Separating Pseudorandom Generators from Logarithmic Pseudorandom States

Mohammed Barhoush

Université de Montréal (DIRO), Montréal, Canada
mbarhoush18@gmail.com

Abstract. Pseudorandom generators (PRGs) are a foundational primitive in classical cryptography, underpinning a wide range of constructions. In the quantum setting, pseudorandom quantum states (PRSs) were proposed as a potentially weaker assumption that might serve as a substitute for PRGs in cryptographic applications. Two primary size regimes of PRSs have been studied: logarithmic-size and linear-size. Interestingly, logarithmic PRSs have led to powerful cryptographic applications, such as digital signatures and quantum public-key encryption with tamper-resilient keys, that have not been realized from their linear counterparts. However, PRGs have only been black-box separated from linear PRSs, leaving open the fundamental question of whether PRGs are also separated from logarithmic PRSs.

In this work, we resolve this open problem. We establish a quantum black-box separation between (quantum-evaluable) PRGs and PRSs of either size regime. Specifically, we construct a unitary quantum oracle with inverse access relative to which no black-box construction of PRG from (logarithmic or linear) PRS exists.

This does not directly separate PRG from some of the applications of SPRS since these applications involve, as a first step, a non-black-box construction of a notion termed $\perp$-PRGs. To address this, we present another unitary separation showing that PRGs are also separated from $\perp$-PRGs. Thus, we obtain separation from digital signatures and quantum public-key encryption.

1 Introduction

Pseudorandom quantum states (PRS) were introduced by Ji, Liu, and Song [15] as the quantum analogue of classical pseudorandom generators (PRG) [6]. Informally, an n-PRS generator is a quantum polynomial-time algorithm that, on input $k \in \{0,1\}^\lambda$, outputs an n-qubit state $|\phi_k\rangle$ that is computationally indistinguishable from a Haar-random state, even when an adversary is given polynomially many copies of the state.

Two size regimes of PRS have emerged as central: (1) short pseudorandom states (SPRS), where the output length n is logarithmic in λ, and (2) long pseudorandom states (LPRS), where n scales linearly with λ. These regimes display markedly different behavior. In contrast to PRGs, no known reduction connects

SPRS and LPRS. In fact, it was shown that there is no black-box construction of SPRS from LPRS [11], and there is evidence that the reverse direction is also difficult [7].

Both SPRS and LPRS can be built from PRGs [15,9]. In the other direction, Kretschmer [16] proved a black-box separation between PRGs and LPRSs,[1] yet this does not rule out a reduction to SPRSs. His argument relies crucially on a concentration inequality for quantum states (Theorem 5.17 in [18]) whose bound is insufficient in the short-state regime. Our approach does not use this result, thereby circumventing that limitation.

The difference between the two regimes is also reflected in their cryptographic power. Long pseudorandom states enable several important primitives such as quantum pseudo-encryption, quantum bit-commitment protocols, length-restricted one-time signatures with quantum public keys, and private-key quantum money [15,2,19]. Nevertheless, there remains a significant gap between the cryptographic power of LPRSs and PRGs.

Short pseudorandom states, in contrast, exhibit a much tighter connection to classical pseudorandomness: through quantum state tomography, one can extract a classical pseudorandom string from SPRSs [1]. However, tomography is inherently probabilistic, leading to a notion termed *pseudodeterministic* PRG*s*—generators that are deterministic on most inputs but may behave non-deterministically on an inverse-polynomial fraction. [2] Nevertheless, this conversion enables many of the applications of PRGs, such as statistically binding commitments with classical communication.

While pseudodeterministic PRGs proved useful, their nondeterminism complicates their use as direct substitutes for standard PRGs in some applications. To mitigate this, [3] introduced the notion of $\perp$-*pseudodeterministic pseudorandom generators (*$\perp$-PRG*s)*, which they build from pseudodeterministic PRGs by marking nondeterministic outcomes with the symbol $\perp$. This seemingly modest modification provides a clean interface for cryptographic constructions to handle nondeterministic behavior explicitly. As a result, it enables SPRS to realize applications traditionally achievable only through standard PRGs—such as many-time digital signatures and quantum public-key encryption with tamper-resilient keys [1,3]—capabilities still beyond reach for LPRSs.

Despite these advances, no complete quantum fully black-box (BB) separation is currently known between PRGs and SPRSs or even PRGs and $\perp$-PRGs, leaving the precise relationship between them unresolved and somewhat limiting the significance of these applications.

Resolving this question sheds light on the fundamental limits of quantum pseudorandomness and clarifies the relationship between these assumptions. While partial progress has been made (see Section 1.2), a complete BB sepa-

[1] The separation is shown relative to quantum-evaluable one-way functions. Since standard deterministic PRGs are also one-way, the result extends to those PRGs as well.

[2] Different notions of pseudodeterminism appear in the literature [5,3]; we follow the definition of [1].

ration between PRGs and SPRSs or $\perp$-PRGs remains open. The importance of this problem has been repeatedly emphasized in recent works [3,1,4], underscoring its role in understanding the hierarchy of assumptions within the emerging framework of MicroCrypt quantum primitives.

1.1 Our Results

In this work, we resolve this open problem by establishing two black-box separations. Both separations hold relative to a unitary quantum oracle with inverse access, capturing a broad class of black-box constructions [11].

First, we separate PRGs from ℓ-PRSs of any size $\ell \in O(\lambda)$, thereby also ruling out reductions to SPRS.

Theorem 1. *There does not exist a BB construction of a* PRG *from an* ℓ-PRS*, for any positive integer-valued, efficiently computable function* $\ell(\lambda) \in O(\lambda)$*, even with inverse access.*

We consider a general notion of PRGs that is quantum-evaluable and permits a negligible probability of error or non-determinism.

Moreover, prior works have shown that SPRSs imply a rich class of cryptographic primitives. These applications proceed in three conceptual steps:

1. Construct pseudodeterministic PRGs from SPRS [1] *(non-BB).*
2. Construct $\perp$-PRG from pseudodeterministic PRGs [3] *(BB).*
3. Use $\perp$-PRGs to achieve some of the same applications achieved with PRG such as digital signatures [1,3].

The first step, unusually, is a non-BB construction: the correctness of the pseudodeterministic PRG is guaranteed only by the *security* of the underlying SPRS. Hence, the construction violates a requirement of BB reductions—namely, that the correctness of the target primitive must follow from the correctness of the source [14,11]. In our separation, this property is used. Consequently, separating PRGs from SPRSs alone is insufficient to separate PRGs from pseudodeterministic PRGs and their many derived applications.

We therefore give a second separation that directly considers $\perp$-PRGs.

Theorem 2. *There does not exist a BB construction of a* PRG *from an* $\perp$-PRG *(even with inverse access).*

Previous work showed that $\perp$-PRGs can be used to derive many applications [1,3]. As a result, we obtain the following as a direct corollary.

Corollary 1. *There does not exist a BB construction of a* PRG *from any of the following primitives:*

1. *(Many-time) existentially unforgeable digital signatures for classical messages with classical keys and signatures (DS).*
2. *CPA-secure quantum public-key encryption of classical messages with tamper-resilient keys and classical ciphertexts (QPKE).*

We note that a version of Theorem 2 was previously established by [4] using completely positive trace-preserving (CPTP) maps, also ruling out constructions of one-way state generators (see Section 1.2). However, CPTP separations exclude a narrower class of constructions than unitary separations. In particular, a CPTP separation does not rule out reductions that rely on unitary access to the underlying primitive or adversary (see [11,4] for further discussion of these distinctions). In contrast, Theorem 2 and Corollary 1 establish, for the first time, general separations between these fundamental cryptographic primitives, encompassing reductions that may employ unitary or inverse access.

1.2 Related Works

Ananth, Lin, and Yuen [1] were the first to formally investigate applications specific to SPRSs. They showed how to construct pseudodeterministic PRGs from SPRSs, and used them to obtain applications such as statistically binding commitments with classical communication and CPA-secure symmetric encryption with classical ciphertexts.

A follow-up work [3] introduced the notion of $\bot$-PRGs, which they build from pseudodeterministic PRGs, and use to achieve applications such as DS and QPKE.

On the negative side, there are several separations that are relevant. Concurrent with this work, Bouaziz–Ermann, Hhan, Muguruza, and Vu [7] introduced a novel conjecture on product states which, if true, would yield a separation between PRGs and SPRSs. Our result strengthens theirs by establishing the separation unconditionally, without relying on any unproven conjecture. Both separations employ a Haar random oracle. Their conjecture essentially asserts that if a PRG relative to this oracle is oracle-dependent—that is, outputs distinct values with non-negligible probability over the distribution of oracles—then the PRG must be non-deterministic on some oracles [3]. Assuming this conjecture, they can directly rule out the existence of a deterministic PRG, which is the hard part of our proof. We do not prove their conjecture, but establish a closely related guarantee via the continuity of quantum states, which suffices to carry out the separation and constitutes the technical core of this work.

