS generator that on input k outputs a *tensor product* of states $|\psi_0\rangle \otimes |\psi_1\rangle \otimes \cdots \otimes |\psi_{2^d-1}\rangle$ where each $|\psi_i\rangle$, is indistinguishable from a Haar-random state.

The tensor product feature is quite useful in applications, as we will see shortly.

Additional Observations. Some additional observations of PRFS are in order:

- Assuming one-way functions, we can generically construct (d, n)-PRFS from any n-qubit PRS for any polynomial d, n. To compute PRFS on key k and input x, first compute a classical PRF on (k, x) and use the resulting output as a key for the n-qubit PRS. Since n-qubit PRS can be based on (post-quantum) one-way functions [23,12], this shows that even PRFS can be based on (post-quantum) one-way functions.

- In the other direction, we can construct n-qubit PRS from any (d, n)-qubit PRFS. On input k, the PRS simply outputs the result of PRFS on input $(k, 0)$.
- Another interesting aspect about PRFS is that it too, like PRS, is separated from (post-quantum) one-way functions. This can be obtained by a generalization of Kretschmer's result [2].

Implications We show that PRFS can effectively replace the usage of pseudorandom generators and pseudorandom functions in many primitives one learns about in "Cryptography 101". Specifically, we focus on two applications of PRFS generators. Later we will show that in fact that we can achieve these two applications from PRS generators only.

Implication 1. One-time Encryption with Short Keys and Long Messages. As a starter illustration of the usefulness of PRFS, we construct from a PRFS generator G a one-time encryption scheme for classical messages. The important feature of this construction is the fact that the message length is much larger than the key length. This is impossible to achieve information-theoretically, even in the quantum setting. This type of one-time encryption schemes, also referred to as *pseudo one-time pad*, is already quite useful, as it implies garbling schemes for P/poly [6] and even garbling for quantum circuits [13].

Theorem 1 (Informal; Pseudo One-time Pad). *Assuming the existence of (d, n)-PRFS with[5] $d = O(\log \lambda)$ and $n = \omega(\log \lambda)$, there exists a one-time encryption scheme for messages of length $\ell = 2^d$.*

We emphasize that in the implication to one-time encryption, we only require PRFS with logarithmic-length inputs.

The construction is simple and a direct adaptation of the construction of one-time encryption from pseudorandom generators. To encrypt a message x of length $\ell \gg \lambda$, output the state $G(k, (1, x_i)) \otimes \cdots \otimes G(k, (\ell, x_\ell))$, where $k \in \{0,1\}^\lambda$ is the symmetric key shared by the encryptor and the decryptor. The decryptor using the secret key k can decode[6] the message x. The security of the encryption scheme follows from the pseudorandomness of PRFS.

Implication 2. Statistically binding commitment schemes. We focus on designing commitment schemes with statistical binding and computational hiding properties. In the classical setting, this notion of commitment schemes can be constructed from any length-tripling PRG [27]. Recently, two independent works [19,4] showed that commitment schemes with aforementioned properties imply maliciously secure multiparty computation protocols with quantum communication in the dishonest majority setting. Of particular interest is the work of [4] who

[5] Recall that λ is the key length.

[6] In the technical sections, we define a QPT algorithm Test that given a state ρ along with k, x, determines if ρ is equal to the output $G(k, x)$. We show the existence of such a test algorithm for any PRFS.

construct the multiparty computation protocol using the commitment scheme as a *black box*. In particular, their construction works even when the commitment scheme uses quantum communication. They then instantiate the underlying commitment scheme from post-quantum one-way functions.

We design commitment schemes based on PRFS instead of one-way functions. First, we present a new definition of the statistical binding property for commitment schemes that utilize quantum communication. The notion of binding for quantum commitment schemes is more subtle than that for classical commitment schemes and has been extensively studied in prior works [30,28,17,4,9]. Our definition generalizes all previously known definitions of statistical binding for quantum commitments, and suffice for applications such as secure multiparty computation. (Our definition is formally presented in Definition 6).

Then we show, assuming the existence of PRFS with certain parameters, the existence of quantum commitment schemes satisfying our definition.

Theorem 2 (Informal). *Assuming the existence of (d, n)-PRFS[7] where $2^d \cdot n \geq 7\lambda$, there exists a statistically binding and computationally hiding commitment scheme.*

By plugging our commitment scheme into the framework of [4], we obtain the following corollary.

Corollary 1 (Informal). *Assumuing the existence of (d, n)-PRFS with $2^d \cdot n \geq 7\lambda$, there exists a maliciously secure multiparty computation protocol in the dishonest majority setting.*

Our construction is an adaptation of Naor's commitment scheme [27]. We replace the use of the PRG in Naor's construction with a PRFS generator and the first message, which is a random string in Naor's construction, instead specifies a random Pauli operator.

Other Implications. Besides the above applications, we show that PRFS (with polynomially-long input length) can also be used to construct other fundamental primitives such as symmetric-key CPA-secure encryption and message authentication codes (see full version). Both primitives guarantee security in the setting when the secret key can be reused multiple times.

Unlike the previous applications (pseudo QOTP and commitments), the straightforward constructions of reusable encryption and MACs require PRFS generators with input lengths $\omega(\log \lambda)$ and ℓ respectively, where ℓ is the length of the message being authenticated. We do not know if such PRFS generators can be constructed from PRS generators in a black box way. Nonetheless, we believe these applications illustrate the usefulness of the concept of PRFS generators.

[7] To simplify the analysis, there is an additional technical property of the PRFS not mentioned here that is required by our construction, called *recognizable abort* (Definition 4). All known constructions of PRFS and PRS (including ours) have the recognizable abort property.

Construction of PRFS Given the interesting implications of PRFS, the next natural step is to focus on constructing PRFS generators. We show that for some interesting range of parameters, we can achieve PRFS from any PRS. In particular, we show the following.

Theorem 3 (Informal). *For $d = O(\log \lambda)$ and $n = d + \omega(\log\log\lambda)$, assuming the existence of a $(d+n)$-qubit PRS generator, there exists a (d, n)-PRFS generator.*

A surprising aspect about the above result is that the starting PRS's output length $d + n = \omega(\log\log\lambda)$ could even be much smaller than the key length λ. In contrast, classical pseudorandom generators with output length less than the input length are trivial.

We remark that if $d \ll \log\lambda$ then it is easy to build PRFS from PRS; chop up the key k into 2^d blocks; to compute the PRFS generator with key k and input x, compute the PRS generator on the x^{th} block of the key. Unfortunately, PRFS with this range of parameters does not appear useful for applications because the seed length is too large. On the other hand, the construction of PRFS generators from PRS generators in Theorem 3 allows for 2^d to be an arbitrarily large polynomial in the key length. Note that this is sufficient for Theorem 1 and Corollary 1. We thus obtain the following corollary.

Corollary 2. *Assuming $(2\log\lambda + \omega(\log\log\lambda))$-qubit PRS, there exist statistically binding commitment schemes and therefore secure computations. Assuming $\omega(\log\lambda)$-qubit PRS, there exist pseudo one-time pad schemes for messages of any polynomial length.*

We remark that the assumptions of Corollary 2 on the PRS generators are essentially *optimal*, in the sense that it is not possible to significantly weaken them. This is because commitment and pseudo one-time pad schemes require computational assumptions on the adversary; on other hand Brakerski and Shmueli [12] demonstrate the existence a "pseudo"-random state generator with output length $c\log\lambda$ for some constant $c < 1$ that is *statistically secure*: in other words, the outputs of the generator are indistinguishable from Haar-random states by *any* distinguisher (not just polynomial-time ones).

Furthermore, it can be shown that PRS generators with $\log\lambda$-qubit outputs require computational assumptions on the adversary and that generators with $(1+\varepsilon)\log\lambda$-qubit outputs imply BQP $\neq$ PP [2].

Concurrent Work A concurrent preprint of Morimae and Yamakawa [26] also construct statistically binding and computationally hiding commitment schemes from PRS, adapting Naor's commitment scheme in a manner similar to ours. We note several differences between their work and ours, with regards to commitment schemes.

1. They show a weaker notion of binding known as *sum-binding*, which roughly says that the *sum* of the probabilities that an adversarial committer can

successfully decommit to the bit 0 and the bit 1 is at most a quantity negligibly close to 1. This notion of binding is not known to be sufficient for general quantum commitment protocols to conclude that PRS implies protocols for secure computation[8]. However, our notion of statistical binding (Definition 6) is sufficient for leveraging the machinery of [4] to obtain quantum protocols for secure computation. Moreover, our definition of statistical binding implies the sum-binding definition[9].

