MicroCrypt Assumptions with Quantum Input Sampling and Pseudodeterminism: Constructions and Separations

Mohammed Barhoush[1⋆], Ryo Nishimaki[2], and Takashi Yamakawa[2]

[1] Université de Montréal (DIRO), Montréal, Canada
mohammed.barhoush@umontreal.ca
[2] NTT Social Informatics Laboratories
ryo.nishimaki@ntt.com, takashi.yamakawa@ntt.com

Abstract. We investigate two natural relaxations of quantum cryptographic primitives. The first involves quantum input sampling, where inputs are generated by a quantum algorithm rather than sampled uniformly at random. Applying this to pseudorandom generators (PRGs) and pseudorandom states (PRSs), leads to the notions denoted as $\mathsf{PRG}^{\mathsf{qs}}$ and $\mathsf{PRS}^{\mathsf{qs}}$, respectively. The second relaxation, $\perp$-pseudodeterminism, relaxes the determinism requirement by allowing the output to be a special symbol $\perp$ on an inverse-polynomial fraction of inputs.

We demonstrate an equivalence between bounded-query logarithmic-size $\mathsf{PRS}^{\mathsf{qs}}$, logarithmic-size $\mathsf{PRS}^{\mathsf{qs}}$, and $\mathsf{PRG}^{\mathsf{qs}}$. Moreover, we establish that $\mathsf{PRG}^{\mathsf{qs}}$ can be constructed from $\perp$-PRGs, which in turn were built from logarithmic-size PRS. Interestingly, these relations remain unknown in the uniform key setting.

To further justify these relaxed models, we present black-box separations. Our results suggest that $\perp$-pseudodeterministic primitives may be weaker than their deterministic counterparts, and that primitives based on quantum input sampling may be inherently weaker than those using uniform sampling.

Together, these results provide numerous new insights into the structure and hierarchy of primitives within MicroCrypt.

Keywords: Quantum Cryptography · Pseudorandom States · Pseudodeterminism · Black-Box Separation

⋆ Part of this work was done while visiting NTT Social Informatics Laboratories as an internship.

Table of Contents

1 Introduction

The search for the minimal assumptions required for quantum cryptography was triggered with the astonishing discovery that pseudorandom states (PRSs) [23] may exist even when quantumly-evaluable [1] one-way functions (OWF) do not, relative to an oracle [2] [25]. PRSs serve as the quantum analog to PRGs, outputting a state instead of a classical string. Critically, this difference does not prevent PRSs supporting some applications similar to those enabled by PRGs, such as commitments, one-time signatures, and one-way state generators (OWSG)s [31, 3].

This separation naturally raised questions on the minimal assumptions required to build quantum cryptographic primitives. Addressing this question has fueled significant research, leading to a variety of quantum assumptions. Different assumptions provide a different balance between how well they replicate OWFs in cryptography and how strong of an assumption they constitute. The resulting assumptions are intricately related in what has now came to be known as *MicroCrypt*. This field comprises various assumptions derived from OWFs, but where the other direction is not known. Despite substantial progress, numerous questions remain unanswered. What is clear, however, is that MicroCrypt is significantly more intricate than its classical counterpart.

Several of the MicroCrypt assumptions introduced parallel their classical counterparts but incorporate quantum elements. For instance, quantum unpredictable state generators [29] and one-way state generators [31] yield quantum outputs, similar to PRSs. Additionally, assumptions such as $\perp$-PRG [4] and one-way puzzles [24] involve only classical communication but rely on quantum computation. These quantum elements are believed to make the assumptions weaker.

Despite advances, using general PRSs as a complete replacement for PRGs in cryptographic applications has been challenging. Some progress has been made in the specific case of logarithmic-size pseudorandom states, which we denote by *short* PRS *(*SPRS*)* as in [2], where tomography can transform the state into a classical pseudorandom string [2]. However, tomography is not deterministic, resulting in what has been termed *pseudodeterministic* PRG. This roughly means that on $1 - 1/\mathsf{poly}(n)$ fraction of inputs, the output is the same except with negligible probability [3]. While these generators proved useful in various applications, the pseudodeterminism is sometimes problematic when using them in place of traditional PRGs. This obstacle motivated a follow-up work [4], that introduced an intermediate notion called $\perp$-PRG, which is built from pseudodeterministic PRGs. With $\perp$-PRGs, the non-deterministic outcomes can be detected and replaced with $\perp$, allowing many PRG applications to proceed by handling $\perp$ cases separately. This approach enabled significant applications, such as many-

[1] In this work, all primitives including OWFs and PRG, refer to the quantum-evaluable versions, unless stated otherwise.

[2] Note that one-way functions and pseudorandom generators are equivalent.

[3] "Pseudodeterminism" is sometimes defined differently in other works [6, 5, 7]. We follow the definition given in [2].

time digital signatures and quantum public-key encryption with tamper-resilient keys, which had eluded MicroCrypt.

While these applications are powerful, $\bot$-PRGs and SPRSs have not been black-box separated from OWFs [4], which somewhat limits the significance of these results. In fact, most MicroCrypt assumptions, such as pseudorandom function-like states with proofs of destruction [8] and efficiently verifiable one-way puzzles (Ev-OWPuzzs) [24, 15], have been conjectured to be weaker than (quantum-evaluable) OWFs, but their separability has not been established. Understanding which assumptions are separated from OWFs and, more generally, the relations among different MicroCrypt primitives is an important goal in the field.

1.1 Our Work

Traditionally, many cryptographic primitives such as PRGs and PRSs rely on inputs sampled uniformly at random. The main idea of our work is that sampling inputs with a quantum procedure, instead of at random, yields fundamentally different primitives. We denote the resulting primitives with a superscript such as $\mathsf{PRG}^{\mathrm{qs}}$s and $\mathsf{PRS}^{\mathrm{qs}}$s.