Another closely related separation is due to Barhoush, Nishimaki, and Yamakawa [4], who showed that $\bot$-PRGs are separated from one-way state generators (OWSGs) relative to a non-unitary CPTP oracle. Recall that OWSGs are the quantum analogue of one-way functions (OWFs), and that every OWF is an OWSG. By contrast, our separations are established relative to a unitary oracle. Notably, unitary separations rule out a broader class of constructions than CPTP separations. In particular, CPTP separations do not preclude reductions that rely on unitary or inverse oracle access to the underlying primitive or attack (see [11,4] for further discussion) and, hence, are considered weaker. At the same time, since OWSGs are potentially a weaker primitive than PRGs, their

[3] An updated version of their work proves the conjecture, however this was done after our work.

result separates $\perp$-PRGs from a weaker target, while our result allows stronger oracle access. These two results are therefore incomparable. The reason a CPTP oracle admits a more direct separation in this setting is that it can encode non-determinism directly into the oracle, enabling the construction of inherently non-deterministic primitives such as $\perp$-PRGs while blocking purely deterministic ones. A unitary oracle must provide a coherent, reversible implementation, and allowing inverse access prevents a direct application of that argument. Our approach instead exploits the tension between the continuous nature of quantum states and the discrete nature of classical strings to rule out deterministic behavior.

All in all, both of the previous separation proofs [7,4] instill non-determinism property through either the oracle or through a conjecture. Meanwhile, our techniques show that non-determinism is already inherent in procedures extracting classical randomness from quantum randomness.

There are other notable separations in MicroCrypt, including the following:

- ($\mathsf{LPRS} \not\Rightarrow \mathsf{PRG}$) : Kretschmer [16] presented a separation between PRGs and LPRSs, implying that LPRS constitute a potentially weaker assumption than PRGs.
- ($\mathsf{LPRS} \not\Rightarrow \mathsf{DS}$) : Coladangelo and Mutreja [11] showed that even digital signatures are separated from LPRS.
- ($\mathsf{LPRS} \not\Rightarrow \mathsf{SPRS}$) : This is a direct corollary to the separation above by [11], but it was also shown through an alternative approach by [8].
- (quantum-evaluable OWF $\not\Rightarrow$ classical-evaluable OWF) : Kretschmer, Qian, and Tal [17] show a separation between classical-evaluable OWFs and quantum-evaluable OWFs relative to a classical oracle. As a corollary, they establish that classically-evaluable OWFs are separated from DS and QPKE. However, our results imply that even quantum-evaluable OWFs are separated from these applications. Note that all notions in our work consider quantum-evaluable algorithms.

1.3 Limitations

A primary limitation of our separation is that it focuses exclusively on BB constructions where the correctness of the target primitive is derived from the correctness of the underlying source. This is a natural condition that is usually satisfied in cryptographic constructions. In fact, the original definition of BB constructions [14], as well as its quantum adaptation [11], treats this "correctness inheritance" as a fundamental requirement. Furthermore, existing literature separating quantum primitives [4] similarly relies on this property, meaning those separations likewise do not account for constructions lacking this feature.

Nonetheless, one could envision a broader definition of BB constructions where target correctness may also stem from the security of the source primitive. Our current results do not exclude such cases, and extending our framework to encompass these "looser" constructions remains a compelling direction for future work.

That being said, this does not seem easy to do using our current proof techniques. Our proof relies on source correctness at multiple critical junctures. This may be a natural consequence of the fact that any separation between PRG and $\perp$-PRG needs to very delicately deal with the correctness of the source as this is the only factor separating these two primitives.

2 Technical Overview

We now give an overview of our two separations.

2.1 Separating PRG from SPRS

We describe our separation between PRGs and ℓ-PRSs for any function $\ell(\lambda) \in O(\lambda)$. The discussion below presents a high-level overview that omits several technical subtleties appearing in the full proof.

We use two unitary oracles in the separation:

1. A PSPACE oracle $\mathcal{C}$, which is used to break any PRG.
2. An ℓ-Common Haar Function-Like State (CHFS) oracle [7] $\mathcal{O}$, whose unitary implementation prepares a fixed state $|\phi_x\rangle$ on input $x \in \{0,1\}^*$. The states are sampled independently from the Haar distribution and have size $\ell(|x|)$, where $|x|$ denotes the length of x.

The oracle $\mathcal{O}$ gives a PRS that is secure relative to $(\mathcal{O}, \mathcal{C})$ with probability 1 over the sampled states, by Theorem 4.4 in [7].

Suppose, toward a contradiction, that there exists a BB construction of a PRG from a PRS. This would imply the existence of a PRG $G^{\mathcal{O}}$ in this oracle model that may query $\mathcal{O}$. Because every choice of the states in $\mathcal{O}$ satisfies PRS correctness, G must also satisfy correctness for every such oracle: the output is the same on every seed except with negligible probability. Note, however, that this negligible bound may depend on $\mathcal{O}$.

We first obtain one correctness bound on a suitable family of oracles. Let $T = T(\lambda) \geq 1$ be a polynomial bound on the number of queries made by G, and set $m = \lceil 4\pi T\lambda \rceil + 1$ and $\delta(\lambda) = 1/(100\lambda m)$. Define the correctness error for G when given access to oracle $\mathcal{O}$ as follows:

$$e_{\mathcal{O}}(\lambda) \coloneqq \mathbb{E}_{k \leftarrow \{0,1\}^\lambda} \left[1 - \max_y \Pr[G^{\mathcal{O}}(k) = y] \right]$$

Define the set of oracles with δ correctness error for all security parameters at least N as follows:

$$F_N \coloneqq \{\mathcal{O} : e_{\mathcal{O}}(\lambda) \leq \delta(\lambda) \text{ for all } \lambda \geq N\}.$$

Correctness implies that every oracle belongs to some F_N.

A *prefix* fixes all states at labels $|x| < a$, for some constant a, and a *completion* is an oracle that agrees with these fixed states. We show that some F_N contains every completion of some prefix.

Suppose for contradiction that, no matter which finite prefix we fix and how large an integer N we choose, there is a completion and a parameter $\lambda \geq N$ for which $e_{\mathcal{O}}(\lambda) > \delta(\lambda)$. This essentially gives us an increasing sequence of security parameters $\lambda_1, \lambda_2, \ldots$ where $e_{\mathcal{O}}(\lambda_i) > \delta(\lambda_i)$. We can carefully choose this sequence so that we first fix the behavior of the oracle up to λ_1, and then we search for the extension to λ_2 that satisfies the bound $e_{\mathcal{O}}(\lambda_2) > \delta(\lambda_2)$. Such a parameter exists by our assumption.

We repeat this procedure, extending the prefix and choosing a larger parameter each time. Hence, this sequence is built inductively and defines one oracle for which the correctness bound fails at infinitely many parameters. This contradicts the correctness of G. Therefore, we get that some F_N contains every completion of some prefix.

Define the oracle I to have this prefix that keeps every completion in F_N and let the oracle output $|0^{\ell(|x|)}\rangle$ at all other entries. Queries to I can be approximated efficiently given a classical approximation of the prefix states. Note that the prefix is fixed, so these states have size that is constant in the security parameter, which allows us to efficiently approximate them. Thus, we can approximate oracle access to I.

For each oracle $\mathcal{O}$ with this fixed prefix, we construct a sequence $\mathcal{O}_1, \ldots, \mathcal{O}_m$ from $\mathcal{O}_1 = \mathcal{O}$ to $\mathcal{O}_m = I$. Keeping the prefix unchanged, we move each full state vector, including its phase, along a shortest path on the unit sphere toward its value in I. Each path has length at most π. Dividing these paths into $m-1$ equal steps makes consecutive vectors close in Euclidean distance, so the corresponding swap unitaries satisfy

$$\|\mathcal{O}_i - \mathcal{O}_{i+1}\|_{\mathrm{op}} \leq \frac{4\pi}{m-1}. \tag{1}$$

Note that in this hybrid argument, we use the continuity property of quantum states, which does not hold for classical strings. In particular, we use continuity to make these small modifications to the states in an undetectable manner from the perspective of the generator. G queries the oracle at most T times, so we get that for every seed k and $i \in [m-1]$,

$$\mathsf{Tr}\left(G^{\mathcal{O}_i}(k), G^{\mathcal{O}_{i+1}}(k)\right) \leq \frac{4\pi T}{m-1} \leq \frac{1}{\lambda}. \tag{2}$$

This also handles inverse queries, since the swap oracles are self-inverse.

For any seed, the output of G cannot change between consecutive oracles: G is deterministic, so two different outputs would contradict Eq. (2). Thus $G^{\mathcal{O}}$ and G^I have the same output with high probability on almost all seeds.

Since I is efficiently simulatable, a PSPACE computation can estimate the output probabilities of G^I. This gives a constant distinguishing advantage, contradicting the security of the PRG.

2.2 Separating PRG from $\perp$-PRG

We now describe how to separate PRG from $\perp$-PRG using a unitary quantum oracle with inverse access. In particular, we use two oracles:

1. A PSPACE oracle, used to break any PRG.
2. A modified quantum random oracle with an abort mechanism $\mathcal{O}$, designed to satisfy the pseudodeterminism property of $\perp$-PRGs. At each input length d, an inverse-polynomial fraction $\mu(d)$ of the inputs may produce either a fixed string or the symbol $\perp$; all remaining inputs produce a fixed string.

It is straightforward to show that $\mathcal{O}$ is a valid $\perp$-PRG. Suppose, toward a contradiction, that there exists a BB construction of a PRG from a $\perp$-PRG. This would imply the existence of a PRG $G^{\mathcal{O}}$.

The first step is similar to that of the previous proof. We establish for the oracle $\mathcal{O}$ the existence of an integer N and a prefix such that every completion of the prefix is in F_N.

The issue is the second step of the previous proof where we applied a hybrid argument using the continuity of quantum states to show that for any oracle with some prefix, the PRG can be simulated efficiently without oracle access. However, for this separation, we cannot use the continuity of quantum states since the oracle no longer outputs quantum states.