2. For the same level of statistical binding security, that is $O(2^{-\lambda})$, they require the existence of a PRS that stretches λ random bits to 3λ qubits of Haar-randomness (i.e., they require the PRS generator to have *stretch*), whereas our result assumes the existence of a PRS that maps λ bits to $2\log\lambda + \omega(\log\log\lambda)$ qubits. On the other hand, they require the pseudorandomness/indistinguishability of a single copy of PRS state versus Haar random, while we require the pseudorandomness to hold again multiple copies, especially when the output length is short.
3. The state generation guarantee required from the underlying PRS is much stricter in their setting. In our work, we require the underlying PRS to only satisfy recognizable abort (Definition 4) whereas in their work, the underlying PRS needs to satisfy a guarantee that is even stronger than perfect state generation (Definition 3).
4. Their commitment scheme is non-interactive whereas our commitment scheme is a two-message scheme. Furthermore, our protocol has a classical opening message while theirs is quantum. However, these differences are rather minor since we can easily adapt our construction to satisfy these requirements, and vice versa.

We also note that the notion of PRFS, its implications and its construction from PRS is unique to our work.

1.2 Discussion: Why Explore a World Without One-Way Functions?

Before getting into the technical overview we address a common question: *"Sure, it is interesting that one can construct commitment schemes and pseudo one-time pad schemes without one-way functions, but will this still matter if someone proves that (post-quantum) one-way functions exist?"*

Our view is the following: it is *not* our goal to avoid one-way functions because we don't believe that they exist[10]. The main motivation is to gain a *deeper understanding* of fundamental cryptographic primitives such as encryption and commitment schemes. As mentioned previously, it has been understood for many

[8] However, in an updated draft of [26], the authors sketch how, for a special form of quantum commitment schemes, sum-binding does imply our notion of statistical binding.

[9] The sum of probabilities that an adversarial decommitter can decommit to 0 and to 1 in the ideal world of our definition (Definition 6) and therefore they sum up to at most negligibly larger than 1 in the real world by our statistical binding guarantee.

[10] The majority of the authors of this paper believe one-way functions exist.

decades that these primitives are inseparable from one-way functions (even in a black box way) in the classical setting. We view our results as revising this understanding in the quantum world: one-way functions are not necessary for these primitives.

Another motivation comes from complexity theory. An oft-repeated storyline is that if P = NP, then one-way functions would not exist and thus most cryptography would be impossible; this scenario has been coined by Impagliazzo as *Algorithmica* as one of his five "complexity worlds" [21]. While most people believe that P $\neq$ NP, it is nonetheless scientifically interesting to study the consequences of other complexity-theoretic outcomes. Our work adds a twist to the usual P = NP storyline: perhaps *QAlgorithmica* – Impagliazzo's *Algorithmica* plus quantum information – can potentially support both an algorithmic *and* cryptographic paradise.

Finally, we believe that studying the possibilities of basing cryptography solely on quantum assumptions is extremely useful for deepening our understanding of quantum information. By restricting ourselves to *not* use one-way functions, we force ourselves to use the unique properties of quantum mechanics to the hilt. For example, our constructions of PRFS generators, pseudo one-time pad and commitment schemes ultimately required us to make use of properties of pseudorandom states such as concentration of measure over the Haar distribution.

Another question that often arises is: *"Is there a candidate construction of PRS generators that do not (obviously) involve one-way functions?"* While Kretschmer [25] showed an oracle separation between pseudorandom states and one-way functions, this is an artificial setting where the oracle is constructed by sampling a Haar-random unitary.

We claim that *random quantum circuits* form natural constructions of pseudorandom states: the generator G interprets the key k as a description of a quantum circuit on n qubits, and G outputs the state $k|0^n\rangle$ (i.e. executes the circuit with the all zeroes input). It has been conjectured in a number of settings that random quantum circuits have excellent pseudorandom properties. For example, the quantum supremacy experiments of Google [3] and UTSC [31] are based on the premise that random n-qubit circuits of sufficiently large depth should generate states that are essentially Haar-random [20]. Random quantum circuits have also been extensively studied as toy models of scrambling in black hole dynamics [15,10,14].

It seems beyond the reach of present-day techniques to prove that polynomial-size random quantum circuits yield pseudorandom states; for one, doing so would separate BQP from PP [25], which would be an incredible result in complexity theory. However, this is a plausible candidate PRS generator, and arguably this construction does not involve one-way functions at all.

1.3 Technical Overview

We first describe the techniques behind the construction of pseudorandom function-like states from pseudorandom quantum states. Then, we will give an overview of the result of statistical binding commitments from PRFS.

PRFS from PRS To construct a (d, n)-PRFS, we start with an $(n + d)$-qubit PRS. For the purposes of the current discussion, we will assume that PRS has *perfect state generation*. That is, the output of PRS is a pure state.

Main Insight: Post-Selection. The construction proceeds as follows: on input key k and $x \in \{0,1\}^d$, first generate a $(d + n)$-qubit PRS state by treating k as the PRS seed. As the PRS satisfies perfect state generation, the output is a pure state and we can write the state as $|\psi\rangle = \sum_{x\in\{0,1\}^d} \alpha_x |x\rangle \otimes |\psi_x\rangle$, where $|\psi_x\rangle$ is a n-qubit state. Suppose we post-select (i.e., condition) on the first d qubits being in the state $|x\rangle$, the remaining n qubits will be in the state $|\psi_x\rangle$, which we define to be the output of the PRFS on input (k, x).

In general, we do not know how to perform post-selection in polynomial-time [1]. However, if the event on which we are post-selecting has an inverse polynomial (where the polynomial is known ahead of time) probability of occurring, then we can efficiently perform post-selection. That is, we repeat the following process $2^d\lambda$ number of times: generate $|\psi\rangle$ by computing the PRS generator on k and then measure the first d qubits in the computational basis. If the first d qubits is x then output the residual state (which is $|\psi_x\rangle$), otherwise continue. If in none of the $2^d\lambda$ iterations, we obtained the first d qubits to be x, we declare failure and output some fixed state $|\perp\rangle$.

We prove that the above PRFS generator satisfies pseudorandomness by making two observations.

Observation 1: Output of PRFS is close to $|\psi_x\rangle$. We need to argue that the probability that the PRFS generator outputs $|\psi_x\rangle$ is negligibly (in λ) close to 1. This boils down to showing that with probability negligibly close to 1, in one of the iterations, the measurement outcome will be x. Indeed if $|\alpha_x|^2$ is roughly $\frac{1}{2^d}$ then this statement is true. But it is a priori not clear how to argue this.

Towards resolving this issue, let us first pretend that $|\psi\rangle$ was instead drawn from the Haar measure. In this case, we can rely upon Lévy's Lemma to argue that $|\alpha_x|^2$ is indeed close to $\frac{1}{2^d}$, with overwhelming probability over the Haar measure. Thus, if $|\psi\rangle$ was drawn from the Haar measure, the probability that the PRFS generator outputs $|\psi_x\rangle$ is negligibly close to 1.

Now, let us go back to the case when $|\psi\rangle$ was a PRS state. Since the PRFS generator is a quantum polynomial-time algorithm, it cannot distinguish whether $|\psi\rangle$ was generated by PRS or whether it was sampled from the Haar measure. This means that the probability that it outputs $|\psi_x\rangle$, when $|\psi\rangle$ was a PRS state, should also be negligibly close to 1.

While ideally we would have liked the PRFS to have perfect state generation, the above construction still satisfies a nice property that we call *recognizable*

abort: the output of the PRFS is either a pure state or it is some known pure state $|\perp\rangle$.

All is left is to show that the post-selected state $|\psi_x\rangle$ is Haar random when $|\psi\rangle$ is Haar random.

Observation 2: Post-selected Haar random state is also Haar random. Haar random states satisfy a property called unitary invariance: applying any unitary on a Haar random state yields a Haar random state. Consider the following distribution $\mathcal{R}$ of unitaries: $R = \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes R_x$, where R_x is a Haar random unitary. Now, applying R, where $R \leftarrow \mathcal{R}$, on a Haar random state $|\psi\rangle = \sum_{x\in\{0,1\}^d} |x\rangle \otimes |\psi_x\rangle$ yields a Haar random state.

Thus, the following two processes yield the same distribution:

- Process 1: Sample $|\psi\rangle = \sum_{x\in\{0,1\}^d} |x\rangle\otimes|\psi_x\rangle$ be a Haar random state. Output $|\psi_x\rangle$.
- Process 2: Sample a Haar random state $|\psi\rangle = \sum_{x\in\{0,1\}^d} |x\rangle \otimes |\psi_x\rangle$. Output $R_x |\psi_x\rangle$.

Notice that the output distribution of Process 2 is Haar random since R_x is a Haar random unitary. From this we can conclude that even the output distribution of Process 1 is also Haar random.

Test Procedure. Classical pseudorandom generators satisfy a verifiability property that we often take for granted: given a value y and a seed k, we can successfully check if y is obtained as an evaluation of a seed k. This feature is implicitly used in many applications of pseudorandom generators. We would like to have a similar feature even for pseudorandom function-like states. More specifically, we would like the following to hold: given a state ρ, a PRFS key k and an input x, check if ρ is close to the output of PRFS on (k, x).