To clarify, a pair of QPT algorithms (QSamp, G) is a $\mathsf{PRG}^{\mathrm{qs}}$ if $\mathsf{QSamp}(1^\lambda)$ samples a λ-bit input, which is then mapped by G to an output of length $\ell > \lambda$, and the following security condition is satisfied: For any QPT adversary $\mathcal{A}$,

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

Naturally, we also require that G is *almost-deterministic*, meaning that it returns the same output on a fixed input except with negligible probability. Notice that this notion differs from a traditional PRG, where the above security condition is guaranteed if the input k is sampled uniformly at random. Similarly, a $\mathsf{PRS}^{\mathrm{qs}}$ consists of a pair of algorithms (QSamp', G') such that for an input $k \leftarrow \mathsf{QSamp}'(1^\lambda)$, polynomial copies of $|\psi_k\rangle \leftarrow G'(k)$ are indistinguishable from polynomial copies of a Haar random state.

While variants of MicroCrypt primitives based on quantum input sampling have been considered before for notions such as OWSGs and PRSs [30, 24, 12], the fundamental distinction between primitives based on quantum versus uniform input sampling has not been previously recognized.

Note that any classical input sampling algorithm can be derandomized and replaced with uniform key sampling. Therefore, using a classical input sampling algorithm is unnecessary. However, a quantum procedure cannot be derandomized and might include non-deterministic quantum computations. Postponing such computations to the evaluation or state-generation phase can result in different outcomes across executions, which is problematic for deterministic primitives such as PRSs and PRGs.

[4] Notably, the separation between PRS and OWFs [25] only applies to linear-sized PRSs and not to SPRS.

Before moving to a detailed explanation, we first list a brief high-level description of the main contributions of this paper. For the sake of simplicity, this summary is somewhat weaker than what we actually accomplish. We show the following:

- The pseudodeterminism error in $\perp$-PRGs can be eliminated if quantum input sampling is allowed. In particular, $\perp$-PRG imply $\mathsf{PRG}^{\mathsf{qs}}$. This stands in contrast to the fact that $\perp$-PRG are not known to imply PRG.
- Primitives with quantum input sampling behave differently from their uniform input sampling counterparts. In particular, we realize that $\mathsf{PRG}^{\mathsf{qs}}$, bounded-copy $\mathsf{SPRS}^{\mathsf{qs}}$, and $\mathsf{SPRS}^{\mathsf{qs}}$ are all equivalent under a certain parameter regime. Such an equivalence is not known to exist in the uniform input sampling setting.
- While PRGs trivially imply $\mathsf{PRG}^{\mathsf{qs}}$s, we show that the reverse implication is unlikely by showing that there is no black-box construction of PRGs from a $\mathsf{PRF}^{\mathsf{qs}}$s, even with unitary and inverse access to the $\mathsf{PRF}^{\mathsf{qs}}$. In addition, we provide more fine-grained black-box separations: separating PRG from $\perp$-PRG, and $\perp$-PRG from $\mathsf{PRG}^{\mathsf{qs}}$, albeit within a weaker oracle access model. Thus, we establish a hierarchy among uniform input sampling, pseudodeterministic, and quantum input sampling primitives within MicroCrypt.

1.1.1 Quantum Input Sampling. In the first part of this work, we introduce natural variants of MicroCrypt primitives that incorporate quantum input sampling, and demonstrate how this framework helps address the issue of pseudodeterminism.

Recall that [4] extended the applicability of SPRS by converting them into $\perp$-PRGs, which inherit many of the useful properties of PRGs. However, $\perp$-PRGs still exhibit a form of non-determinism due to the possibility of outputting $\perp$, which may be problematic in certain applications. Specifically, for a $\perp$-PRG, there exists a set of "good" inputs, that produce deterministic outputs, and an inverse-polynomial fraction of inputs termed "bad" inputs, that may yield $\perp$. To address this non-determinism, a natural solution is to test inputs during the sampling process to ensure that only good inputs are selected. We show that this technique can be used to construct a $\mathsf{PRG}^{\mathsf{qs}}$ from a $\perp$-PRG, thus resolving the pseudodeterminism issue. Note that this approach necessitates a quantum input sampling procedure instead of traditional uniform input sampling, since the $\perp$-PRG itself may be a quantum algorithm.

We utilize this result, along with other key insights, to establish the following relationships among primitives with quantum input sampling. Specifically, we show fully black-box constructions for the following:

1. $\mathsf{PRG}^{\mathsf{qs}}$ from bounded-copy $\mathsf{SPRS}^{\mathsf{qs}}$ ($\mathsf{BC\text{-}SPRS}^{\mathsf{qs}}$).
2. $\mathsf{SPRS}^{\mathsf{qs}}$ from $\mathsf{PRG}^{\mathsf{qs}}$.
3. $\mathsf{BQ\text{-}PRU}^{\mathsf{qs}}$ [5] from $\mathsf{PRG}^{\mathsf{qs}}$.

[5] This stands for bounded-query pseudorandom unitaries with quantum key sampling (see Definition 16).

4. $\mathsf{PRU}^{\mathrm{qs}}$ from $\mathsf{PRF}^{\mathrm{qs}}$.

These findings mean that $\mathsf{PRG}^{\mathrm{qs}}$, $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$, and $\mathsf{SPRS}^{\mathrm{qs}}$ can all be built from one another under a certain parameter regime. This relationship is surprising, as it is not known to hold in the uniform input sampling setting. Specifically, it is not known how to construct a PRG from a SPRS, nor how to build a SPRS from a bounded-copy SPRS.