Instead, we apply a similar argument using the non-determinism of the oracle $\mathcal{O}$. In particular, $\mathcal{O}$ has an inverse-polynomial fraction μ of inputs that are allowed to be non-deterministic. On such inputs, the output can be $\perp$ with any arbitrary probability. Hence, we can utilize the continuity of this $\perp$ probability in place of the continuity of quantum states. Note, however, that we cannot simply vary the probability of $\perp$ on all inputs, since only a μ fraction of inputs are allowed to output $\perp$.

More specifically, for each non-deterministic input, we gradually increase the abort probability to 1, then change its fixed string to 0 while it always aborts, and then gradually decrease the abort probability to 0. The unitary implementation of the oracle varies continuously along these changes. This is because we can change the non-$\perp$ output to any value when the probability of outputting $\perp$ is 1 without changing the output of the oracle. These modifications have converted the non-deterministic inputs to return 0 with probability 1.

Now the oracle is completely deterministic, so it gives us the ability to choose another μ fraction of inputs and start increasing the probability of outputting $\perp$ until it is 1, then change the non-$\perp$ output to 0, and then gradually decrease the probability of $\perp$ to 0.

We repeat this process for all blocks at each length that G may query. There are only polynomially many relevant blocks because μ is inverse-polynomial. Subdividing each change finely enough therefore gives the required polynomial-length sequence of hybrids that allows us to show that $G^{\mathcal{O}}$ can be simulated efficiently with some degree of accuracy.

The PSPACE test then distinguishes the output of $G^{\mathcal{O}}$ from uniform with constant advantage. This contradicts the assumed BB construction.

Acknowledgments: We would like to thank Samuel Bouaziz–Ermann for helpful discussions and an anonymous reviewer for finding a small issue in a previous version of this work, which we have corrected. We also acknowledge the use of

ChatGPT Astra specifically for this version of the work for helping us simplify the fixed-prefix argument and obtain a shorter proof.

3 Preliminaries

3.1 Notation

We let $[n] = \{1, 2, \ldots, n\}$ and let $\mathsf{negl}(x)$ denote any nonnegative function that is asymptotically smaller than the inverse of any positive polynomial. All logarithms are base 2.

We let $x \leftarrow X$ denote that x is sampled according to the distribution X. If X is a set, then $x \leftarrow X$ simply means x is chosen uniformly at random from the set. We let $\Pi_{m,n} = (\{0,1\}^n)^{\{0,1\}^m}$ denote the set of functions mapping $\{0,1\}^m \to \{0,1\}^n$.

We refer the reader to [20] for a detailed introduction to quantum information. We let $\mathcal{S}(\mathcal{H})$ and $\mathcal{U}(\mathcal{H})$ denote the sets of unit vectors and unitary operators, respectively, on the Hilbert space $\mathcal{H}$. We let $\mathsf{Haar}(\mathbb{C}^d)$ denote the uniform probability measure on $\mathcal{S}(\mathbb{C}^d)$. We let $\mathsf{Tr}(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_1$ denote the trace distance between density matrices. For classical distributions, this is the total variation distance, $\mathsf{Tr}(p, q) = \frac{1}{2}\sum_x |p(x) - q(x)|$.

We follow the standard notation for quantum algorithms. We say that a quantum algorithm A is QPT if it is a uniform polynomial-time quantum algorithm.

The security parameter 1^λ is implicit in algorithm inputs when omitted. For an algorithm A with classical output and a family of inputs $x = x(\lambda)$, we say that $A(x)$ is *non-deterministic* if two independent evaluations yield distinct values with non-negligible probability in λ. At a fixed parameter, we say that $A(x)$ is *c-non-deterministic*, for $c \in (0, 1)$, if this probability is at least c.

3.2 Pseudorandom Primitives

We define pseudorandom states (PRSs), first introduced in [15].

Definition 1 (Pseudorandom State Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be a function in λ. A QPT algorithm PRS is called a* n-pseudorandom state generator (PRS) *if the following holds:*

- (Correctness) *On any input $k \in \{0,1\}^\lambda$, $\mathsf{PRS}(k)$ outputs an n-qubit state.*
- (Security) *For any polynomial $t(\cdot)$ and QPT distinguisher $\mathcal{A}$:*

$$\left| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[\mathcal{A}\left(\mathsf{PRS}(k)^{\otimes t(\lambda)} \right) = 1 \right] - \Pr_{|\phi\rangle \leftarrow \mathsf{Haar}(\mathbb{C}^{2^n})} \left[\mathcal{A}\left(|\phi\rangle^{\otimes t(\lambda)} \right) = 1 \right] \right| \leq \mathsf{negl}(\lambda).$$

We focus on two regimes of PRS*, based on the state size n:*

1. $n = \Theta(\log(\lambda))$, *which we call* short pseudorandom states (SPRSs).
2. $n = \Theta(\lambda)$, *which we call* long pseudorandom states (LPRSs).

We also define PRGs, allowing quantum evaluation with negligible correctness error averaged over uniform seeds.

Definition 2 (Pseudorandom Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be polynomial in λ. A QPT algorithm G mapping $\{0,1\}^\lambda$ to $\{0,1\}^n$ is called an* n-pseudorandom generator (PRG) *if*

- (Expansion) $n > \lambda$ *for all* $\lambda \in \mathbb{N}$.
- (Correctness) *There is a negligible bound such that, for each λ, there are strings $\{y_k \in \{0,1\}^n\}_{k\in\{0,1\}^\lambda}$ satisfying*

$$\Pr_{k\leftarrow\{0,1\}^\lambda}[G(k) = y_k] \geq 1 - \mathsf{negl}(\lambda). \tag{3}$$

- (Security) *For any QPT distinguisher $\mathcal{A}$:*

$$\left|\Pr_{k\leftarrow\{0,1\}^\lambda}[\mathcal{A}(G(k)) = 1] - \Pr_{y\leftarrow\{0,1\}^n}[\mathcal{A}(y) = 1]\right| \leq \mathsf{negl}(\lambda).$$

We now recall the definition of $\perp$-PRGs from [3]. First, we define the following useful operator.

Definition 3 (ls-$\perp$). *We define the operator*

$$\mathsf{ls}\text{-}\perp(a,b) := \begin{cases} \perp & \text{if } a = \perp \\ b & \text{otherwise.} \end{cases}$$

Definition 4 ($\perp$-Pseudorandom Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be polynomial in λ. A QPT algorithm G mapping $\{0,1\}^\lambda$ to $\{0,1\}^n \cup \{\perp\}$, is a* (μ, n)-$\perp$-pseudodeterministic pseudorandom generator ($\perp$-PRG) *if:*

1. (Expansion) $n(\lambda) > \lambda$ *for all* $\lambda \in \mathbb{N}$.
2. (Pseudodeterminism/Correctness) *There exists a constant $c > 0$ such that $\mu(\lambda) = O(\lambda^{-c})$ and for sufficiently large $\lambda \in \mathbb{N}$ there exists a set $\mathcal{G}_\lambda \subseteq \{0,1\}^\lambda$ satisfying the following conditions. The negligible bounds below are uniform over the indicated keys:*
   (a)
   $$\Pr_{k\leftarrow\{0,1\}^\lambda}[k \in \mathcal{G}_\lambda] \geq 1 - \mu(\lambda).$$
   (b) For every $k \in \mathcal{G}_\lambda$ there exists a non-$\perp$ value $y \in \{0,1\}^n$ such that:
   $$\Pr[G(k) = y] \geq 1 - \mathsf{negl}(\lambda). \tag{4}$$
   (c) For every $k \in \{0,1\}^\lambda$, there exists a non-$\perp$ value $y \in \{0,1\}^n$ such that:
   $$\Pr[G(k) \in \{y, \perp\}] \geq 1 - \mathsf{negl}(\lambda). \tag{5}$$

3. (Security) *For every polynomial $q = q(\lambda)$ and QPT distinguisher $\mathcal{A}$, there exists a negligible function ϵ such that:*

$$\left|\Pr\left[\begin{array}{l} k \leftarrow \{0,1\}^\lambda \\ y_1 \leftarrow G(k) \\ \quad\vdots \\ y_q \leftarrow G(k) \end{array} : \mathcal{A}(y_1,...,y_q)=1\right] - \Pr\left[\begin{array}{l} k \leftarrow \{0,1\}^\lambda \\ y \leftarrow \{0,1\}^n \\ y_1 \leftarrow \mathsf{ls}\text{-}\bot(G(k),y) \\ \quad\vdots \\ y_q \leftarrow \mathsf{ls}\text{-}\bot(G(k),y) \end{array} : \mathcal{A}(y_1,\ldots,y_q)=1\right]\right| \leq \epsilon(\lambda)$$

$\bot$-PRGs were constructed from SPRSs in [3].

Lemma 1 (Corollary 1 [3]). *If there exists $(c\log\lambda)$-SPRS for some constant $c > 12$, then there exists a $(O(\lambda^{-c/12+1}), \lambda^{c/12})$-$\bot$-PRG.*

3.3 Common Haar Function-like State Oracle

We first recall the definition of a swap unitary [10].