Let us start with a simple case when the PRFS satisfies perfect state generation property and moreover, PRFS generator is a unitary G. We can express PRFS state generation as follows: on input a key k, input x and ancillas $|k\rangle\otimes|x\rangle\otimes|0\rangle$, G outputs $|\psi_{k,x}\rangle\otimes|\phi\rangle$. The state $|\psi_x\rangle$ is deisgnated to be the PRFS state corresponding to input x and the state $|\phi\rangle$ is discarded as the garbage state.

Suppose we need to test if a state ρ is the output of PRFS on k and x. The test procedure is defined as follows:

- Compute $G(|k\rangle \otimes |x\rangle \otimes |0\rangle)$,
- Swap the register containing the PRFS state with ρ,
- Apply $G^\dagger$ on the resulting state and,
- Measure the resulting state and output 1 if the outcome is $(k, x, 0)$. Otherwise, output 0.

Since unitaries preserve fidelity between the states, we can show that the following holds: the above test procedure outputs 1 with probability proportional to $F(\rho, |\psi_{k,x}\rangle\langle\psi_{k,x}|)$. More precisely, the test procedure outputs 1 with probability $\mathrm{Tr}(|\psi_{k,x}\rangle\langle\psi_{k,x}|\,\rho)$.

The above test procedure can be suitably generalized if the PRFS satisfies the (weaker) state generation with recognizable abort property. If the PRFS generator is a quantum circuit then we designate G, in the above test procedure, to be a purification of this quantum circuit.

Statistical Binding Commitments We show how to construct statistical binding quantum commitments from PRFS.

Definition. A statistical binding quantum commitment scheme consists of two interactive phases between a sender and a receiver: a commit phase and a reveal phase. In both the phases, the communication between the parties can be quantum. In the commit phase, the sender commits to a bit b. In the reveal phase, the committer reveals b and the receiver either accepts or rejects.

We require that any (even unbounded) sender cannot commit to bit b in the commit phase and then successfully open to $1-b$ in the reveal phase. Formalizing this can be tricky in the setting where the communication channel is quantum. For example, consider the following attack: an adversarial sender can send a uniform superposition of commitments of 0 and 1 and then open to one of them in the reveal phase. Any definition we come up should handle this attack.

We propose an extractor-based definition. Consider an adversarial sender S^*. Let us define the ideal experiment as follows: execute the commit phase with S^*. After the commit phase, apply an extractor on the receiver's state. The output of the extractor is a bit b^* along with the collapse state σ_R. Execute the reveal phase; let b be the bit opened to by S^*. Output Fail if $b \neq b^*$ and R accepts. Otherwise, output S^*'s final state (after the execution of the Reveal phase) along with R's decision, which is either the decommitted bit of the sender or it is $\perp$. Similarly, we can define real experiment as follows: We execute the commit phase and the reveal phase between S^* and R and then output the final state of S^* along with R's decision.

Going back to the earlier superposition attack, the extractor would, with equal probability, collapse to either commitment of 0 or collapse to commitment of 1.

We say that the quantum commitment scheme satisfies statistical binding if the output distributions of the real and ideal experiments are statistically close. Our definition of statistical binding generalizes all the previous definitions of statistical binding in the context of quantum commitments [30,28,17,4,9]. Refer to Section 5.1 for a detailed comparison with prior definitions.

We also require the quantum commitment scheme to satisfy computational hiding: in the commit phase, any quantum poly-time receiver cannot tell apart whether the sender committed to 0 or 1.

Construction. Our construction is a direct adaptation of Naor's commitment scheme [27], i.e. the same protocol but simply replacing PRG with PRFS. We start with a (d, n)-PRFS, where $d = O(\log(\lambda))$ and $n \geq 1$.

- In the commit phase, the receiver sends a random $2^d n$-qubit Pauli $P = P_0 \otimes \cdots P_{2^d-1}$ to the sender, where each P_i is a 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 $\mathbf{c} = \bigotimes_{x\in[2^d]} P_i^b \sigma_{k,x} P_i^b$, where $\sigma_{k,x} = PRFS(k,i)$ to the receiver.
- In the reveal phase, the sender sends (k,b) to the receiver. The receiver accepts if $P^b \mathbf{c} P^b$ is a tensor product of the PRFS evaluations of (k,x), for all $x = 0,\ldots,2^d-1$.

From the pseudorandomness property of PRFS, hiding follows. To prove that the above scheme satisfies binding, we describe the extractor first. It again helps to think of PRFS as satisfying the perfect state generation property. The extractor applies the following projection $\{\Pi_0, \Pi_1, I-(\Pi_0+\Pi_1)\}$, where Π_b projects onto the subspace spanned by $T_b = \left\{\bigotimes_{x\in\{0,1\}^{2^d}} P^b \left|\psi_{k,x}\right\rangle\!\left\langle\psi_{k,x}\right| P^b \;:\; \forall k \in \{0,1\}^\lambda\right\}$, where $|\psi_{k,x}\rangle$ is the output of $PRFS(k,x)$. If Π_b succeeds then b is designated to be the extracted bit. At the core of proving the indistinguishability of the real and the ideal world is the following fact: applying a projector that projects onto T_b (as done by the extractor), followed by the projector $\bigotimes_{x\in\{0,1\}^{2^d}} P^b \left|\psi_{k,x}\right\rangle\!\left\langle\psi_{k,x}\right| P^b$ (as done by the receiver) is the equivalent to only applying the projector $\bigotimes_{x\in\{0,1\}^{2^d}} P^b \left|\psi_{k,x}\right\rangle\!\left\langle\psi_{k,x}\right| P^b$.

While our actual proof is conceptually similar to the proof sketched above, there are some crucial details that we shoved under the rug. Firstly, $I-(\Pi_0+\Pi_1)$ is not necessarily a projection since the projections Π_0 and Π_1 need not be orthogonal. Secondly, the PRFS generation is not perfect and we need to work with recognizable abort property. Nonetheless we circumvent these issues and show that the above construction still works. We refer the reader to Section 5.2 for more details.

1.4 Future Directions

We end this section with some future directions and open questions.

Properties of pseudorandom states. Given a PRS generator G mapping λ-bit keys to n-qubit states, is it possible to construct in as black-box fashion as possible, a PRS generator G' with longer output length (but same length key)? In other words, it is possible to arbitrarily *stretch* the output of a PRS?

Is it possible to construct PRFS generators (with polynomial-length inputs) from PRS generators in a black-box fashion? Are there separations?

More applications of pseudorandom states. One of Impagliazzo's "five worlds" is called *MiniCrypt*, which represents a world where one-way functions exist but we do not have public-key cryptography. In this world, applications such as symmetric-key encryption, commitment schemes, secure multiparty computation, and digital signatures are possible to achieve.

It appears that we can obtain most MiniCrypt primitives from PR(F)S; for example this paper shows that we can get symmetric-key encryption, commitments, and secure multiparty computation. However it is a tantalizing open question of whether we can also build digital signatures from PR(F)S. Morimae and Yamakawa show that an analogue of one-time Lamport signatures can be constructed from PRS [26], but obtaining many-time signatures from PR(F)S seems more challenging.

More generally, what are other cryptographic applications of pseudorandom states?

Other quantum assumptions. What are other interesting "fully quantum" assumptions that can we base cryptography on? Can we base cryptography on the assumption $\mathsf{BQP} \neq \mathsf{PP}$? We note that Chia, Chou, Zhang, Zhang also suggest the possibility of basing cryptography on the assumption that a quantum version of the Minimum Circuit Size Problem is hard [16, Open Problem 9].

Acknowledgements

We thank Tomoyuki Morimae, Takashi Yamakawa, Jun Yan, and Fermi Ma for their very helpful feedback and discussions about pseudorandom quantum states.

2 Pseudorandom States

The notion of pseudorandom states were first introduced by Ji, Liu, and Song in [23]. We reproduce their definition here:

Definition 1 (PRS Generator [23]). *We say that a QPT algorithm G is a* pseudorandom state (PRS) generator *if the following holds.*

1. ***State Generation**. There is a negligible function $\varepsilon(\cdot)$ such that 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 $\varepsilon(\lambda)$ such that for all λ, we have*
$$\left| \Pr_{k\leftarrow\{0,1\}^\lambda}\left[A_\lambda(G_\lambda(k)^{\otimes t(\lambda)}) = 1\right] - \Pr_{|\vartheta\rangle\leftarrow\mathscr{H}_{n(\lambda)}}\left[A_\lambda(|\vartheta\rangle^{\otimes t(\lambda)}) = 1\right]\right| \leq \varepsilon(\lambda) .$$

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 4 ([23,12]). *If post-quantum one-way functions exist, then there exist PRS generators for all polynomial output lengths.*

2.1 Pseudorandom Function-Like State (PRFS) Generators

In this section, we present our definition of pseudorandom function-like state (PRFS) generators. PRFS generators generalize PRS generators in two ways: first, in addition to the secret key k, the PRFS generator additionally takes in a (classical) input x. The security guarantee of a PRFS implies that, even if x is adversarily chosen, the output state of the generator is indistinguishable from Haar-random. 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.