Furthermore, as a direct consequence of these results, we obtain both a method to reduce the output length of a $\mathsf{SPRS}^{\mathrm{qs}}$ and a way to transform a SPRS into a $\mathsf{SPRS}^{\mathrm{qs}}$ with a longer output length.

1.1.2 Separations. In the second part of our work, we extend our analysis with separation results that highlight the distinctions between quantum input sampling and uniform input sampling and between $\perp$-pseudodeterminism and determinism.

Our separation results demonstrate the impossibilities of certain types of black-box constructions. There are different variants of black-box constructions in quantum cryptography (see [16] for an exposition). We informally define the two variants considered in this work.

Definition 1 (Informal version of Definition 5). *A QPT algorithm $G^{(\cdot)}$ is a* fully black-box construction of a primitive Q from a primitive P with inverse access *if there is a QPT algorithm $S^{(\cdot)}$ such that for every unitary implementation U of P:*

- *$G^{U,U^{-1}}$ is an implementation of Q.*
- *Every attack $\mathcal{A}$ that breaks the security of $G^{U,U^{-1}}$, and every unitary implementation $\tilde{\mathcal{A}}$ of $\mathcal{A}$, it holds that $S^{\tilde{\mathcal{A}},\tilde{\mathcal{A}}^{-1}}$ breaks the security of U.*

We also consider a less general notion limited to *completely-positive-trace-preserving (CPTP) maps*. These are quantum channels that are not necessarily unitary.

Definition 2 (Informal version of Definition 6). *A QPT algorithm $G^{(\cdot)}$ is a* fully black-box construction of Q from CPTP access to P *if there is a QPT algorithm $S^{(\cdot)}$ such that for every CPTP implementation $\mathcal{C}$ of P:*

- *$G^{\mathcal{C}}$ is an implementation of Q.*
- *Every attack $\mathcal{A}$ that breaks the security of $G^{\mathcal{C}}$, it holds that $S^{\mathcal{A}}$ breaks the security of U.*

Since CPTP maps are not necessarily unitary and cannot be purified or inverted, Definition 2 only includes constructions that do not assume purified/inverse/unitary access. On the other hand, Definition 1 includes constructions that use such access, thus covering a broader range of constructions. We discuss these distinctions more thoroughly in Section 1.1.3, but for now, we state the results.

Our first and main separation is between PRGs and $\mathsf{PRF}^{\mathrm{qs}}$s, with inverse access.

Theorem 1 (Informal version of Theorem 9). *There does not exist a fully black-box construction of a* PRG *from a (quantum-query-secure)* PRF^{qs}*s with inverse access.*

Given that $\mathsf{PRF}^{\mathrm{qs}}$s inherits many applications of PRFs, we obtain other new separations as a corollary.

Corollary 1. *There is no fully black-box construction of a* PRG *from the following primitives with inverse access:*

1. PRG^{qs}*,* SPRS^{qs}*, linear-sized* PRS^{qs}*, and* PRU^{qs}*.*
2. *Statistically-binding, computationally hiding bit commitments with classical communication (BC-CC).*
3. *Existentially unforgeable message authentication codes of classical messages with classical communication (EUF-MAC).*
4. *CCA2-secure symmetric encryption with classical keys and ciphertexts (CCA2-SKE).*
5. *EV-OWPuzzs.*

Our second separation is between $\perp$-pseudodeterministic notions and deterministic ones, but with CPTP access.

Theorem 2 (Informal version of Theorem 12). *There does not exist a fully black-box construction of a* OWSG *from CPTP access to a* $\perp$-PRG*.*

Note that the OWSGs considered in this paper are those with pure-state outputs and with uniform key generation. Our separation is further emphasized by the fact that a OWSG is considered weaker than a PRG, since PRSs imply OWSGs and PRGs are separated from PRSs [25]. As $\perp$-PRGs have broad applicability [4], this result yields additional separations as corollaries.

Corollary 2 (Informal version of Corollary 7). *There does not exist a fully black-box construction of a* OWSG*s from CPTP access to:*

1. $\perp$-PRF*s.*
2. *(Many-time) existentially unforgeable digital signatures of classical messages with classical keys and signatures (EUF-DS).*
3. *CPA-secure quantum public-key encryption of classical messages with tamper-resilient keys and classical ciphertexts (CPA-QPKE).*

Our third separation shows that CPTP access to $\mathsf{PRF}^{\mathrm{qs}}$s is insufficient for constructing $\perp$-PRGs. Recall that our second separation establishes a gap between PRGs and $\perp$-PRGs. Taken together, this means the third separation strengthens the first by demonstrating a separation between $\mathsf{PRF}^{\mathrm{qs}}$ and $\perp$-PRG. However, this result is more limited in scope, as it only applies to CPTP access.

Theorem 3 (Informal version of Theorem 14). *There does not exist a fully black-box construction of a* $\perp$-PRG *from CPTP access to a (quantum-query-secure)* PRF^{qs}*s.*

We obtain other new separations as a corollary.

Corollary 3 (Informal version of Corollary 8). *There does not exist fully black-box constructions of* $\perp$-PRG*s or* SPRS*s from CPTP access to:*

1. PRG^{qs}, SPRS^{qs}, *linear-sized* PRS^{qs}, *and* PRU^{qs}.
2. *BC-CC.*
3. *EUF-MAC and CCA2-SKE.*
4. *EV-OWPuzzs.*

Note that separations listed in Corollaries 1 to 3 were not known prior to this work. This highlights how $\perp$-PRGs and $\mathsf{PRG}^{\mathrm{qs}}$ not only aid in building applications for SPRS, but also in establishing separations among well-studied MicroCrypt assumptions that may be difficult to separate otherwise. For instance, EV-OWPuzzs have been studied and introduced as a potentially weaker replacement to (quantum-evaluable) OWFs, but no separation existed prior to our work.