Definition 5. *For an n-qubit unit vector $|\phi\rangle$ orthogonal to $|0^n\rangle$, the swap (or reflection) unitary is defined by*

$$S_{|\phi\rangle} = |0^n\rangle\langle\phi| + |\phi\rangle\langle 0^n| + I_\bot$$

where we assume w.l.o.g. that $|\phi\rangle$ is orthogonal to $|0^n\rangle$, as we can always add $|1\rangle$ to make it orthogonal. $I_\bot$ is the identity on the subspace orthogonal to $\mathrm{span}\{|0^n\rangle, |\phi\rangle\}$.

We now define the (unitarized) *Common Haar Function-Like State (CHFS) oracle* as in [7].

Definition 6 (CHFS oracle). *Let $\ell = \ell(\lambda)$ be a positive integer-valued, efficiently computable, polynomially bounded function, also defined at $\lambda = 0$. We denote by O_ℓ the family of unitary oracles obtained from all choices of the vectors below. Writing $\mathcal{O} \leftarrow \mathsf{O}_\ell$ means sampling those vectors independently according to the Haar distribution:*

- ***Randomness:*** *Independently sample $|\phi_x\rangle \leftarrow \mathsf{Haar}(\mathbb{C}^{2^{\ell(|x|)}})$ for each $x \in \{0,1\}^*$ and define*
$$\Phi = \{|\phi_x\rangle|1\rangle\}_{x\in\{0,1\}^*}.$$
- ***Setup:*** *A family of oracles $\mathcal{O}^\Phi := (S_x^\Phi)_{x\in\{0,1\}^*} \leftarrow \mathsf{O}_\ell$ is chosen by randomly sampling Φ, where $S_x^\Phi := S_{|\phi_x\rangle|1\rangle}$ denotes the swap unitary as defined in Definition 5.*

– ***Query:*** *The oracle takes as a query a quantum state ρ_{XYZ} such that $|Y| = \ell(|X|) + 1$ and applies the unitary*

$$\begin{aligned}\mathcal{O}^{\Phi} &:= \sum_{x\in\{0,1\}^{|X|}} |x\rangle\langle x|_X \otimes S_x^{\Phi} \\ &= \sum_{x\in\{0,1\}^{|X|}} |x\rangle\langle x|_X \otimes S_{|\phi_x\rangle|1\rangle},\end{aligned}$$

on registers XY of ρ_{XYZ}, leaving Z unchanged, where S_x^{Φ} acts on Y. Here $|X|$ and $|Y|$ denote the numbers of qubits in these registers.

Remark 1. For simplicity, we often discard the superscript Φ in $\mathcal{O}^{\Phi}$. Furthermore, for a classical string $x \in \{0,1\}^*$, we write $\mathcal{O}(x)$ to denote the query $\mathcal{O}(|x\rangle|0^{\ell(|x|)+1}\rangle)$.

This query returns $|x\rangle|\phi_x\rangle|1\rangle$; discarding the label and flag leaves the state $|\phi_x\rangle$.

3.4 Black-Box Separation

Black-box separating primitives using oracles was first considered in [14] and later formalized in the quantum setting in [11]. We briefly define the relevant notions for this work.

Definition 7. *A* primitive *P is a pair $P = (\mathcal{F}_P, \mathcal{R}_P)$* [4] *where $\mathcal{F}_P$ is a set of quantum channels, and $\mathcal{R}_P$ is a relation over pairs $(G, \mathcal{A})$ of quantum channels, where $G \in \mathcal{F}_P$.*

A quantum channel G is an implementation *of P if $G \in \mathcal{F}_P$. If G is additionally a QPT channel, then we say that G is an* efficient implementation *of P. A quantum channel $\mathcal{A}$ P-*breaks *$G \in \mathcal{F}_P$ if $(G, \mathcal{A}) \in \mathcal{R}_P$. We say that G is a* secure implementation *of P if G is an implementation of P such that no QPT channel P-breaks it. The primitive P* exists *if there exists an efficient and secure implementation of P.*

A unitary U implements P if it implements some channel in $\mathcal{F}_P$. We say that $\mathcal{A}$ P-breaks U when it breaks that channel.

We now formalize the notion of constructions relative to an oracle.

Definition 8. *We say that a primitive P exists relative to an oracle $\mathcal{O}$ if:*

– *There exists a QPT oracle algorithm $G^{(\cdot)}$ such that $G^{\mathcal{O}} \in \mathcal{F}_P$.*
– *The security of $G^{\mathcal{O}}$ holds against all QPT adversaries with access to $\mathcal{O}$, i.e., for every QPT oracle algorithm $\mathcal{A}^{(\cdot)}$, $(G^{\mathcal{O}}, \mathcal{A}^{\mathcal{O}}) \notin \mathcal{R}_P$.*

[4] We can think of $\mathcal{F}_P$ to mean the "correctness" conditions of P and $\mathcal{R}_P$ to mean the "security" conditions of P.

We are now ready to define the notion of fully black-box construction.

The correctness requirement below applies to every implementation of P, including inefficient and insecure ones. Its negligible correctness bound may depend on the fixed implementation U; one bound for all U is not required.

Definition 9. *A QPT algorithm $G^{(\cdot)}$ is a fully black-box construction (BB) of Q from P **with inverse access** if the following two conditions hold:*

1. *For every unitary implementation U of P, $G^{U,U^{-1}} \in \mathcal{F}_Q$.*
2. *There is a QPT algorithm $S^{(\cdot)}$ such that, for every unitary implementation U of P, every adversary $\mathcal{A}$ that Q-breaks $G^{U,U^{-1}}$, and every unitary implementation $\tilde{\mathcal{A}}$ of $\mathcal{A}$, it holds that $S^{\tilde{\mathcal{A}},\tilde{\mathcal{A}}^{-1}}$ P-breaks U.*

The following result from [11] shows the relation between BB constructions and oracle separations.

Theorem 3 (Theorem 4.2 in [11]). *Assume there exists a BB construction of a primitive Q from a primitive P with inverse access. Then, for any unitary $\mathcal{O}$, if P exists relative to $(\mathcal{O}, \mathcal{O}^{-1})$, then Q exists relative to $(\mathcal{O}, \mathcal{O}^{-1})$.*

4 Separation

In this section, we present our separation of PRG from PRS.

Theorem 4. *Let $\ell = \ell(\lambda) \in O(\lambda)$ be a positive integer-valued, efficiently computable function of the security parameter $\lambda \in \mathbb{N}$ and let $n = n(\lambda)$ be an integer-valued polynomial satisfying $n > \lambda$. There does not exist a BB construction of an n-PRG from an ℓ-PRS with inverse access.*

Proof. Our approach is to apply any proposed BB construction to a PRS based on a unitary oracle. We show that the construction's correctness and security requirements cannot both hold.

Our proof is relative to two unitary oracles: a fixed (unitarized) oracle $\mathcal{C}$ for a PSPACE-complete language and an ℓ-CHFS oracle $\mathcal{O}$ sampled from a set of oracles O_ℓ. The oracle $\mathcal{C}$ is independent of $\mathcal{O}$. Given that both oracles are self-inverse, it is sufficient to give access to $\mathcal{T} \coloneqq (\mathcal{C}, \mathcal{O})$.

Assume for contradiction that there exists a BB construction of an n-PRG from an ℓ-PRS with inverse access. First, we state the following result which follows directly from Theorem 4.4 in [7].

Claim 1. *There exists an ℓ-PRS relative to $\mathcal{T}$. Its unitary implementation and inverse only use oracle access to $\mathcal{O}$. It satisfies security with probability 1 over the distribution of $\mathcal{O}$ and correctness for all $\mathcal{O} \in \mathsf{O}_\ell$.*

Proof. On input $k \in \{0,1\}^\lambda$, query $\mathcal{O}$ on $|k\rangle|0^{\ell(\lambda)+1}\rangle$ to obtain $|k\rangle|\phi_k\rangle|1\rangle$, and output $|\phi_k\rangle$. The unitary implementation and its inverse each use one query to $\mathcal{O}$. Correctness holds for every choice of the oracle states. Security follows from Theorem 4.4 in [7], with key length λ and no function input, including access to $\mathcal{C}$. □

Applying the assumed BB construction to the PRS in Claim 1 gives a QPT generator $\overline{G} : \{0,1\}^\lambda \to \{0,1\}^n$ that only queries $\mathcal{O}$. By the BB correctness requirement, $\overline{G}^{\mathcal{O}}$ satisfies correctness for every $\mathcal{O} \in \mathsf{O}_\ell$, with a negligible bound that may depend on $\mathcal{O}$. By the security reduction, it is secure relative to $\mathcal{T}$ whenever the underlying PRS is secure, and hence with probability 1 over the distribution of $\mathcal{O}$. Assume $\overline{G}$ queries the oracle at most $T = T(\lambda) \geq 1$ times for some polynomial T, and define

$$m = m(\lambda) \coloneqq \lceil 4\pi T(\lambda)\lambda \rceil + 1, \qquad \delta(\lambda) \coloneqq \frac{1}{100\lambda m(\lambda)}.$$

We first show that fixing finitely many oracle entries gives a correctness bound that holds for every completion of those entries. For simplicity, a *prefix* fixes all states defining the oracle at lengths $d < a$ for some constant a, and a *completion* is any oracle in the family that agrees with these fixed states.