Definition 2 (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 $(\{x_1, \ldots, x_{s(\lambda)}\} \subseteq \{0,1\}^{d(\lambda)})_\lambda$, there exists a negligible function $\varepsilon(\cdot)$ such that for all λ,*

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

We also 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. One could consider stronger notions of security where the indistinguishability property holds even when the adversary is allowed to query the PRFS generator adaptively, or even in superposition. We explore these stronger notions in forthcoming work [2].

State Generation Guarantees. As mentioned above, our definition of PRFS generator does not require that the output of the generator is always a pure state. However, we will see later that a consequence of the PRFS security guarantee is that the output of the generator is *close* to a pure state for an overwhelming fraction of keys k.

Nevertheless, for applications it is sometimes more useful to also consider a stronger guarantee on the state generation of a PRFS generator.

Definition 3 (Perfect state generation). *A $(d(\lambda), n(\lambda))$-PRFS generator G satisfies perfect state generation, if for every $k \in \{0,1\}^\lambda$ and $x \in \{0,1\}^{d(\lambda)}$, there exists an $n(\lambda)$-qubit pure state $|\psi\rangle$ such that $G_\lambda(k, x) = |\psi\rangle\langle\psi|$.*

We observe that an $n(\lambda)$-PRS generator defined in Definition 1 is by definition equivalent to an $(0, n(\lambda))$-PRFS generator with perfect state generation.

In general, it may be difficult to construct PR(F)S with perfect state generation as the state generation could occasionally fail; for example, the generator may perform a (quantum) rejection sampling procedure in order to output the

state. The scalable PRS generators of Brakerski and Shmueli [12] is an example of this. To capture a very natural class of PRFS generators (including the one constructed in this paper), we define the notion of a PRFS generator where $G(k,x)$ outputs a convex combination of a fixed pure state $|\psi_{k,x}\rangle$ or a known abort state $|\perp\rangle$.

Definition 4 (Recognizable abort). *A $(d(\lambda), n(\lambda))$-PRFS generator G has the* recognizable abort property *if for every $k \in \{0,1\}^\lambda$ and $x \in \{0,1\}^{d(\lambda)}$ there exists an $n(\lambda)$-qubit pure state $|\psi\rangle$ and $0 \leq \eta \leq 1$ such that $G_\lambda(k,x) = \eta\,|\psi\rangle\langle\psi| + (1-\eta)\,|\perp\rangle\langle\perp|$, where $\perp$ is a special symbol*[11].

Note that this definition alone does not have any constraint on η being close to 1. However, the security guarantee of a PRFS generator implies that η will be negligibly close to 1 with overwhelming probability over the choice of k.[12] We also note that a PRFS generator with perfect state generation trivially has the recognizable abort property with $\eta = 1$ for all k, x.

2.2 Testing Pseudorandom States

Given a state ρ, it is useful to know whether it is the output of a PRFS generator with key k and input x. One approach would be to invoke the generator to get some number of copies and perform SWAP tests. Unfortunately, this approach would only achieve polynomially small error, which is undesirable for cryptographic applications where we want negligible security. Another approach is to "uncompute" the state generation. The issue with this approach is that it is not clear how to do it when the state generation is not perfect, or if it outputs some additional auxiliary states that we do not know how to uncompute.

In the following, we will show how to use the generator in a semi-black-box way to test any PRFS states. We first state a general Lemma that shows how to convert any circuit that generates a state ρ into a tester (of sorts) for the state ρ.

Lemma 1 (Circuit output tester). *Let G denote a (generalized) quantum circuit that takes no input and outputs an n-qubit mixed state ρ. Then there exists a circuit* Test *with boolean output such that:*

1. *For all density matrices σ_{EQ} where* Q *is an n-qubit register, applying the circuit* Test *on register* Q *yields the following state on registers* EF *where* F

[11] One can think of $|\perp\rangle$ as the $(n+1)$-qubit state $|100\cdots0\rangle$ with the first qubit indicating whether the generator aborted or not. If the generator doesn't abort, then it outputs $|0\rangle \otimes |\psi\rangle$ for some pure state $|\psi\rangle$ (called the *correct output state* of G on input (k,x)). The distinguisher in the definition of PRFS generator would then only get the last n qubits as input.

[12] The argument is as follows: if η were on average noticeably far from 1, then a purity test using SWAP tests would distinguish the outputs from Haar random states which are pure. This is formalized in the full version.

stores the decision bit:

$$(I_{\mathsf{E}} \otimes \mathsf{Test}_{\mathsf{Q}})(\sigma_{\mathsf{EQ}}) = \sum_b \mathrm{Tr}_{\mathsf{Q}}\Big((I_{\mathsf{E}} \otimes M_b)\sigma_{\mathsf{EQ}}\Big) \otimes |b\rangle\langle b|_{\mathsf{F}}$$

where $M_1 = \rho^2$ *and* $M_0 = I - M_1$.

2. *Furthermore,* Test *runs the unitary part*[13] *of* G *as a black box, and if the complexity of* G *is* T*, the complexity of* Test *is* $O(T + n)$.

Due to space constraints, we defer the proof of this lemma to the full version.

We note that if a PRFS satisfies perfect state generation, then the Test algorithm corresponding to the circuit $G_\lambda(k, x)$ implements a projection onto the state $|\psi_{k,x}\rangle = G_\lambda(k, x)$ in the case that the Test accepts (i.e. outputs 1). If the PRFS satisfies the weaker recognizable abort property, we get that the Test algorithm implements a *scaled* projection onto the correct state $|\psi_{k,x}\rangle$.

Corollary 3 (PRFS tester with recognizable abort). *Let* G *be a* (d, n)*-PRFS generator with the recognizable abort property. Then there exists a QPT algorithm* Test *such that for all* λ, $k \in \{0,1\}^\lambda$ *and* $x \in \{0,1\}^{d(\lambda)}$*, for all density matrices* σ_{EQ} *where* Q *is an* $n(\lambda)$*-qubit register, applying* $\mathsf{Test}(k, x, \cdot)$ *to register* Q *yields the following state on registers* EF *where* F *stores the decision bit:*

$$(I_{\mathsf{E}} \otimes \mathsf{Test}_{\mathsf{Q}})(k, x, \sigma_{\mathsf{EQ}}) = \sum_b \mathrm{Tr}_{\mathsf{Q}}\Big((I_{\mathsf{E}} \otimes M_b)\sigma_{\mathsf{EQ}}\Big) \otimes |b\rangle\langle b|_{\mathsf{F}}$$

where $M_1 = \eta^2 |\psi\rangle\langle\psi|$ *and* $M_0 = I - M_1$ *with* $\eta, |\psi\rangle$ *(which generally depend on* k, x*) are those guaranteed by the recognizable abort property.*

Proof. Fix λ and $k \in \{0,1\}^\lambda, x \in \{0,1\}^{d(\lambda)}$. By the recognizable abort property, we know that $G_\lambda(k, x) = \eta |\psi\rangle\langle\psi| + (1 - \eta) |\perp\rangle\langle\perp|$. We implement the circuit Test by first testing whether the input state is $|\perp\rangle$ (which we can do since it is a fixed known state), rejecting if so, and otherwise applying the test circuit from Lemma 1 with the circuit $G_{k,x}$ that takes no input and outputs $\rho = G_\lambda(k, x)$. Since we projected the input state to have no overlap with $|\perp\rangle$, we get that

$$\rho\, \sigma\, \rho = \eta^2\ |\psi\rangle\langle\psi|\ \sigma\ |\psi\rangle\langle\psi|$$

as desired. □

Next we analyze a *product* of Test algorithms run in parallel on different qubits of a (possibly entangled) state.

Corollary 4 (Product of PRFS testers with recognizable abort). *Let* G *be a* (d, n)*-PRFS generator with the recognizable abort property and let* Test *denote the corresponding tester algorithm given by Corollary 3. Fix* $\lambda, t \in \mathbb{N}$*. For all* $k_1, \ldots, k_t \in \{0,1\}^\lambda$ *and for all* $x_1, \ldots, x_t \in \{0,1\}^{d(\lambda)}$*, define the QPT*

[13] See the full version for a definition of the unitary part of a generalized quantum circuit.

algorithm $\mathsf{Test}^{\otimes t}$ *that given an* $t \cdot n(\lambda)$*-qubit density matrix* σ *behaves as follows: for all* $i = 1, \ldots, t$*, on the* i*'th block of* $n(\lambda)$ *qubits of* σ*, run the algorithm* $\mathsf{Test}_\lambda(k_i, x_i, \cdot)$*. Output* 1 *if and only if all* t *invocations of* Test *output* 1*.*