Our results give a natural hierarchy in MicroCrypt as depicted in Fig. 1.

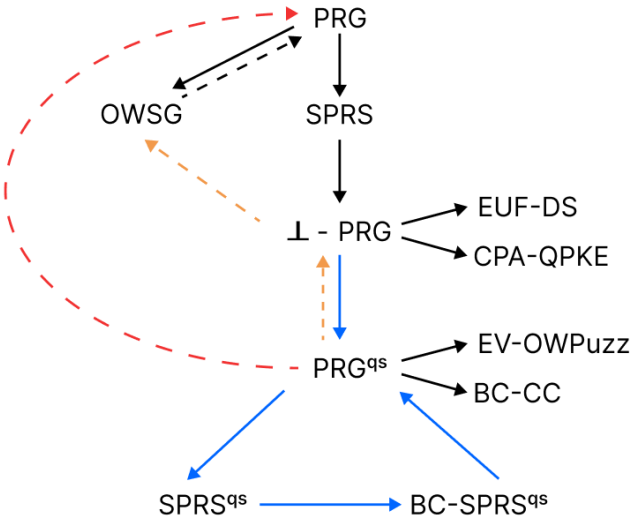

Fig. 1. The black straight arrows indicate implications that are trivial or from previous works [31, 2, 4]. The black dotted arrow indicates a separation from previous work [25]. The blue straight arrows are implications from this work. The red dotted arrow is a separation under inverse access from this work. The orange dotted arrows are separations under CPTP access from this work.

1.1.3 Discussion of Separations. Separation results make use of the fact that fully black-box constructions relativize [22, 16]. In particular, to establish the non-existence of black-box constructions granting unitary/inverse access in the plain model (Definition 5), it is sufficient to show that there does not exist such constructions relative to a unitary oracle with inverse access. Similarly, to rule out black-box constructions with CPTP access (Definition 6), it is sufficient to show that no such constructions exist relative to a CPTP oracle.

In particular, our first separation (Theorem 1), separating PRG from inverse access to $\mathsf{PRF}^{\mathrm{qs}}$s, is achieved by demonstrating the unattainability of such constructions relative to a unitary oracle with inverse access. Advantageously, this separation precludes black-box constructions that use purified/inverse/unitary versions of the adversary. Certain reductions, such as in quantum proofs of knowledge [35, 34] and quantum traitor tracing [39], rely on inverse access to the adversary.

On the other hand, the second and third separations (Theorems 2 and 3) are proven by demonstrating the impossibility relative to CPTP oracles, which are not unitary and cannot be purified or inverted. As a result, these separations are weaker as they only preclude black-box constructions that do not assume access to purified/inverse/unitary versions of the adversary.

However, in many black-box constructions, such strong forms of access (e.g., purification, unitarization, or inversion of the adversary) are unnecessary. Indeed, many known black-box constructions in this domain, such as $\mathsf{PRG} \to \perp\text{-}\mathsf{PRG}$, $\perp\text{-}\mathsf{PRG} \to \mathsf{PRG}^{\mathrm{qs}}$, and $\mathsf{PRG} \to \mathsf{OWSG}$ do not require such strong access. Hence, we believe that separations based on CPTP oracles still yields meaningful results. Notably, prior works [19, 20] have also considered CPTP oracle separations.

That said, we also investigated whether the second and third separations could be based on unitary oracles. For the separation between deterministic and psuedodeterministic primitives, specifically OWSGs and $\perp$-PRG, doing so appears particularly challenging: our separation strategy relies on inherent randomness in the oracle to achieve a level of pseudodeterminism, which is critical to ensuring that the oracle cannot be leveraged to build deterministic primitives such as OWSGs, while still being useful in building $\perp$-PRGs. Yet, another difficulty emerged when attempting to separate $\perp$-PRG from $\mathsf{PRF}^{\mathrm{qs}}$ using a unitary oracle, due to the pseudodeterminism of $\perp$-PRG, as we shall discuss in Section 1.3. An interesting avenue for future work is to strengthen the second or third separation by basing them on unitary oracles.

1.2 Relation to Previous Work

We discuss relation to previous work.

- Prior research has typically defined PRSs and OWSGs with uniform input sampling. However, some works have defined them with quantum input sampling, such as [30, 24]. Nevertheless, the relations among certain MicroCrypt primitives with quantum sampling and the advantage of quantum sampling

in yielding potentially weaker assumptions have not been previously recognized. The latter insight is crucial for known MicroCrypt primitives as well, enabling us to identify multiple new separations, as outlined in the previous section (Corollaries 1 to 3). None of the separations mentioned were known prior to this work.
- Previous research did not establish a connection between quantum input sampling and $\perp$-pseudodeterminism. This connection enabled us to address the pseudodeterminism inherent in $\perp$-PRGs (built in [4] from SPRS) by converting them to $\mathsf{PRG}^{\mathrm{qs}}$s.
- It may seem that EV-OWPuzzs can be viewed as a one-way function with a quantum input sampler. However, critically, these puzzles are non-deterministic. In fact, [15] use this property to show that uniform and quantum sampling versions of these puzzles are equivalent. Hence, quantum input sampling seems more interesting to study in deterministic notions such as PRGs.
- All definitions in this work consider quantum-evaluable algorithms, and we never require any algorithm to be classically-evaluable. This is noteworthy since, very recently, [26] used a classical oracle to show a separation between classically-evaluable OWFs and quantumly-evaluable OWFs. Our separations are not comparable with theirs. However, as a direct result, they find that classically-evaluable OWFs are black-box separated from many cryptographic applications of quantum-evaluable OWFs, such as EUF-MAC and CCA2-SKE. Given that many of these applications can be built from $\mathsf{PRF}^{\mathrm{qs}}$ in the same way, our separations imply that even OWSGs are separated from these applications under CPTP access, and (quantum-evaluable) OWFs are separated from these applications under inverse access.