Claim 2. *There exist constants a, λ_0 and fixed states $\{|\phi^*_x\rangle\}_{|x|<a}$, each obtained by normalizing a nonzero vector whose coordinates have rational real and imaginary parts, such that every $\mathcal{O} \in \mathsf{O}_\ell$ with these entries satisfies, for every $\lambda \geq \lambda_0$,*

$$e_{\mathcal{O}}(\lambda) \coloneqq \mathbb{E}_{k \leftarrow \{0,1\}^\lambda}\left[1 - \max_y \Pr[\overline{G}^{\mathcal{O}}(k) = y]\right] \leq \delta(\lambda). \tag{6}$$

Proof. For every fixed $\mathcal{O} \in \mathsf{O}_\ell$, correctness implies that $e_{\mathcal{O}}$ is negligible. For each $N \in \mathbb{N}$, let

$$F_N \coloneqq \{\mathcal{O} \in \mathsf{O}_\ell : e_{\mathcal{O}}(\lambda) \leq \delta(\lambda) \text{ for every } \lambda \geq N\}.$$

Let $\mathsf{O}'_\ell \subseteq \mathsf{O}_\ell$ consist of the oracles whose states, at every label, are obtained by normalizing nonzero vectors with rational real and imaginary coordinates.

First, we show that there exist a, N and a prefix of this form such that every completion of the prefix in O'_ℓ belongs to F_N. Suppose otherwise. Starting with the empty prefix and $a_0 = 0$, we choose strictly increasing parameters $\lambda_1, \lambda_2, \ldots$ and fix longer prefixes as follows. At step j, choose a completion $\mathcal{O}^j \in \mathsf{O}'_\ell$ of the already fixed prefix and a parameter $\lambda_j \geq j$, larger than all previous parameters, such that

$$e_{\mathcal{O}^j}(\lambda_j) > \delta(\lambda_j).$$

Such a choice exists by our supposition. Choose $a_j > \max\{a_{j-1}, j\}$ larger than every query length that can be submitted by $\overline{G}$ when run on seed of length λ_j. This is possible because $\overline{G}$ runs in polynomial time. Fix all entries with $|x| < a_j$ to their values in $\mathcal{O}^j$. No previously fixed entry changes.

Since $a_j > j$, the successive prefixes define one oracle $\mathcal{O}^* \in \mathsf{O}'_\ell \subseteq \mathsf{O}_\ell$, and for every j,

$$e_{\mathcal{O}^*}(\lambda_j) = e_{\mathcal{O}^j}(\lambda_j) > \delta(\lambda_j).$$

This contradicts correctness for $\mathcal{O}^*$, since $e_{\mathcal{O}^*}$ is negligible and δ is inverse polynomial. Thus there exist a, N and a fixed prefix of the required form whose every completion in O'_ℓ belongs to F_N.

We now allow arbitrary states outside this prefix. Fix any completion $\mathcal{O} \in \mathsf{O}_\ell$ and any $\lambda \geq N$. For each integer $r \geq 1$, approximate the state vector at every label that $\overline{G}$ can query at this parameter to Euclidean error at most $1/r$, using normalized vectors with rational real and imaginary coordinates. Such approximations exist by approximating each coordinate by rational numbers and then normalizing. Keep the fixed prefix unchanged, and set all remaining unfixed entries to $|0^{\ell(|x|)}\rangle$. This gives a completion $\mathcal{O}^{(r)} \in \mathsf{O}'_\ell$ with $e_{\mathcal{O}^{(r)}}(\lambda) \leq \delta(\lambda)$.

The swap definition gives

$$\left\| S_{|\phi\rangle|1\rangle} - S_{|\psi\rangle|1\rangle} \right\|_{\mathrm{op}} \leq 4 \left\| |\phi\rangle - |\psi\rangle \right\|_2 .$$

Since the oracle acts separately on each label, the same bound applies to superposition queries, including queries entangled with other registers. Replacing the at most $T(\lambda)$ queries one at a time changes the probability of any output event by at most $4T(\lambda)/r$. This also covers inverse queries, since the swap oracles are self-inverse. Therefore,

$$e_{\mathcal{O}}(\lambda) \leq e_{\mathcal{O}^{(r)}}(\lambda) + \frac{4T(\lambda)}{r} \leq \delta(\lambda) + \frac{4T(\lambda)}{r}.$$

Letting r tend to infinity gives $e_{\mathcal{O}}(\lambda) \leq \delta(\lambda)$. This holds for every completion $\mathcal{O} \in \mathsf{O}_\ell$ of the prefix and every $\lambda \geq N$. Taking $\lambda_0 = N$ proves the claim. □

From now on, O_ℓ denotes the family of completions of this fixed prefix, and $\mathcal{O} \leftarrow \mathsf{O}_\ell$ means that the remaining entries are sampled independently from the Haar distribution. The prefix may depend on $\overline{G}$.

The rational descriptions of the fixed states can be hardwired into an algorithm, since a is constant and states have rational amplitudes. To simulate their swaps, we use the observation that a swap is a reflection [13, Section 5]. Namely, for each $|x| < a$, let

$$|v_x\rangle \coloneqq \frac{\left|0^{\ell(|x|)+1}\right\rangle - |\phi_x^*\rangle|1\rangle}{\sqrt{2}}.$$

Then

$$S_{|\phi_x^*\rangle|1\rangle} = I - 2|v_x\rangle\langle v_x|.$$

The known coordinates determine a unitary V_x satisfying $V_x|0^{\ell(|x|)+1}\rangle = |v_x\rangle$. Since its dimension is constant, the constructive Solovay–Kitaev algorithm [12, Theorem 1 and Sections 3 and 5] allows us to compute a circuit $\widetilde{V}_x$ approximating V_x to operator-norm error $\eta/2$, in time polynomial in $\log(1/\eta)$.

Claim 1 still holds under this distribution. Given a QPT adversary for the restricted family, simulate each query on a label $|x| < a$ using the fixed prefix to error $2^{-\lambda}$, and use the original oracle on all other labels. Each query has a fixed input length, so this replaces the entire query when that length is less than a and leaves it unchanged otherwise. The simulation is coherent, and inverse queries are handled by reversing the simulation circuit. Access to $\mathcal{C}$ is unchanged. A polynomial number of queries gives negligible total error.

Define the oracle I to be the element of O_ℓ whose state is $|\phi^*_x\rangle$ for $|x| < a$ and $|0^{\ell(|x|)}\rangle$ otherwise. Queries to I can be approximated to negligible error η in polynomial time without any oracle access.

Lemma 2. *For every $\mathcal{O} \in \mathsf{O}_\ell$ and sufficiently large λ, all but a $\frac{1}{10\lambda}$ fraction of inputs $k \in \{0,1\}^\lambda$ have a string y_k such that*

$$\Pr[\overline{G}^{\mathcal{O}}(k) = y_k] \geq \frac{9}{10}, \tag{7}$$

$$\Pr[\overline{G}^{I}(k) = y_k] \geq \frac{9}{10}. \tag{8}$$

Proof. Fix $\mathcal{O} \in \mathsf{O}_\ell$ and $\lambda \geq \max\{\lambda_0, 2\}$. We define a sequence of m oracles $\mathcal{O}_1, \ldots, \mathcal{O}_m$ starting with $\mathcal{O}_1 = \mathcal{O}$ and ending with $\mathcal{O}_m = I$. For every label x, gradually move its state vector along a shortest path on the unit sphere from its value in $\mathcal{O}$ to its value in I. We use the full vectors, including their phases, and keep the entries with $|x| < a$ unchanged. Each path has length at most π. Splitting it into $m-1$ equal steps gives consecutive vectors at Euclidean distance at most $\pi/(m-1)$.

For any two unit vectors $|\phi\rangle, |\psi\rangle$, the corresponding swaps satisfy

$$\left\| S_{|\phi\rangle|1\rangle} - S_{|\psi\rangle|1\rangle} \right\|_{\mathrm{op}} \leq 4 \left\| |\phi\rangle - |\psi\rangle \right\|_2 .$$

This follows directly from the swap definition in Definition 5. Since the oracle acts separately on each label, it follows that

$$\|\mathcal{O}_i - \mathcal{O}_{i+1}\|_{\mathrm{op}} \leq \frac{4\pi}{m-1}.$$

Replacing the at most T queries one at a time therefore gives, for every $i \in [m-1]$ and every input k,

$$\mathsf{Tr}\left(\overline{G}^{\mathcal{O}_i}(k), \overline{G}^{\mathcal{O}_{i+1}}(k)\right) \leq \frac{4\pi T}{m-1} \leq \frac{1}{\lambda}. \tag{9}$$

The same bound holds for inverse queries, since the swap oracles are self-inverse.

Every $\mathcal{O}_i$ belongs to O_ℓ, so Eq. (6) applies to it. By Markov's inequality, at most a $10\delta(\lambda)$ fraction of inputs have no output occurring with probability at least $9/10$ under $\overline{G}^{\mathcal{O}_i}$. Taking a union bound over the m oracles, all but a

$$10m\delta(\lambda) = \frac{1}{10\lambda}$$

fraction of inputs have such an output at every intermediate oracle.

Fix any remaining input k. Its most likely output cannot change between consecutive oracles: two distinct outputs, each occurring with probability at least $9/10$ in its respective distribution, would give trace distance at least $4/5$, contradicting Eq. (9). Hence the same string y_k satisfies Eqs. (7) and (8). □

We now give a search attack against $\overline{G}$ using the PSPACE oracle $\mathcal{C}$. Since I can be efficiently approximated, $\overline{G}^I$ can be simulated without any oracle access. A PSPACE computation can therefore estimate $\Pr[\overline{G}^I(k) = y]$ for every seed k.