Then $\mathsf{Test}^{\otimes t}$ *satisfies the following. For all density matrices* σ_{EQ} *where* Q *is an* $t \cdot n(\lambda)$*-qubit register, applying* $\mathsf{Test}^{\otimes t}$ *to register* Q *yields the following state on registers* EQF *where* F *stores the decision bit:*

$$(I_{\mathsf{E}} \otimes \mathsf{Test}^{\otimes t})(\sigma_{\mathsf{EQ}}) = \sum_b \mathrm{Tr}_{\mathsf{Q}}\Big((I_{\mathsf{E}} \otimes M_b)\sigma_{\mathsf{EQ}}\Big) \otimes |b\rangle\langle b|_{\mathsf{F}}$$

where $M_1 = \eta^2 |\psi\rangle\langle\psi|$ *and* $M_0 = I - M_1$ *with* $|\psi\rangle = |\psi_{k_1,x_1}\rangle \otimes \cdots \otimes |\psi_{k_t,x_t}\rangle$*, and* $\eta = \eta_{k_1,x_1} \cdots \eta_{k_t,x_t}$ *where* $|\psi_{k_i,x_i}\rangle, \eta_{k_i,x_i}$ *for* $i = 1, ..., t$ *are the values guaranteed by the recognizable abort property.*

Proof. This follows from the fact that each invocation of $\mathsf{Test}(k_i, x_i, \cdot)$, conditioned on accepting, implements a (scaled) projection $\eta_{k_i,x_i} |\psi_{k_i,x_i}\rangle\langle\psi_{k_i,x_i}|$ on a disjoint register of σ. □

We note that the previous two Corollaries establish the behavior of the Test procedure for *every* fixed key k (or sequence of keys, in the case of Corollary 4). The next Lemma establishes the behavior of the Test procedure when given outputs of *any* PRFS generator (even ones without recognizable abort); the bounds are stated *on average* over a uniformly random key k.

Lemma 2 (Self-testing PRFS). *Let* G *be a* (d, n)*-PRFS generator and* $\mathsf{Test}(k, x, \cdot)$ *denote the tester algorithm for* $G(k, x)$ *given by Lemma 1. There exists a negligible function* $\nu(\cdot)$ *such that for all* λ*, for all* $x \neq y$*,*

$$\Pr_k[\mathsf{Test}(k, x, G(k, x)) = 1] \geq 1 - \nu(\lambda),$$

and

$$\Pr_k[\mathsf{Test}(k, x, G(k, y)) = 1] \leq 2^{-n(\lambda)} + \nu(\lambda).$$

Due to space constraints, we defer the proof of this to the full version.

3 Constructing PRFS from PRS

In this section we present our construction of PRFS generators using PRS generators, which are seemingly weaker objects. As mentioned in the introduction, there is a trivial construction of PRFS from PRS. Let G be a PRS generator. Define the PRFS generator G' with input length $d(\lambda) = O(\log \lambda)$, where $G'_{\lambda'}(k, x) = G_\lambda(k_x)$ with $\lambda' = 2^{d(\lambda)}\lambda$ and k_x denoting the x'th block of λ bits in $k \in \{0,1\}^{\lambda'}$. However, this simple construction is such that the input length is always at most logarithmic in the seed length. This, as far as we can tell, is not very useful for applications.

We are going to present a more interesting construction: we will build a PRFS generator for *any* input length $d(\lambda)$ that is at most constant times $\log \lambda$,

as long as the the output length of the starting PRS generator is at least $2d(\lambda)+\omega(\log\log\lambda)$. Although the input length may appear modest, such PRFS generators are sufficient for most of the applications we consider in this paper. We find it an intriguing question of whether it is possible to construct PRFS generators with longer input lengths from PRS generators in a black box way.

Theorem 5. *Let $d(\lambda), n(\lambda)$ be functions such that $d(\lambda) = O(\log\lambda)$ and $n(\lambda) = d(\lambda)+\omega(\log\log\lambda)$. Let G denote a $(n(\lambda)+d(\lambda))$-PRS generator. Then there exists a $(d(\lambda), n(\lambda))$-PRFS generator F with the recognizable abort property, such that for all λ the circuit F_λ invokes the G_λ as a black box.*

The rest of this section is dedicated to proving the theorem. For notational clarity we use the abbreviations $d = d(\lambda)$ and $n = n(\lambda)$.

The construction of the PRFS generator is given by the following circuit $F_\lambda(k, x)$. On input key $k \in \{0,1\}^\lambda$, input $x \in \{0,1\}^d$, repeat the following $2^d \cdot \lambda$ times:

- Compute the $(d+n)$-qubit state $\rho_k \leftarrow G_\lambda(k)$.
- Measure the first d qubits of ρ_k in the computational basis to obtain a string $y \in \{0,1\}^d$. If $y = x$, then output the remaining n qubits. Otherwise, continue.

If the measurement outcomes was different from x in all the $2^d\lambda$ iterations, set $\sigma_{k,x} = |\bot\rangle\langle\bot|$. Let the output be $\sigma_{k,x}$.

The algorithm $F = \{F_\lambda\}_\lambda$ is uniform QPT because for each λ, the running time of the circuit F_λ is going to be $O(2^d \cdot \lambda)$ times the complexity of running G_λ, which is QPT since $d = O(\log\lambda)$ and G is QPT. It is easy to see that even if G (as a PRFS generator) only satisfies recognizable abort (instead of perfect generation), F still satisfies recognizable abort by construction. Therefore, the construction also works with the PRS generator constructed by Brakerski and Shmueli [12].

Due to space constraints, we defer the proof of security to the full version.

4 Quantum Pseudo One-Time Pad from PRFS

The first application of PRFS we present is the Quantum Pseudo One-Time Pad (QP-OTP). In classical cryptography, a pseudo one-time pad is like the one-time pad except the key length is shorter than the length of the plaintext message. This is often presented in introductory cryptography courses as a basic example of using pseudorandomness to achieve a cryptographic task that is impossible in the information-theoretic setting. Here, we use a PRFS in place of a PRG to encrypt (classical) messages.

We point out that without knowing about the notion of PRFS, it appears difficult and challenging to construct secure quantum one-time pad schemes directly from PRS generators alone.

Definition 5 (Quantum Pseudo One-Time Pad). *We say that a pair of QPT algorithms* $(\mathsf{Enc}, \mathsf{Dec})$ *is a* quantum pseudo one-time pad (QP-OTP) *for messages of length* $\ell(\lambda) > \lambda$ *for some polynomial* $\ell(\cdot)$ *if the following properties are satisfied:*

- ***Correctness****: There exists a negligible function* $\varepsilon(\cdot)$ *such that for every* λ*, every* $x \in \{0,1\}^{\ell}$*,*

$$\Pr_{\substack{k \leftarrow \{0,1\}^{\lambda}, \\ \sigma \leftarrow \mathsf{Enc}_{\lambda}(k,x)}} [\mathsf{Dec}_{\lambda}(k,\sigma) = x] \geq 1 - \varepsilon(\lambda).$$

- ***Security****: There exist a polynomial* $n(\cdot)$ *such that for every polynomial* $t(\cdot)$*, for every nonuniform QPT adversary* A*, there exists a negligible function* $\varepsilon(\cdot)$ *where for every* λ *and* $x \in \{0,1\}^{\ell}$*,*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^{\lambda}, \\ \sigma \leftarrow \mathsf{Enc}_{\lambda}(k,x)}} \left[A_{\lambda}(\sigma^{\otimes t}) = 1\right] - \Pr_{|\vartheta_1\rangle,\ldots,|\vartheta_{\ell}\rangle \leftarrow \mathscr{H}_n} \left[A_{\lambda}((|\vartheta_1\rangle \otimes \cdots \otimes |\vartheta_{\ell}\rangle)^{\otimes t}) = 1\right] \right| \leq \varepsilon(\lambda),$$

  where we have abbreviated $n = n(\lambda)$*,* $\ell = \ell(\lambda)$*, and* $t = t(\lambda)$*.*

Here the security holds even if the adversary could see multiple copies of the same ciphertexts, which might be useful for certain applications, for example when the communication channel is adversarially lossy. However, when $t = 1$, we can see that the security implies that the ciphertext is computationally indistinguishable to random bit strings of length ℓn (or a maximally mixed state).

To construct such a quantum pseudo one-time pad, let G be a $(d(\lambda), n(\lambda))$-PRFS generator where $d(\lambda) \geq \lceil \log \ell(\lambda) \rceil + 1$ and $n(\lambda) = \omega(\log \lambda)$. We interpret $G_{\lambda}(k,\cdot)$ as taking inputs of the form (i,b) where $i \in [\ell(\lambda)]$ and $b \in \{0,1\}$. Let Test denote the test algorithm from Lemma 2.

Fix λ and let $\ell = \ell(\lambda)$, $d = d(\lambda)$, and $n = n(\lambda)$.

1. $\mathsf{Enc}_{\lambda}(k,x)$: on input $k \in \{0,1\}^{\lambda}$ and a message $x \in \{0,1\}^{\ell}$, do the following:
   - For every $i \in [\ell]$, compute $\sigma_i \leftarrow G_{\lambda}(k,(i,x_i))$.
   - Set $\sigma = \sigma_1 \otimes \cdots \otimes \sigma_{\ell}$.
   - Output the ciphertext state σ.
2. $\mathsf{Dec}_{\lambda}(k,\sigma)$: on input k, ℓn-qubit ciphertext state σ, perform the following operations:
   - Parse σ as $\sigma_1 \otimes \cdots \otimes \sigma_{\ell}$.
   - For $i \in [\ell]$, execute $\mathsf{Test}(k,(i,0),\sigma_i)$. If it accepts, set $x_i = 0$. Otherwise, set $x_i = 1$.
   - Output $x = x_1 \cdots x_{\ell}$.