1.3 Technical Overview

We now describe our results in more detail.

1.3.1 Quantum Input Sampling. Our study reveals surprising equivalences among several variants of MicroCrypt primitives utilizing quantum input sampling.

BC-SPRS^{qs}s imply PRG^{qs}s. Our first result is that *bounded-copy* $\mathsf{SPRS}^{\mathrm{qs}}$ imply $\mathsf{PRG}^{\mathrm{qs}}$s. Prior works [2, 4] demonstrated that SPRSs enable $\perp$-PRGs. We first extend this result by showing that bounded-copy SPRSs (BC-SPRS) suffice for this construction.

At first glance, using BC-SPRS appears infeasible because each evaluation of the $\perp$-PRG exhausts several copies of the SPRS and the adversary has access to arbitrarily many evaluations in the security experiment. However, the $\perp$-PRG only uses these copies to perform tomography and extract a classical string. Crucially, the extracted strings remain largely consistent across evaluations. Thus, the information gained through multiple evaluations can be simulated with only a limited number of SPRS copies. By formalizing this observation, we construct a $\perp$-PRG from a BC-SPRS.

Our next idea is to show that $\perp$-PRGs imply (deterministic) PRG$^{\text{qs}}$s, thereby showing that quantum input sampling resolves the pseudodeterminism problem. The idea is simple: to construct a PRG$^{\text{qs}}$ from a $\perp$-PRG, we search for a good input for the $\perp$-PRG during the input sampling phase and use this input during evaluation. However, this approach sacrifices uniform input sampling, which compromises the security reduction given in [2, 4], thereby only giving a *weak* PRG$^{\text{qs}}$. Standard amplification techniques are then applied to achieve strong security. Hence, we obtain PRG$^{\text{qs}}$ from BC-SPRS.

Finally, since we are allowed to use a quantum input sampler for a PRG$^{\text{qs}}$, we can perform the same conversion starting instead with a BC-SPRS$^{\text{qs}}$. Therefore, we obtain PRG$^{\text{qs}}$s from BC-SPRS$^{\text{qs}}$.

PRG^{qs}s imply SPRS^{qs}s. We also establish the converse: SPRS$^{\text{qs}}$s can be built from PRG$^{\text{qs}}$s. This follows in the same way as the construction of PRSs from PRFs given in [23], but instantiated with a polynomial-domain PRF. Note that PRF$^{\text{qs}}$ with polynomial domain can be trivially derived from PRG$^{\text{qs}}$s [6].

Modifying the Size of SPRS. By leveraging the above equivalence, we obtain a way to decrease the output length of a SPRS$^{\text{qs}}$. Simply convert a SPRS$^{\text{qs}}$ into a PRG$^{\text{qs}}$, and then convert this back into a SPRS$^{\text{qs}}$. Due to the change in parameters during this conversion, the resulting SPRS$^{\text{qs}}$ is smaller in size. Interestingly, a similar result is not known for SPRS.

Furthermore, these equivalences can also be leveraged to increase the output length of a SPRS. Note that it is trivial to extend the output length of a $\perp$-PRG by composition. For instance, if G_λ is a $\perp$-PRG mapping $\{0,1\}^\lambda$ to $\{0,1\}^{2\lambda}$, i.e. with expansion factor of 2, then the composition $G_{2\lambda} \circ G_\lambda$ is a $\perp$-PRG with expansion factor 4. Hence, starting with a SPRS, building a $\perp$-PRG, extending the output length of the $\perp$-PRG sufficiently through composition, building a PRG$^{\text{qs}}$, and finally converting this to a SPRS$^{\text{qs}}$, we obtain a method to convert a SPRS into a SPRS$^{\text{qs}}$ with larger output length.

Other Constructions. On the downside, we face an obstacle in the quantum input sampling regime when attempting to build a PRF$^{\text{qs}}$s from PRG$^{\text{qs}}$s. While a PRG$^{\text{qs}}$ with sufficient expansion easily implies a PRF$^{\text{qs}}$ with polynomial domain, it is not clear if a PRG$^{\text{qs}}$ can be used to build full-fledged PRF$^{\text{qs}}$ with exponential domain. Note that the standard conversion of a PRG to a PRF [21], and its quantum adaption [38], both implicitly use the uniform input sampling property of PRGs. Hence, adapting this conversion to the quantum input sampling setting is an interesting open question.

Fortunately, PRF$^{\text{qs}}$ with polynomial domain can still be useful. Using domain extension techniques [18], we convert them into *bounded-query* PRF$^{\text{qs}}$s with exponential domain. These, in turn, enable the construction of bounded-copy linear-length PRS$^{\text{qs}}$ and bounded-query PRU$^{\text{qs}}$, following a similar approach outlined in [23, 27] for the uniform sampling setting.

[6] The output of the PRG$^{\text{qs}}$ can be interpreted as the complete description of a PRF$^{\text{qs}}$ with polynomial domain.

1.3.2 Separation Results We present three separation results in this work that complement the positive results discussed above. We only provide a simplified overview of the core ideas of the separations and many technical considerations are not discussed here to maintain clarity.

Separating PRG*s from* PRF^{qs}*s.* Our first and main separation is between PRGs and PRF$^{\text{qs}}$s. This separation is established using a unitary oracle with access to the inverse. As discussed in Section 1.1.3, this precludes a wide class of black-box constructions.