On input y, let $\mathcal{A}^{\mathcal{C}}$ output 1 if some $k \in \{0,1\}^\lambda$ has an estimated probability at least $2/3$, and output 0 otherwise. The search over all seeds uses polynomial space, so $\mathcal{A}$ can perform it with a query to $\mathcal{C}$.

For each seed, at most one string can pass this test. Thus at most 2^λ strings are accepted, and

$$\Pr_{y \leftarrow \{0,1\}^n}[\mathcal{A}^{\mathcal{C}}(y) = 1] \leq 2^{\lambda - n} \leq \frac{1}{2}. \tag{10}$$

On the other hand, fix any $\mathcal{O} \in \mathsf{O}_\ell$. By Lemma 2, all but a $\frac{1}{10\lambda}$ fraction of seeds have the same most likely output y_k under $\overline{G}^{\mathcal{O}}$ and $\overline{G}^I$, with probability at least $9/10$ in both cases. This string passes the test. By Eq. (6), the average probability that $\overline{G}^{\mathcal{O}}$ fails to output its most likely string is at most $\delta(\lambda)$. Therefore,

$$\Pr_{k \leftarrow \{0,1\}^\lambda}[\mathcal{A}^{\mathcal{C}}(\overline{G}^{\mathcal{O}}(k)) = 1] \geq 1 - \frac{1}{10\lambda} - \delta(\lambda). \tag{11}$$

Combining Eqs. (10) and (11), the distinguishing advantage is at least

$$\frac{1}{2} - \frac{1}{10\lambda} - \delta(\lambda), \tag{12}$$

which is bounded below by a positive constant for sufficiently large λ. This attack works for every $\mathcal{O} \in \mathsf{O}_\ell$, contradicting the security of $\overline{G}$ whenever the underlying PRS is secure.

Therefore, there does not exist a BB construction of an n-PRG from an ℓ-PRS with inverse access. □

5 Separating PRG from ⊥-PRG

To demonstrate the separation between PRGs and ⊥-PRGs, we will use two independent unitary oracles: a fixed (unitarized) oracle $\mathcal{C}$ for a PSPACE-complete language and a ⊥-pseudodeterministic random oracle $\mathcal{O}$ described in Construction 1, similar to one used in [3].

Construction 1. *Let $\lambda \in \mathbb{N}$ be the security parameter, let $c > 0$ be a constant, and set $\mu(\lambda) = \lambda^{-c}$. Let ℓ be an integer-valued polynomial with $\ell(\lambda) > \lambda$.*

Sample $O_\lambda \leftarrow \Pi_{\lambda,\ell}$. Independently, sample a function $Q_\lambda : \{0,1\}^\lambda \to [0,1]$ by choosing each value uniformly from

$$\left\{ \frac{j}{2^\lambda} : j = 0, \ldots, 2^\lambda - 1 \right\}.$$

Sample a uniformly random set $\mathcal{P}_\lambda \subseteq \{0,1\}^\lambda$ of size $\lfloor \mu(\lambda) 2^\lambda \rfloor$, and let P_λ be its indicator function.

For each $x \in \{0,1\}^\lambda$, define

$$p_x \coloneqq \sin^2\left(\frac{\pi}{2} Q_\lambda(x)\right), \qquad |\phi_x\rangle \coloneqq \begin{cases} \sqrt{p_x}|\perp\rangle + \sqrt{1-p_x}|O_\lambda(x)\rangle, & P_\lambda(x) = 1, \\ |O_\lambda(x)\rangle, & P_\lambda(x) = 0. \end{cases}$$

Define the unitary oracle

$$\mathcal{O}_\lambda \coloneqq \mathcal{O}[P_\lambda, Q_\lambda, O_\lambda] \coloneqq \sum_{x \in \{0,1\}^\lambda} |x\rangle\langle x| \otimes S_{|\phi_x\rangle},$$

where $|0\rangle$, $|\perp\rangle$, and the encoded output strings $|y\rangle$ are distinct computational-basis states, so that the initialization state $|0\rangle$ is orthogonal to every possible output, including the encoded string $|0^{\ell(\lambda)}\rangle$,

$$S_{|\phi_x\rangle} \coloneqq |0\rangle\langle\phi_x| + |\phi_x\rangle\langle 0| + I_\perp,$$

and $I_\perp$ is the identity on the subspace orthogonal to $\mathrm{span}\{|0\rangle, |\phi_x\rangle\}$*. Thus,*

$$\mathcal{O}_\lambda\big(|x\rangle|0\rangle\big) = |x\rangle|\phi_x\rangle, \qquad \mathcal{O}_\lambda^{-1} = \mathcal{O}_\lambda.$$

We define the "good" set $\mathcal{G}_\lambda^{\mathcal{O}}$ for $\mathcal{O}_\lambda$ as follows:

$$\mathcal{G}_\lambda^{\mathcal{O}} \coloneqq \{x \in \{0,1\}^\lambda : P_\lambda(x) = 0\}.$$

We let $\mathcal{O}_{\mu,\ell}$ denote the family of oracles $(\mathcal{O}_\lambda)_{\lambda \in \mathbb{N}}$ defined above, allowing arbitrary functions O_λ, arbitrary functions $Q_\lambda : \{0,1\}^\lambda \to [0,1]$, and arbitrary indicator functions P_λ satisfying

$$\big|\{x : P_\lambda(x) = 1\}\big| \leq \lfloor \mu(\lambda) 2^\lambda \rfloor.$$

Sampling $\mathcal{O} \leftarrow \mathcal{O}_{\mu,\ell}$ means using the distribution in Construction 1, independently at each input length.

When we write $\mathcal{O}_\lambda(x)$ as a classical output, we mean the outcome obtained by applying the oracle to $|x\rangle|0\rangle$ and measuring the output register. The measurement is performed by the calling algorithm.

The following lemma follows directly from the definition of $\mathcal{O}$.

Lemma 3. *$\mathcal{O}_\lambda$ has the following properties:*

- $\Pr_{x \leftarrow \{0,1\}^\lambda}\left[x \in \mathcal{G}_\lambda^{\mathcal{O}}\right] \geq 1 - \mu$.
- *For every $x \in \mathcal{G}_\lambda^{\mathcal{O}}$, there exists a non-$\perp$ value $y \in \{0,1\}^\ell$ such that:*

$$\Pr\left[\mathcal{O}_\lambda(x) = y\right] = 1.$$

- *For every $x \notin \mathcal{G}_\lambda^{\mathcal{O}}$, there exists a probability $p_x \in [0,1]$ and non-$\perp$ value $y \in \{0,1\}^\ell$ such that:*
  1. $\Pr\left[\mathcal{O}_\lambda(x) = y\right] = 1 - p_x$.

2. $\Pr[\mathcal{O}_\lambda(x) = \bot] = p_x$.

Theorem 5. *Let $\lambda \in \mathbb{N}$ be the security parameter. For any integer-valued polynomials $w(\lambda), \ell(\lambda) > \lambda$ and pseudodeterminism error $\mu(\lambda) = \lambda^{-c}$ for $c > 0$, there does not exist a BB construction of a w-PRG from a $(3\mu, \ell)$-$\bot$-PRG with inverse access.*

Proof. Our approach is to apply any proposed BB construction to a $\bot$-PRG based on the self-inverse unitary oracles $\mathcal{T} := (\mathcal{C}, \mathcal{O})$. We show that the construction's correctness and security requirements cannot both hold.

Assume for contradiction that there exists a BB construction of a w-PRG $\overline{G}^F$ from a $(3\mu, \ell)$-$\bot$-PRG F, with inverse access. First, we show that there exists a (μ, ℓ)-$\bot$-PRG relative to the oracles $(\mathcal{O}, \mathcal{C})$.

Claim 3. *For every fixed admissible pair of sequences P, Q chosen independently of O, with probability 1 over O, the generator that evaluates $\mathcal{O}_\lambda = \mathcal{O}[P_\lambda, Q_\lambda, O_\lambda]$ and measures its output is a (μ, ℓ)-$\bot$-PRG relative to $(\mathcal{O}, \mathcal{C})$. Furthermore, correctness is satisfied for every $\mathcal{O} \in \mathcal{O}_{\mu,\ell}$.*

Proof. By Lemma 3, $\mathcal{O}$ satisfies the correctness/pseudodeterminism condition of a (μ, ℓ)-$\bot$-PRG.

For security, we need to show that for any P, Q and with probability 1 over the distribution of O: for every non-uniform QPT distinguisher $\mathcal{A}$ and polynomial $q = q(\lambda)$:

$$\left| \Pr\left[\begin{matrix} k \leftarrow \{0,1\}^\lambda \\ y_1 \leftarrow \mathcal{O}_\lambda(k) \\ \vdots \\ y_q \leftarrow \mathcal{O}_\lambda(k) \end{matrix} : \mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1, ..., y_q) = 1 \right] - \Pr\left[\begin{matrix} k \leftarrow \{0,1\}^\lambda \\ y \leftarrow \{0,1\}^\ell \\ y_1 \leftarrow \mathsf{Is}\text{-}\bot(\mathcal{O}_\lambda(k), y) \\ \vdots \\ y_q \leftarrow \mathsf{Is}\text{-}\bot(\mathcal{O}_\lambda(k), y) \end{matrix} : \mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1, \ldots, y_q) = 1 \right] \right| \leq \mathsf{negl}(\lambda)$$

Let Z_λ be the function that outputs 0^ℓ on any input and let $\mathcal{Z}_\lambda := \mathcal{O}[P_\lambda, Q_\lambda, Z_\lambda]$. Note that

- $\mathcal{Z}_\lambda$ is independent of O_λ,
- $\mathcal{O}_\lambda(k) = \mathsf{Is}\text{-}\bot(\mathcal{O}_\lambda(k), O_\lambda(k)) = \mathsf{Is}\text{-}\bot(\mathcal{Z}_\lambda(k), O_\lambda(k))$,
- $\mathsf{Is}\text{-}\bot(\mathcal{O}_\lambda(k), y) = \mathsf{Is}\text{-}\bot(\mathcal{Z}_\lambda(k), y)$.