Lemma 3. $(\mathsf{Enc}, \mathsf{Dec})$ *satisfies the correctness property of a quantum pseudo one-time pad according to Definition 5.*

Proof. Fix λ and let $\ell = \ell(\lambda)$. Fix a message $x \in \{0,1\}^\ell$. Let $\sigma_{k,i} = G_\lambda(k,(i,x_i))$ and let $\sigma_k = \sigma_{k,1} \otimes \cdots \otimes \sigma_{k,\ell}$.

Consider the decryption process. Fix an index $i \in [\ell]$. By Lemma 2, the probability that $\mathsf{Test}\Big(k,(i,0),\sigma_{k,i}\Big)$ accepts (on average over k) is negligibly close to 1 if $x_i = 0$, and it is negligibly close to 0 if $x_i = 1$, on average over the key k (here we use the fact that the output length of the PRFS generator is $\omega(\log\lambda)$, so that $2^{-n(\lambda)}$ is negligible). Thus the probability that the correct bit x_i gets decoded is negligibly close to 1. Taking a union bound over all indices $i \in [\ell]$, we get that the probability of decoding x is negligibly close to 1, over the randomness of the key k and the decryption algorithm. □

Lemma 4. $(\mathsf{Enc},\mathsf{Dec})$ *satisfies the security property of quantum pseudo one-time pad according to Definition 5.*

Proof. We prove the security via a hybrid argument. Let $n(\lambda)$ denote the output length of the PRFS generator G. Fix λ, and let $\ell = \ell(\lambda)$, $n = n(\lambda)$, and $t = t(\lambda)$. Fix a message $x \in \{0,1\}^\ell$. Consider a nonuniform QPT adversary A such that A_λ takes as input t copies of an ℓn-qubit density matrix σ.

Hybrid H_1. Sample $k \leftarrow \{0,1\}^\lambda$. Compute $\sigma \leftarrow \mathsf{Enc}_\lambda(k,x)$. The output of the hybrid is the output of the adversary A_λ on input $\sigma^{\otimes t}$.

Hybrid H_2. Consider the following QPT algorithm B_λ: it takes as input $(i_1,b_1),\ldots,(i_\ell,b_\ell) \in [\ell]\times\{0,1\}$ and a tn-qubit state $\sigma_1^{\otimes t}\otimes\cdots\otimes\sigma_\ell^{\otimes t}$. The algorithm B runs the adversary A_λ on input $(\sigma_1\otimes\cdots\otimes\sigma_\ell)^{\otimes t}$ and returns A_λ's output.

Sample $k \leftarrow \{0,1\}^\lambda$. Compute t copies of $\sigma \leftarrow \mathsf{Enc}_\lambda(k,x)$. The output of this hybrid is the output of B_λ on input $((1,x_1),\ldots,(\ell,x_\ell))$ and $\sigma^{\otimes t} = \sigma_1^{\otimes t}\otimes\cdots\otimes\sigma_\ell^{\otimes t}$.

Hybrid H_3. Sample t copies of Haar-random states $|\vartheta_1\rangle,\ldots,|\vartheta_\ell\rangle \leftarrow \mathscr{H}_n$. The output of this hybrid is the output of B_λ on input $((1,x_1),\ldots,(\ell,x_\ell))$ and $|\vartheta_1\rangle^{\otimes t}\otimes\cdots\otimes|\vartheta_\ell\rangle^{\otimes t}$.

We now argue the indistinguishability of the hybrids. Clearly, hybrids H_1 and H_2 are identical by construction (the adversary B_λ ignores its first input and runs A_λ on input $\sigma^{\otimes t}$). Hybrids H_2 and H_3 are indistinguishable because of the pseudorandomness property of the PRFS generator G. Notice that, by construction, the output of hybrid H_3 is $A_\lambda((|\vartheta_1\rangle\otimes\cdots\otimes|\vartheta_\ell\rangle)^{\otimes t})$. □

5 Quantum Bit Commitments from PRFS

5.1 Definition

We consider the notion of quantum commitment scheme with statistical binding and computational hiding property. This is analogous to a classical commitment scheme where the messages are allowed to be quantum states. We in particular focus on bit commitments where the the committed message is a single bit.

We can generically achieve commitments of long messages by composing many instantiations of the bit-commitment scheme in parallel.

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 $\mu \in \{0, 1, \bot\}$ 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 define the properties satisfied by a commmitment scheme.

Statistical Binding. We start by discussing the statistical binding property. The classical statistical binding property could be rephrased as the following in the quantum setting: for any adversarial (possibly unbounded) committer C^*_λ, we require that at the end of the commit phase, with high probability over the measurement randomness of the receiver, there is a unique bit that C^*_λ can decommit to in the reveal phase. Unfortunately, this idealistic notion is not always possible to achieve: in some quantum commitment protocols where the receiver does not measure everything, it is possible for C^*_λ to send a uniform superposition of commitments of 0 and 1 and later can open to either 0 or 1 with equal probability. This attack was observed and taken into account in many works, including but not limited to [30,28,17,9].

To account for this issue, we consider a notion where an extraction procedure can be applied on the state of the receiver after the commit phase. The output is the receiver's post-extraction state along with the extracted bit b. We revise the statistical binding property guarantee to informally require the following: (a) whether the extractor is applied or not is imperceivable to the committer and (b) the committer can almost never decommit to $1 - b$ if the extracted bit is b.

Definition 6 (Statistical Binding). *We say that a quantum commitment scheme (C, R) satisfies* statistical binding *if for any (non-uniform) adversarial committer $C^* = \{C^*_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a (possibly inefficient) extractor algorithm $\mathcal{E}$ such that the following holds:*

$$\mathrm{TD}\left(\mathsf{RealExpt}^{C^*}_\lambda, \mathsf{IdealExpt}^{C^*,\mathcal{E}}_\lambda\right) \leq \nu(\lambda),$$

for some negligible function $\nu(\lambda)$, where the experiments $\mathsf{RealExpt}^{C^*}_\lambda$ and $\mathsf{IdealExpt}^{C^*,\mathcal{E}}_\lambda$ are defined as follows.

- $\mathsf{RealExpt}_\lambda^{C^*}$: *Execute the commit phase to obtain the joint state* $\sigma_{C^*R} \leftarrow \mathsf{Commit}\langle C^*_\lambda, R_\lambda\rangle$. *Execute the reveal phase to obtain the trit* $\mu \leftarrow \mathsf{Reveal}\langle C^*_\lambda, R_\lambda, \sigma_{C^*R}\rangle$. *Output the pair* (τ_{C^*}, μ) *where* τ_{C^*} *is the final state of the committer.*
- $\mathsf{IdealExpt}_\lambda^{C^*,\mathcal{E}}$: *Execute the commit phase to obtain the joint state* $\sigma_{C^*R} \leftarrow \mathsf{Commit}\langle C^*_\lambda, R_\lambda\rangle$. *Apply the extractor* $I \otimes \mathcal{E}$ *on* σ_{C^*R} *(acting* only on the receiver's part*) to obtain a new joint committer-receiver state* σ'_{C^*R} *along with* $b' \in \{0, 1, \bot\}$. *Execute the reveal phase to obtain the trit* $\mu \leftarrow \mathsf{Reveal}\langle C^*_\lambda, R_\lambda, \sigma'_{C^*R}\rangle$. *Let* τ_{C^*} *denote the final state of the committer. If* $\mu = \bot$ *or* $\mu = b'$, *then output* (τ_{C^*}, μ). *Otherwise, output a special symbol* $\mathfrak{E}$ *(unused in the real experiment) indicating extraction error.*

Remark 1. Many prior works consider statistical binding for quantum commitments. We highlight the main differences between our definition and the prior notions.

- Comparison with [30,28,17]: the statistical binding property is formalized by requiring the states of the (honest) committer when committing to bits 0 and 1 to be far in trace distance. While their definition is cleaner (and probably equivalent to our notion), in our opinion, it is unwieldy to use their definition for applications. Specifically, one has to either implicitly or explicitly come up with an extractor in the security proofs for applications [30,17] and moreover, show that the indistinguishability of the real and the ideal world holds against dishonest committers. On the other hand, we incorporate these technical difficulties as requirements in our definition making it easier to use in applications.
  Another downside of the statistical binding property there is that in order for the sum-binding property to be useful in applications, it is common to additionally require the opening phase to follow the "canonical" opening protocol, where the committer sends the purification of the mixed state sent in the committing phase, and the receiver performs a rank-1 projection to check the state. This implies that *both* parties must keep their part of the state coherent between the two phases. However, our definition gives the flexibility of the reveal phase having purely classical communication.
- Comparison with [9]: A related work by [9] considers statistical binding of quantum commitments called classical binding. The main difference is the following. In their notion, the honest receiver applies a measurement that collapses the commitment into a quantum state and a classical string in such a way that the classical string information theoretically determines the message. They then use this feature to show that in some applications, the opening of the commitment can be classical. Our definition is also more general in the sense that the honest receiver is not required to do any measurement and the collapsing happens implicitly in the ideal world during the execution of extractor.