We define three oracles based on a pair of random functions (O, P):

- $\mathcal{C}$: An oracle for membership in a PSPACE-complete language.
- σ: A unitary "flip" oracle that swaps the state $|0^{2n}\rangle$ with the state $|\psi_n\rangle = \sum_{x\in\{0,1\}^n} |x\rangle|O(x)\rangle$ and acts as the identity on all other orthogonal states [7].
- $\mathcal{O}$: Unitary of the classical function that maps (x, y, z) to $P(x, z)$ if $O(x) = y$ and to $\perp$ otherwise.

It is straightforward to construct a PRF$^{\text{qs}}$ relative to these oracles. The quantum key generation algorithm of the PRF$^{\text{qs}}$ queries $\sigma(|0^{2n}\rangle)$ and measures the response in the computational basis to obtain a pair $(x^*, O(x^*))$, which serves as the secret key. On input z, the evaluation algorithm computes $\mathcal{O}(x^*, O(x^*), z) = P(x^*, z)$ and outputs the result $P(x^*, z)$. Since the same key is reused across evaluations, this yields deterministic outputs. Furthermore, given that a quantum-accessible random oracle acts as a (quantum-query-secure) PRF [33], it is not difficult to show that this construction constitutes a PRF$^{\text{qs}}$.

The more difficult part is showing that PRGs cannot exist relative to these oracles. Our strategy is to show that any candidate generator must be independent of the oracles $(\sigma, \mathcal{O})$ in order to establish that it can be broken using $\mathcal{C}$. In particular, measuring $\sigma(|0^{2n}\rangle)$ returns a different pair $(x, O(x))$ each time, and $\mathcal{O}$ then evaluates a different function $P(x, \cdot)$ depending on the value of x. On the other hand, a PRG should produce almost-deterministic outputs, meaning that it cannot depend on this inconsistent randomness.

To make this rigorous, we make small modifications to the oracle and argue that the generator's output should remain stable under such perturbations. The main idea is that a PRG cannot distinguish between oracles that differ only negligibly in the states they produce. Given that a PRG produces deterministic classical values, this implies that its output must remain invariant under such small perturbations. By applying this reasoning inductively, we conclude that the generator's output must remain stable even if the oracles are replaced entirely.

This, in turn, implies that the oracles can be simulated using internal randomness, effectively yielding a PRG that does not require oracle access. In other words, since the PRG's output is the same regardless of the oracle, we can simulate its computation by sampling random strings directly, rather than

[7] Similar "flip" oracles were used in recent works: [14, 11, 9].

querying the oracles. But a PRG that does not use oracle access to $(\sigma, \mathcal{O})$ can easily be broken with a PSPACE oracle. Thus, we establish that relative to the unitary oracles $(\sigma, \mathcal{O}, \mathcal{C})$, there is no fully black-box construction of a PRG from a $\mathsf{PRF}^{\mathrm{qs}}$ with inverse access. Given that fully black-box constructions relativize, we obtain the impossibility of such constructions in the plain model.

Separating OWSG*s from* $\perp$-PRG*s.* We show that there does not exist a fully black-box construction of a OWSG from CPTP access to a $\perp$-PRG. We prove this by demonstrating the separation relative to a CPTP oracle. This result is somewhat surprising, since OWSGs are considered weaker than PRGs, as evidenced by existing separations between them [25]. Instead, our separation emphasizes the distinction in determinism to separate OWSGs from $\perp$-PRGs. Recall, a $\perp$-PRG is the same as a PRG except on a $1/\mathsf{poly}(n)$ fraction of inputs, the algorithm may return $\perp$ sometimes.

Our separation leverages two oracles. The first oracle is for membership in a PSPACE-complete language, used to break any OWSG. The second oracle, denoted $\mathcal{O}$, is a modified quantum random oracle with an abort mechanism. This oracle exhibits inherent $\perp$-pseudodeterminism, making it well-suited for constructing $\perp$-PRGs but unsuitable for deterministic primitives like OWSGs. One downside of this design is that it complicates lifting the oracle to a unitary map.

The oracle $\mathcal{O}$ operates as follows: on input $x \in \{0,1\}^n$, it computes $(a_x, b_x, c_x) \leftarrow O(x)$, where O is a random function mapping n-bits to $3n$-bits. The first component, a_x, determines whether x is a "good" (deterministic) or "bad" (may evaluate to $\perp$) input. If x is deemed bad, which occurs with $1/\mathsf{poly}(n)$ probability, then $\mathcal{O}$ outputs c_x with probability $1 - \frac{b_x}{2^n}$ and $\perp$ with $\frac{b_x}{2^n}$ probability, where b_x is interpreted as an integer. Otherwise, if x is deemed good, $\mathcal{O}$ outputs c_x with probability 1. It is easy to verify that $\mathcal{O}$ behaves as a valid $\perp$-PRG.

The challenge lies in showing that $\mathcal{O}$ cannot be used for constructing OWSGs. Formalizing this requires some effort, but the main idea is that a generator, relying on $\mathcal{O}(x)$ for some x, cannot infer whether x is good or bad with certainty, given that the error probability (i.e., the chance of receiving $\perp$) can be very small. When x is bad, small changes in the error probability cannot be distinguished in polynomial time. As a result, the generator's output must remain stable under such small variations. Extending this reasoning, we show that the output must remain unchanged even when $\mathcal{O}(x)$ always returns $\perp$. Informally, this implies that the generator's output is independent of $\mathcal{O}$. This independence allows us to construct an OWSG that does not use $\mathcal{O}$. But since OWSGs cannot exist relative to a PSPACE oracle [13], we reach a contradiction. Hence, no fully black-box construction of an OWSG from a $\perp$-PRG is possible in this setting.

Separating $\perp$-PRG*s from* PRF^{qs}*s.* Our third result separates $\perp$-PRGs from CPTP access to a $\mathsf{PRF}^{\mathrm{qs}}$s. We prove this result by demonstrating a separation relative to a CPTP oracle. This separation, like our first, hinges on the distinction between quantum and classical input sampling procedures. We use a similar oracle

setup—$(\sigma, \mathcal{O}, \mathcal{C})$—but in this case, σ is defined as a CPTP map that samples x uniformly at random and outputs the pair $(x, O(x))$.