Therefore, $\mathcal{A}^{\mathcal{O},\mathcal{C}}$ needs to distinguish between evaluations of $\mathsf{Is}\text{-}\bot(\mathcal{Z}_\lambda(k), y)$ and $\mathsf{Is}\text{-}\bot(\mathcal{Z}_\lambda(k), O_\lambda(k))$.

Lemma 2.2 from [21] states that a random oracle acts as a PRG i.e.:

$$\mathop{\mathbb{E}}_{O \leftarrow \Pi_{\lambda,\ell}} \left[\left| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[\mathcal{A}^O(O(k)) = 1 \right] - \Pr_{y \leftarrow \{0,1\}^\ell} \left[\mathcal{A}^O(y) = 1 \right] \right| \right] \leq \frac{1}{2^{\lambda/4}}.$$

Note that this result even holds against unbounded-time adversaries as long as the number of queries to the oracle is polynomial. Hence, this result also holds against adversaries with access to a PSPACE-oracle:

$$\mathop{\mathbb{E}}_{O \leftarrow \Pi_{\lambda,\ell}} \left[\left| \Pr_{k \leftarrow \{0,1\}^\lambda} \left[\mathcal{A}^{O,\mathcal{C}}(O(k)) = 1 \right] - \Pr_{y \leftarrow \{0,1\}^\ell} \left[\mathcal{A}^{O,\mathcal{C}}(y) = 1 \right] \right| \right] \leq \frac{1}{2^{\lambda/4}}.$$

Next, notice that for any functions P, Q, distinguishing between evaluations of $\mathsf{ls}\text{-}\bot(\mathcal{Z}_\lambda(k), y)$ and $\mathsf{ls}\text{-}\bot(\mathcal{Z}_\lambda(k), O_\lambda(k))$ is just as hard as distinguishing the two scenarios in the equation above, given that $\mathcal{Z}_\lambda$ is independent of O_λ. Therefore,

$$\mathop{\mathbb{E}}_{O\leftarrow\Pi_{\lambda,\ell}}\left[\left|\Pr_{(y_1,\ldots,y_q)\leftarrow D^0_{\mathcal{Z},O}}\left[\mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1,\ldots,y_q)=1\right]-\Pr_{(y_1,\ldots,y_q)\leftarrow D^1_{\mathcal{Z}}}\left[\mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1,\ldots,y_q)=1\right]\right|\right]\leq 2^{-\lambda/4}$$

where,

$$D^0_{\mathcal{Z},O}\coloneqq\begin{bmatrix}k\leftarrow\{0,1\}^\lambda\\ y_1\leftarrow\mathsf{ls}\text{-}\bot(\mathcal{Z}_\lambda(k),O_\lambda(k))\\ \vdots\\ y_q\leftarrow\mathsf{ls}\text{-}\bot(\mathcal{Z}_\lambda(k),O_\lambda(k))\end{bmatrix}\quad D^1_{\mathcal{Z}}\coloneqq\begin{bmatrix}k\leftarrow\{0,1\}^\lambda\\ y\leftarrow\{0,1\}^\ell\\ y_1\leftarrow\mathsf{ls}\text{-}\bot(\mathcal{Z}_\lambda(k),y)\\ \vdots\\ y_q\leftarrow\mathsf{ls}\text{-}\bot(\mathcal{Z}_\lambda(k),y)\end{bmatrix}$$

By Markov inequality, we get that

$$\Pr_{O\leftarrow\Pi_{\lambda,\ell}}\left[\left|\Pr_{(y_1,\ldots,y_q)\leftarrow D^0_{\mathcal{Z},O}}\left[\mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1,\ldots,y_q)=1\right]-\Pr_{(y_1,\ldots,y_q)\leftarrow D^1_{\mathcal{Z}}}\left[\mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1,\ldots,y_q)=1\right]\right|\geq 2^{-\lambda/8}\right]\leq 2^{-\lambda/8}$$

By Borel-Cantelli Lemma, since $\sum_\lambda 2^{-\lambda/8}$ converges, with probability 1 over the distribution of O, it holds that

$$\left|\Pr_{(y_1,\ldots,y_q)\leftarrow D^0_{\mathcal{Z},O}}\left[\mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1,\ldots,y_q)=1\right]-\Pr_{(y_1,\ldots,y_q)\leftarrow D^1_{\mathcal{Z}}}\left[\mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1,\ldots,y_q)=1\right]\right|\leq 2^{-\lambda/8},$$

except for finitely many $\lambda\in\mathbb{N}$. There are countably many quantum algorithms $\mathcal{A}$ making polynomial queries to $(\mathcal{O},\mathcal{C})$, so this bound holds for every such adversary. Therefore, $\mathcal{O}$ is a $\bot$-PRG for any P, Q and with probability 1 over the distribution of O. □

We instantiate F by the generator above. Its unitary implementation and inverse only query $\mathcal{O}$, so the resulting generator $\overline{G}^{\mathcal{O}}$ only queries $\mathcal{O}$. By the assumed BB construction, it satisfies correctness for every $\mathcal{O}\in\mathcal{O}_{3\mu,\ell}$. Indeed, Lemma 3, with μ replaced by 3μ, gives source correctness throughout this larger family.

Assume $\overline{G}$ makes at most T queries, on inputs of length at most T, where $T=T(\lambda)\geq\lambda$ is polynomial. Define

$$m=m(\lambda)\coloneqq\lceil 2\pi T(\lambda)\lambda\rceil,\qquad n=n(\lambda)\coloneqq\left\lceil\frac{24m(\lambda)}{\mu(T(\lambda))}\right\rceil,$$
$$\delta(\lambda)\coloneqq\frac{1}{100\lambda n(\lambda)}.$$

Here n bounds the number of intermediate oracles used below.

Claim 4. *There exist constants a, λ_0 and fixed tables $(P_d^*, Q_d^*, O_d^*)_{d<a}$, with rational entries in Q_d^*, such that every $\mathcal{O} \in \mathcal{O}_{3\mu,\ell}$ with these functions, for every $\lambda \geq \lambda_0$,*

$$e_{\mathcal{O}}(\lambda) \coloneqq \mathbb{E}_{k \leftarrow \{0,1\}^\lambda} \left[1 - \max_y \Pr[\overline{G}^{\mathcal{O}}(k) = y] \right] \leq \delta(\lambda). \tag{13}$$

Proof. We use the same argument as in Claim 2. First restrict to oracles whose Q-entries are rational. Suppose that no finite prefix with rational Q-entries has a correctness bound $e_{\mathcal{O}}(\lambda) \leq \delta(\lambda)$ for all sufficiently large λ and all such completions.

Starting with the empty prefix, choose strictly increasing parameters $\lambda_j \geq j$ and a completion $\mathcal{O}^j$ of the previously fixed tables such that

$$e_{\mathcal{O}^j}(\lambda_j) > \delta(\lambda_j).$$

Such a choice exists by our supposition. Choose $a_j > \max\{a_{j-1}, j, T(\lambda_j)\}$, with $a_0 = 0$, and fix all tables at lengths $d < a_j$ to those of $\mathcal{O}^j$. Every later completion has the same output distribution at λ_j, since all possible query lengths have been fixed. The successive prefixes therefore define one admissible oracle $\mathcal{O}^*$ with rational Q-entries satisfying

$$e_{\mathcal{O}^*}(\lambda_j) > \delta(\lambda_j) \qquad \text{for every } j.$$

This contradicts correctness for $\mathcal{O}^*$, since its error is negligible and δ is inverse polynomial. Thus there exist a, N and a fixed prefix such that every completion with rational Q-entries satisfies the bound for all $\lambda \geq N$.

Now fix any completion $\mathcal{O} \in \mathcal{O}_{3\mu,\ell}$ of this prefix and any $\lambda \geq N$. For each integer $r \geq 1$, replace every Q-entry outside the prefix by a rational value in $[0, 1]$ within $1/r$, leaving P, O and the prefix unchanged. Call the resulting oracle $\mathcal{O}^{(r)}$. Its correctness error at λ is at most $\delta(\lambda)$.

By Construction 1, each prepared state changes by at most $\pi/(2r)$ in Euclidean distance, and its swap changes by at most $2\pi/r$ in operator norm. This bound also holds for superposition queries. Replacing the at most $T(\lambda)$ queries one at a time, including inverse queries, gives

$$e_{\mathcal{O}}(\lambda) \leq e_{\mathcal{O}^{(r)}}(\lambda) + \frac{2\pi T(\lambda)}{r} \leq \delta(\lambda) + \frac{2\pi T(\lambda)}{r}.$$

Letting r tend to infinity gives the required bound. Taking $\lambda_0 = N$ proves the claim. □

Define for any $d \geq a$,

$$b_d \coloneqq \lfloor \mu(d) 2^d \rfloor.$$

We now keep this prefix fixed and sample all remaining entries as in Construction 1. The security conclusion of Claim 3 is unchanged: the finite rational prefix can be hardcoded and simulated to negligible error, while the remaining entries

retain their original distribution. The resulting oracle is a $(3\mu, \ell)$-$\perp$-PRG, since the fixed prefix satisfies the 3μ budget and all longer lengths satisfy the μ budget.