Remark 2. One has to be careful when using quantum commitments in a larger system if the receiver's state is quantum after the commit phase. As an example, suppose we design a protocol where the quantum commitment held by the

receiver before the reveal phase is used inside another cryptographic protocol. Then we might not be able to invoke binding if the state is destroyed, whereas classically the state could always be copied. Nevertheless, this is a generic caveat of quantum commitments and is not an artifact of any specific definition of binding.

Computational Hiding. We define the computational hiding property below. This is the natural quantum analogue of the classical computational hiding property. In the literature, this property is also sometimes referred to as quantum concealing.

Definition 7 (Computational Hiding). *We say that a quantum commitment scheme (C, R) satisfies computational 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:*

$$\Big|\Pr\left[D_\lambda(\sigma_{R^*}) = 1 \ : \ \sigma_{CR^*} \leftarrow \mathsf{Commit}\langle C_\lambda(0), R^*_\lambda\rangle\right] - \Pr\left[D_\lambda(\sigma_{R^*}) = 1 \ : \ \sigma_{CR^*} \leftarrow \mathsf{Commit}\langle C_\lambda(1), R^*_\lambda\rangle\right]\Big| \leq \nu(\lambda),$$

for some negligible function $\nu(\cdot)$, where σ_{R^*} is obtained by tracing out the committer's part of the state σ_{CR^*}.

5.2 Construction

We now present the main theorem of this section, which shows that statistically binding quantum commitment schemes can be constructed from PRFS.

Theorem 6. *Assuming the existence of $(d(\lambda), n(\lambda))$-PRFS satisfying recognizable abort (Definition 4) with $2^d \cdot n \geq 7\lambda$, there exists a commitment scheme satisfying statistical completeness, statistical binding (Definition 6) and computational hiding (Definition 7).*

We note that, combined with Theorem 5 which constructs PRFS generators with $\Omega(\log\lambda)$ input length and recognizable abort property from PRS generators, we can obtain quantum commitment schemes from PRS generators. We present the construction, which is inspired by Naor's commitment scheme [27].

The main building block is a $(d(\lambda), n(\lambda))$-PRFS, denoted by $G = \{G_\lambda(\cdot,\cdot)\}_{\lambda\in\mathbb{N}}$. Since $n \geq 1$, we assume $d(\lambda) = \lceil \log \frac{7\lambda}{n} \rceil = O(\log\lambda)$ to ensure the efficiency of the algorithm. This is without loss of generality since we can generically shrink the input length for a PRFS by padding zeroes. Let $\mathsf{Test}_\lambda^{\otimes 2^{d(\lambda)}}$ be the product PRFS tester corresponding to G as guaranteed by Corollary 4.

We describe the commitment scheme, (C, R) as follows. For notational convenience, we abbreviate $n = n(\lambda)$, $d = d(\lambda)$.

1. *Commit Phase*:

- The receiver R_λ samples a uniformly random m-qubit Pauli operator P, where $m = 2^d \cdot n$. We write P as $P_0 \otimes \cdots \otimes P_{2^d-1}$, where P_i is an n-qubit Pauli operator[14]. It sends P to the committer.
- The committer C_λ on input a bit $b \in \{0,1\}$, does the following:
  - It samples $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - For every $x \in \{0,1\}^d$, computes $\sigma_{k,x} \leftarrow G_\lambda(k,x)$.

  It sends the commitment $\mathbf{c} = \bigotimes_{x\in\{0,1\}^d} \widetilde{\sigma}_{k,x}$, where $\widetilde{\sigma}_{k,x} = P_x^b \sigma_{k,x} P_x^b$, to the receiver.

2. *Reveal Phase*: The committer sends $(k,b) \in \{0,1\}^\lambda \times \{0,1\}$ as the decommitment to the receiver. The receiver outputs b if and only if $\mathsf{Test}_\lambda^{\otimes 2^d}\left(\{k,x\}_x, P^b \mathbf{c} P^b\right) = 1$ where $P^b = \bigotimes_{x\in\{0,1\}^{2^d}} P_x^b$. Otherwise the receiver outputs $\perp$.

Lemma 5. *If G is a PRFS, then there exists a negligible function $\nu(\cdot)$ such that the probability that the honest receiver accepts the honest committer's opening is at least $1 - \nu(\lambda)$.*

Proof. This follows immediately from Lemma 2 and union bound as 2^d is polynomial in λ. □

Due to space constraints, we defer the security proof of this construction to the full version.

5.3 Application: Secure Computation

In this section, we show how to base secure computation solely on the existence of a PRS. While there are two works [4,19] showing that post-quantum one-way functions and quantum communication suffice to obtain protocols for secure computation, the construction of Bartusek, Coladangelo, Khurana, and Ma [4] has the advantage that it uses the starting commitment scheme as a black box. We recall their main theorem.

Theorem 7 (Implicit from [4]). *Assuming the existence of quantum statistically binding bit commitments, maliciously secure computation protocols (in the dishonest majority setting) for P/poly exist.*

Comparison of the definitions of statistical binding. The application of Theorem 7 would be straightforward except for one subtlety, which is that we are using a more general definition of the statistical binding property than required by their work. Their notion of statistical binding is tailored to commitment schemes with classical messages as it suffices for their purposes. We first recall their definition of statistical binding in the full version of their work [5], and show that it seems strictly stronger than our definition.

[14] To sample $P = \bigotimes_i P_i$, the receiver can sample uniformly random bits $\alpha_1, \beta_1, \ldots, \alpha_m, \beta_m$, and let $P_i = X^{\alpha_i} Z^{\beta_i}$ where X and Z are the single-qubit Pauli operators.

Definition 8 ([5, Definition 3.2]). *A bit commitment scheme is statistically binding if for every unbounded-size committer $\mathcal{C}^*$, there exists a negligible function $\nu(\cdot)$ such that with probability at least $1-\nu(\lambda)$ over the measurement randomness in the commitment phase, there exists a bit $b \in \{0,1\}$ such that the probability that the receiver accepts b in the reveal phase is at most $\nu(\lambda)$.*

Lemma 6. *If a commitment scheme satisfies Definition 8, then it also satisfies Definition 6.*

Proof. Since a malicious committer can always "purify" his measurements via the deferred measurement principle, without loss of generality we assume the only measurements in the commit phase are only done by the honest receiver. By Definition 8, there exists a classical function $\mathcal{E}$ that maps the honest receiver's measurement outcomes m to a bit so that the probability that the receiver accepts $1-\mathcal{E}(m)$ is negligible (also known as the correctness of the extractor). As $\mathcal{E}$ only acts on the measurement outcome that is therefore guaranteed to be classical, $\mathcal{E}$ commutes with the committer's and receiver's operations. Furthermore, the output of $\mathcal{E}$ is also classical by definition. Therefore, the only difference between the real world and the ideal world is introduced by the extraction error in the ideal world, and thus the statistical indistinguishability follows immediately by the correctness of the extractor. □

Our protocol cannot satisfy this property since the honest receiver does not measure the committer's message in any way, and therefore in general it is possible for the committer to generate an equal superposition of commitment to 0 and commitment to 1, in which case this binding property is violated, as the receiver will open to 0 and 1 with equal probability. Nonetheless, Definition 6 is very similar to Definition 8. In particular, Definition 6 says that there is an implicit measurement that could be done to extract the committed bit in a way unnoticeable to the malicious committer as well as the honest receiver. Intuitively, whenever we would like to invoke Definition 8, we can switch to the ideal world where the bit is extracted, and then this "ideal scheme" essentially satisfies Definition 8. We formalize this intuition with the following lemma.

Definition 9. *We call (C,R) a quantum commitment scheme with an inefficient receiver if it satisfies the requirements of a commitment scheme except that R need not be a QPT algorithm.*

Let (C,R) and (C,R') be two quantum commitment schemes with an inefficient receiver. We call them statistically indistinguishable against malicious committers, if the outcome of any (unbounded) nonuniform experiment described below can only distinguish R from R' with negligible advantage.