Constructing a $\mathsf{PRF}^{\mathrm{qs}}$ relative to these oracles proceeds analogously to the first separation and is relatively straightforward. The more involved part is proving that $\perp$-PRGs cannot exist relative to this oracle setup. Intuitively, since σ outputs fresh, random pairs $(x, O(x))$ with each call, every evaluation of the $\perp$-PRG interacts with an essentially independent instantiation of $\mathcal{O}$. Because $\perp$-PRGs are expected to exhibit some degree of determinism, they cannot depend on these inconsistent oracle outputs. More precisely, we show that the behavior of the $\perp$-PRG can be simulated by replacing oracle access with sampling random strings directly.

Consequently, there would exist a $\perp$-PRG that operates without oracle access yet remains secure against adversaries with access to a PSPACE oracle, which is a contradiction.

Importantly, this argument breaks down if σ were defined as in the first separation. In that setting, σ is a unitary map, so its outputs are no longer distinct. Furthermore, unlike in the first separation, we cannot argue that the $\perp$-PRG output remains invariant under small perturbations to the oracle responses: slight variations in σ could induce slight differences in the probability of returning $\perp$. Consequently, the set of "bad" inputs as well as the evaluations on this set can change across different oracles. To avoid this issue, we require that σ yield classical outputs, ensuring that different evaluations receive completely independent query results.

2 Preliminaries

2.1 Notations

We let $[n] = \{1, 2, \ldots, n\}$, $[n : n + k] = \{n, n + 1, \ldots, n + k\}$, and $y_{[n:n+k]} = y_n y_{n+1} \ldots y_{n+k}$ for every $n, k \in \mathbb{N}$ and string y of length at least $n + k$. Furthermore, we let $\mathsf{negl}(x)$ denote any function that is asymptotically smaller than the inverse of any polynomial.

We let $x \leftarrow X$ denote that x is chosen from the values in X, 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 say $(a, \cdot) \in X$ if there exists an element b such that $(a, b) \in X$. We let $\Pi_{n,m} = (\{0,1\}^m)^{\{0,1\}^n}$ denote the set of functions mapping n-bits to m-bits, and Π_n denote the set of permutations on n-bits.

We refer the reader to [32] for a detailed exposition to preliminary quantum information. We let $\mathcal{S}(\mathcal{H})$ and $\mathcal{U}(\mathcal{H})$ denote the set of unit vectors and unitary operators, respectively, on the Hilbert space $\mathcal{H}$ and let $\mathsf{Haar}(\mathbb{C}^d)$ denote the Haar measure over $\mathbb{C}^d$ which is the uniform measure over all d-dimensional unit vectors. We let d_{TD} denote the total trace distance between two density matrices or two distributions.

We follow the standard notations to define quantum algorithms. We say that a quantum algorithm A is QPT if it consists of a family of quantum algorithms $\{A_\lambda\}_\lambda$ such that the run-time of each algorithm A_λ is bounded by some

polynomial $p(\lambda)$. Furthermore, we say that a quantum algorithm $A = \{A_\lambda\}_\lambda$ is *almost-deterministic* if there exists a negligible function ϵ, such that for every $\lambda \in \mathbb{N}$ and every input x in the domain of A_λ, there exists an (possibly quantum) output y satisfying $\Pr[A_\lambda(x) = y] \geq 1-\epsilon(\lambda)$. We also avoid using the λ subscript in algorithms to avoid excessive notation.

We say A has *quantum-query access* to a classical function P to mean that it is given polynomial quantum queries to the unitary map $U_P : |x\rangle \to |x\rangle|P(x)\rangle$.

2.2 Black-Box Separation

The notion of oracle black-box separations was first considered in [22] and later formalized in the quantum setting in [16]. We just recall the definitions relevant for this work from [16].

Definition 3. *A* primitive *P is a pair $P = (\mathcal{F}_P, \mathcal{R}_P)$* [8] *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.*

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

Definition 4. *We say that a primitive P exists relative to an oracle $\mathcal{O}$ if:*

- *There exists QPT oracle-access 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 all QPT $\mathcal{A}$, $(G^{\mathcal{O}}, \mathcal{A}^{\mathcal{O}}) \notin \mathcal{R}_P$.*

We are now ready to define the notion of fully black-box construction. We define two versions: under unitary/inverse access and under CPTP access.

Definition 5. *A QPT algorithm $G^{(\cdot)}$ is a fully black-box construction 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 [16] shows the relation between fully black-box constructions and oracle separations.

[8] 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.

Theorem 4 (Theorem 4.2 in [16]). *Assume there exists a fully black-box 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 $(\mathcal{O}, \mathcal{O}^{-1})$.*

We now consider the setting of CPTP oracles.

Definition 6. *A QPT algorithm $G^{(\cdot)}$ is a fully black-box construction of Q from **CPTP access** to P if the following two conditions hold:*

1. *For every CPTP implementation $\mathcal{C}$ of P, $G^{\mathcal{C}}$ is an implementation of Q.*
2. *There is a QPT algorithm $S^{(\cdot)}$ such that, for every CPTP implementation $\mathcal{C}$ of P, every adversary $\mathcal{A}$ that Q-breaks $G^{\mathcal{C}}$, it holds that $S^{\mathcal{A}}$ P-breaks $\mathcal{C}$.*

We also obtain a relationship between fully black-box constructions and oracle separations in the CPTP setting.