Define $\mathcal{Z}$ to retain the fixed prefix and, at every length $d \geq a$, to use $P_d = Q_d = 0$ and $O_d(x) = 0^{\ell(d)}$. Thus $\mathcal{Z} \in \mathcal{O}_{3\mu,\ell}$ can be approximated efficiently.

Lemma 4. *For every oracle $\mathcal{O}$ with the fixed prefix and the μ budget at all lengths $d \geq a$, and for sufficiently large λ, all but a $\frac{1}{10\lambda}$ fraction of inputs $k \in \{0,1\}^\lambda$ have a string y_k such that*

$$\Pr[\overline{G}^{\mathcal{O}}(k) = y_k] \geq \frac{9}{10}, \tag{14}$$

$$\Pr[\overline{G}^{\mathcal{Z}}(k) = y_k] \geq \frac{9}{10}. \tag{15}$$

Proof. Fix $\mathcal{O}$ and take $\lambda \geq \max\{a, \lambda_0, 2\}$. We first construct a sequence of at most n oracles from $\mathcal{O}$ to $\mathcal{Z}$. We change the classical outputs by first moving them to $\perp$.

For each $q \in [\lceil 2/\mu(T) \rceil]$, define

$$S_{T,q} \coloneqq \{x \in \{0,1\}^d : \ a \leq d \leq T, \ (q-1)b_d \leq x < qb_d\}, \tag{16}$$

where x is interpreted as an integer. These sets partition all inputs with lengths between a and T: at each length d, the number of blocks is at most $\lceil 2/\mu(d) \rceil \leq \lceil 2/\mu(T) \rceil$.

Starting from the tables (P, Q, O) of $\mathcal{O}$, perform the following steps for each block $S_{T,q}$. Each change applies to every x in that block and leaves all other entries unchanged.

1. Repeatedly set $Q(x) \leftarrow \max(Q(x) - 1/m, 0)$ until $Q(x) = 0$ throughout the block.
2. Set $P(x) \leftarrow 1$ throughout the block.
3. Repeatedly set $Q(x) \leftarrow \min(Q(x) + 1/m, 1)$ until $Q(x) = 1$ throughout the block.
4. Set $O(x) \leftarrow 0^{\ell(|x|)}$ throughout the block.
5. Repeatedly set $Q(x) \leftarrow \max(Q(x) - 1/m, 0)$ until $Q(x) = 0$ throughout the block.
6. Set $P(x) \leftarrow 0$ throughout the block.

Record the oracle after each change. Finally, replace entries at lengths greater than T by those of $\mathcal{Z}$; $\overline{G}$ never queries these entries.

At each length $d \geq a$, the support of P is contained in its original support together with the current block. Hence it has size at most $2b_d \leq \lfloor 3\mu(d)2^d \rfloor$. The prefix is unchanged. Therefore, every intermediate oracle belongs to $\mathcal{O}_{3\mu,\ell}$ and satisfies Eq. (13).

Each block introduces at most $3m + 3$ changes. Including the initial oracle and the final change above length T, the number of oracles is at most

$$(3m+3)\left\lceil \frac{2}{\mu(T)} \right\rceil + 2 \leq \frac{24m}{\mu(T)} \leq n.$$

By repeating the final oracle if necessary, write the sequence as $\mathcal{O}_1, \ldots, \mathcal{O}_n$, with $\mathcal{O}_1 = \mathcal{O}$ and $\mathcal{O}_n = \mathcal{Z}$.

It remains to bound the distance between consecutive evaluations. When $P(x) = 1$, the prepared state is

$$\sin\left(\frac{\pi Q(x)}{2}\right)|\perp\rangle + \cos\left(\frac{\pi Q(x)}{2}\right)|O(x)\rangle.$$

Changing $Q(x)$ by at most $1/m$ changes this vector by at most $\pi/(2m)$ in Euclidean distance. The corresponding swaps differ by at most $2\pi/m$ in operator norm, using

$$\|S_{|\phi\rangle} - S_{|\psi\rangle}\|_{\mathrm{op}} \leq 4\||\phi\rangle - |\psi\rangle\|_2.$$

Changing P at $Q = 0$ leaves the oracle unchanged. Changing O at $P = 1, Q = 1$ also leaves it unchanged, since the prepared state is then $|\perp\rangle$. These bounds hold for superposition queries as well, because the oracle acts separately on each label.

Replacing the at most T queries one at a time therefore gives, for every $i \in [n-1]$ and every input k,

$$\mathsf{Tr}\left(\overline{G}^{\mathcal{O}_i}(k), \overline{G}^{\mathcal{O}_{i+1}}(k)\right) \leq \frac{2\pi T}{m} \leq \frac{1}{\lambda}. \tag{17}$$

The final change above length T has no effect on the computation. The same bounds apply to inverse queries, since the oracles are self-inverse.

By Eq. (13) and Markov's inequality, at most a $10\delta(\lambda)$ fraction of seeds have no output occurring with probability at least $9/10$ at any fixed intermediate oracle. Taking a union bound over the n oracles, all but a

$$10n\delta(\lambda) = \frac{1}{10\lambda}$$

fraction of seeds have such an output at every intermediate oracle. For each remaining seed, this output cannot change along the sequence: distinct outputs occurring with probability at least $9/10$ in consecutive distributions would give trace distance at least $4/5$, contradicting Eq. (17). Thus the same string satisfies Eqs. (14) and (15). □

We now give a search attack against $\overline{G}$ using $\mathcal{C}$. The fixed prefix of $\mathcal{Z}$ has finite rational tables and all its remaining entries are zero. Hence a PSPACE computation can estimate $\Pr[\overline{G}^{\mathcal{Z}}(k) = y]$ and distinguish the output from random. This contradicts the existence of a PRG in the same way as in the previous separation.

Therefore, there does not exist a BB construction of a w-PRG from a $(3\mu, \ell)$-$\perp$-PRG with inverse access. □

References

1. Ananth, P., Lin, Y.T., Yuen, H.: Pseudorandom strings from pseudorandom quantum states. arXiv preprint arXiv:2306.05613 (2023)

2. Ananth, P., Qian, L., Yuen, H.: Cryptography from pseudorandom quantum states. In: Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part I. pp. 208–236. Springer (2022)
3. Barhoush, M., Behera, A., Ozer, L., Salvail, L., Sattath, O.: Signatures from pseudorandom states via ⊥-prfs (2024), https://arxiv.org/abs/2311.00847
4. Barhoush, M., Nishimaki, R., Yamakawa, T.: Microcrypt assumptions with quantum input sampling and pseudodeterminism: Constructions and separations. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 516–548. Springer (2025)
5. Bartusek, J.: Secure quantum computation with classical communication. In: Theory of Cryptography Conference. pp. 1–30. Springer (2021)
6. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudorandom bits. SIAM Journal on Computing **13**(4), 850–864 (1984). https://doi.org/10.1137/0213053
7. Bouaziz-Ermann, S., Hhan, M., Muguruza, G., Vu, Q.H.: On limits on the provable consequences of quantum pseudorandomness. Cryptology ePrint Archive, Paper 2025/1863 (2025), https://eprint.iacr.org/2025/1863
8. Bouaziz-Ermann, S., Muguruza, G.: Quantum pseudorandomness cannot be shrunk in a black-box way (2024), https://doi.org/10.48550/arXiv.2402.13324
9. Brakerski, Z., Shmueli, O.: Scalable pseudorandom quantum states. In: Annual International Cryptology Conference. pp. 417–440. Springer (2020)
10. Chen, B., Coladangelo, A., Sattath, O.: The power of a single haar random state: constructing and separating quantum pseudorandomness. arXiv preprint arXiv:2404.03295 (2024)
11. Coladangelo, A., Mutreja, S.: On black-box separations of quantum digital signatures from pseudorandom states. In: Theory of Cryptography Conference. pp. 289–317. Springer (2024)
12. Dawson, C.M., Nielsen, M.A.: The Solovay–Kitaev algorithm (2005), https://arxiv.org/abs/quant-ph/0505030
13. Goldin, E., Zhandry, M.: Translating between the common haar random state model and the unitary model (2025), https://arxiv.org/abs/2503.11634
14. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Proceedings of the twenty-first annual ACM symposium on Theory of computing. pp. 44–61 (1989)
15. Ji, Z., Liu, Y.K., Song, F.: Pseudorandom quantum states. In: Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part III 38. pp. 126–152. Springer (2018)
16. Kretschmer, W.: Quantum pseudorandomness and classical complexity. In: 16th Conference on the Theory of Quantum Computation, Communication and Cryptography (2021)
17. Kretschmer, W., Qian, L., Tal, A.: Quantum-computable one-way functions without one-way functions. arXiv preprint arXiv:2411.02554 (2024)
18. Meckes, E.S.: The random matrix theory of the classical compact groups, vol. 218. Cambridge University Press (2019)
19. Morimae, T., Yamakawa, T.: Quantum commitments and signatures without one-way functions. In: Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part I. pp. 269–295. Springer (2022)

20. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge University Press, New York, NY, USA (2000). https://doi.org/10.1017/CBO9780511976667
21. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part III 37. pp. 520–551. Springer (2018)