- *The algorithm has an arbitrary non-uniform input state $|\psi_\lambda\rangle$, and interacts as a committer with either R or R' via the commitment scheme.*
- *The algorithm can choose to abort the interaction at any stage. Otherwise at the end of the interaction, R or R' outputs his decision as a classical symbol $\mu \in \{0,1,\perp\}$ to the algorithm.*

– *The algorithm performs an arbitrary channel on his internal state as the output.*

Lemma 7. *If a commitment scheme (C, R) satisfies Definition 6, then there is a commitment scheme $(C, \tilde{R})$ with an inefficient receiver that satisfies Definition 8. Furthermore, these two commitment schemes are statistically indistinguishable against malicious committers; and $\tilde{R}$ is the same as R, except that at the end of the commit phase, the extractor $\mathcal{E}$ of (C, R) is applied on the receiver's state, and its output is saved in a separate register.*

Proof. Note that $(C, \tilde{R})$ is the same receiver as the ideal experiment of Definition 6, except that at the end we always run the honest receiver as usual instead of checking whether the extraction is correct, and therefore this change is statistically indistinguishable to the committer by Definition 6.

To show that it satisfies Definition 8, we notice that assume the extractor's measurement outcome is b (if it is $\perp$ then set b to 0), the probability that the committer can open to $1-b$ is negligible, as otherwise the ideal world will have a non-negligible weight on extraction error $|\mathfrak{E}\rangle\langle\mathfrak{E}|$, which contradicts Definition 6. □

It is not hard to see that by leveraging Lemmas 6 and 7, we can recover Theorem 7 even with our definition of statistical binding (Definition 6). The proof of this is not very enlightening and we defer the details to the full version. By instantiating the statistically binding bit commitments in Theorem 7 with PRS (Theorem 6 and Theorem 5), we obtain the following corollary.

Corollary 5. *Assuming the existence of $(2\log\lambda+\omega(\log\log\lambda))$-PRS, there exists maliciously secure computation protocol for P/poly in the dishonest majority setting.*

References

1. Aaronson, S.: Quantum computing, postselection, and probabilistic polynomial-time. Proceedings: Mathematical, Physical and Engineering Sciences **461**(2063), 3473–3482 (2005), http://www.jstor.org/stable/30047928
2. Ananth, P., Qian, L., Yuen, H.: Manuscript (in preparation). (2022)
3. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G.S.L., Buell, D.A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M.P., Hartmann, M.J., Ho, A., Hoffmann, M., Huang, T., Humble, T.S., Isakov, S.V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P.V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandrà, S., McClean, J.R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M.Y., Ostby, E., Petukhov, A., Platt, J.C., Quintana, C., Rieffel, E.G., Roushan, P., Rubin, N.C., Sank, D., Satzinger, K.J., Smelyanskiy, V., Sung, K.J., Trevithick, M.D., Vainsencher, A., Villalonga, B., White, T., Yao, Z.J., Yeh, P., Zalcman, A., Neven, H., Martinis,

J.M.: Quantum supremacy using a programmable superconducting processor. Nature **574**(7779), 505–510 (10 2019). https://doi.org/10.1038/s41586-019-1666-5

4. Bartusek, J., Coladangelo, A., Khurana, D., Ma, F.: One-way functions imply secure computation in a quantum world. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 467–496. Springer (2021). https://doi.org/10.1007/978-3-030-84242-0_17
5. Bartusek, J., Coladangelo, A., Khurana, D., Ma, F.: One-way functions imply secure computation in a quantum world (2021)
6. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols (extended abstract). In: Ortiz, H. (ed.) Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA. pp. 503–513. ACM (1990). https://doi.org/10.1145/100216.100287
7. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of International Conference on Computers, Systems & Signal Processing, Dec. 9-12, 1984, Bangalore, India. pp. 175–179 (1984)
8. Bennett, C.H., Brassard, G., Crépeau, C., Skubiszewska, M.: Practical quantum oblivious transfer. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings. Lecture Notes in Computer Science, vol. 576, pp. 351–366. Springer (1991). https://doi.org/10.1007/3-540-46766-1_29
9. Bitansky, N., Brakerski, Z.: Classical binding for quantum commitments. In: Nissim, K., Waters, B. (eds.) Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13042, pp. 273–298. Springer (2021). https://doi.org/10.1007/978-3-030-90459-3_10
10. Bouland, A., Fefferman, B., Vazirani, U.V.: Computational pseudorandomness, the wormhole growth paradox, and constraints on the ads/cft duality (abstract). In: Vidick, T. (ed.) 11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA. LIPIcs, vol. 151, pp. 63:1–63:2. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPIcs.ITCS.2020.63
11. Brakerski, Z., Shmueli, O.: (pseudo) random quantum states with binary phase. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11891, pp. 229–250. Springer (2019). https://doi.org/10.1007/978-3-030-36030-6_10
12. Brakerski, Z., Shmueli, O.: Scalable pseudorandom quantum states. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12171, pp. 417–440. Springer (2020). https://doi.org/10.1007/978-3-030-56880-1_15
13. Brakerski, Z., Yuen, H.: Quantum garbled circuits (2020)
14. Brandão, F.G., Chemissany, W., Hunter-Jones, N., Kueng, R., Preskill, J.: Models of quantum complexity growth. PRX Quantum **2**, 030316 (7 2021). https://doi.org/10.1103/PRXQuantum.2.030316
15. Brown, W., Fawzi, O.: Scrambling speed of random quantum circuits (2013)

16. Chia, N., Chou, C., Zhang, J., Zhang, R.: Quantum meets the minimum circuit size problem. In: Braverman, M. (ed.) 13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA. LIPIcs, vol. 215, pp. 47:1–47:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.ITCS.2022.47
17. Fang, J., Unruh, D., Yan, J., Zhou, D.: How to base security on the perfect/statistical binding property of quantum bit commitment? Cryptology ePrint Archive, Report 2020/621 (2020), https://ia.cr/2020/621
18. Goldreich, O.: A note on computational indistinguishability. Information Processing Letters **34**(6), 277–281 (1990). https://doi.org/10.1016/0020-0190(90)90010-U
19. Grilo, A.B., Lin, H., Song, F., Vaikuntanathan, V.: Oblivious transfer is in miniqcrypt. In: Canteaut, A., Standaert, F. (eds.) Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12697, pp. 531–561. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_18
20. Harrow, A., Mehraban, S.: Approximate unitary t-designs by short random quantum circuits using nearest-neighbor and long-range gates (2018)
21. Impagliazzo, R.: A personal view of average-case complexity. In: Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995. pp. 134–147. IEEE Computer Society (1995). https://doi.org/10.1109/SCT.1995.514853
22. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Johnson, D.S. (ed.) Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA. pp. 44–61. ACM (1989). https://doi.org/10.1145/73007.73012
23. Ji, Z., Liu, Y., Song, F.: Pseudorandom quantum states. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10993, pp. 126–152. Springer (2018). https://doi.org/10.1007/978-3-319-96878-0_5
24. Kilian, J.: Founding cryptography on oblivious transfer. In: Simon, J. (ed.) Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA. pp. 20–31. ACM (1988). https://doi.org/10.1145/62212.62215
25. Kretschmer, W.: Quantum pseudorandomness and classical complexity. In: Hsieh, M. (ed.) 16th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2021, July 5-8, 2021, Virtual Conference. LIPIcs, vol. 197, pp. 2:1–2:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.TQC.2021.2
26. Morimae, T., Yamakawa, T.: Quantum commitments and signatures without one-way functions (2021). https://doi.org/10.48550/ARXIV.2112.06369
27. Naor, M.: Bit commitment using pseudorandomness. Journal of Cryptology **4**(2), 151–158 (1 1991). https://doi.org/10.1007/BF00196774
28. Unruh, D.: Computationally binding quantum commitments. In: Fischlin, M., Coron, J. (eds.) Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9666, pp. 497–527. Springer (2016). https://doi.org/10.1007/978-3-662-49896-5_18

29. Wiesner, S.: Conjugate coding. SIGACT News **15**(1), 78–88 (1983). https://doi.org/10.1145/1008908.1008920
30. Yan, J., Weng, J., Lin, D., Quan, Y.: Quantum bit commitment with application in quantum zero-knowledge proof (extended abstract). In: Elbassioni, K.M., Makino, K. (eds.) Algorithms and Computation - 26th International Symposium, ISAAC 2015, Nagoya, Japan, December 9-11, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9472, pp. 555–565. Springer (2015). https://doi.org/10.1007/978-3-662-48971-0_47
31. Zhu, Q., Cao, S., Chen, F., Chen, M.C., Chen, X., Chung, T.H., Deng, H., Du, Y., Fan, D., Gong, M., Guo, C., Guo, C., Guo, S., Han, L., Hong, L., Huang, H.L., Huo, Y.H., Li, L., Li, N., Li, S., Li, Y., Liang, F., Lin, C., Lin, J., Qian, H., Qiao, D., Rong, H., Su, H., Sun, L., Wang, L., Wang, S., Wu, D., Wu, Y., Xu, Y., Yan, K., Yang, W., Yang, Y., Ye, Y., Yin, J., Ying, C., Yu, J., Zha, C., Zhang, C., Zhang, H., Zhang, K., Zhang, Y., Zhao, H., Zhao, Y., Zhou, L., Lu, C.Y., Peng, C.Z., Zhu, X., Pan, J.W.: Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. Science Bulletin **67**(3), 240–245 (2022). https://doi.org/10.1016/j.scib.2021.10.017