Theorem 5. *Assume there exists a fully black-box construction of a primitive Q from CPTP access to P. Then, for any CPTP oracle $\mathcal{O}$, if P exists relative to $\mathcal{O}$, then Q exists relative to $\mathcal{O}$.*

The proof of the above result follows in the same way as the proof of Theorem 4.2 in [16].

2.3 MicroCrypt Primitives

We recall several MicroCrypt assumptions relevant to this work.

First, we recall the notion of one-way state generators (OWSGs). In this work, we only consider *pure* OWSGs meaning that the output is always a pure state.

Note that different works define correctness of a OWSG slightly differently. Our notion requires that on any input, the generator produces a fixed pure-state except with negligible probability. This encompasses some earlier definitions, such as in [13], but is less general than other variants, such as in [31]. We note, however, that our separation can be adapted to a slightly more general notion which only requires that the generator produce a fixed pure-state except with negligible probability on all but a negligible fraction of inputs.

However, our separation does not apply to certain more general notions such as in [31]. Specifically, our separation result uses our correctness condition and the result does not hold for the more generalized notion of [31]. We do not believe this to be a major issue since, to our knowledge, all known constructions of OWSGs satisfy our correctness condition.

Definition 7 (One-Way State Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be polynomial in λ. An almost-deterministic [9] QPT*

[9] Recall, G is *almost-deterministic* if on any input, G outputs the same value except with negligible probability. See Section 2.1 for the formal definition.

algorithm G is called a n-one-way state generator (OWSG)*, if it generates n-qubit pure-states and for any polynomial $t = t(\lambda)$ and QPT distinguisher $\mathcal{A}$, there exists a function $\epsilon(\cdot)$ such that:*

$$\mathsf{Adtg}^{\mathsf{OWSG}}_{\mathcal{A},G}(1^\lambda, 1^t) := \Pr\left[\mathsf{Exp}^{\mathsf{OWSG}}_{\mathcal{A},G}(1^\lambda, 1^t) = 1\right] \leq \epsilon(\lambda).$$

We say that G is a OWSG *if for every QPT $\mathcal{A}$, ϵ is a negligible function. We say that G is a* weak OWSG *if for every QPT $\mathcal{A}$, $\epsilon \leq \frac{1}{p}$ for some polynomial p.*

$\mathsf{Exp}^{\mathsf{OWSG}}_{\mathcal{A},G}(1^\lambda, 1^t)$:

1. Sample $k \leftarrow \{0,1\}^\lambda$.
2. For each $i \in [t+1]$ generate $|\psi_i\rangle \leftarrow G(k)$.
3. $k' \leftarrow \mathcal{A}(\otimes_{i\in[t]}|\psi_i\rangle)$. Let $|\phi_{k'}\rangle \leftarrow G(k')$.
4. Measure $|\psi_{t+1}\rangle$ with $\{|\phi_{k'}\rangle\langle\phi_{k'}|, I - |\phi_{k'}\rangle\langle\phi_{k'}|\}$ and if the result is $|\phi_{k'}\rangle\langle\phi_{k'}|$, then output $b = 1$, and output $b = 0$ otherwise.

[13] show that if $\mathsf{PSPACE} = \mathsf{BQP}$, then OWSGs do not exist. In fact, the attack presented succeeds with probability at least $\frac{1}{2}$.

Lemma 1 ([13]). *For any* OWSG *G, there exists a PSPACE algorithm $\mathcal{A}$ and a polynomial $t = t(\lambda)$ such that*

$$\mathsf{Adtg}^{\mathsf{OWSG}}_{\mathcal{A},G}(1^\lambda, 1^t) \geq \frac{1}{2}.$$

We define pseudorandom states (PRSs), first introduced in [23].

Definition 8 (Pseudorandom State Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be polynomial in λ. An almost-deterministic QPT algorithm PRS is called a* n-pseudorandom state generator (PRS)*, if it generates n-qubit pure-states and the following holds:*

For any polynomial $t(\cdot)$ and QPT distinguisher $\mathcal{A}$:

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

We divide PRS *into three regimes, based on the state size n:*

1. *$n = c \cdot \log(\lambda)$ with $c \ll 1$.*
2. *$n = c \cdot \log(\lambda)$ with $c \geq 1$, which we call* short pseudorandom states (SPRSs).
3. *$n = \Omega(\lambda)$, which we call* long pseudorandom states (LPRSs).

We will also recall the standard definition for PRGs.

Definition 9 (Pseudorandom Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be polynomial in λ. An almost-deterministic QPT algorithm G mapping $\{0,1\}^\lambda$ to $\{0,1\}^n$ is called a* n-pseudorandom generator (PRG)*, if $n > \lambda$ for all $\lambda \in \mathbb{N}$ and for any QPT distinguisher $\mathcal{A}$:*

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

2.4 Pseudodeterministic Pseudorandom Strings from Pseudorandom States

We describe the procedure given in [2] to extract classical pseudorandom strings from pseudorandom states. The original procedure is, for some states, pseudodeterministic meaning that running this procedure on the same state may yield different outcomes each time. However, it was shown that there exists a good set of states such that the extraction procedure is deterministic.

We first recall the notion of tomography, which is used to extract a classical approximate description of a quantum state.

Lemma 2 (Corollary 7.6 [1]). *For any error tolerance $\delta = \delta(\lambda) \in (0, 1]$ and any dimension $d = d(\lambda) \in \mathbb{N}$ using at least $t = t(\lambda) := 36\lambda d^3/\delta$ copies of a d-dimensional density matrix ρ, the process $\mathsf{Tomography}(\rho^{\otimes t})$ runs in polynomial time with respect to λ, d, $1/\delta$ and outputs a matrix $M \in \mathbb{C}^{d\times d}$ such that*

$$\Pr\left[\|\rho - M\| \leq \delta : M \leftarrow \mathsf{Tomography}(\rho^{\otimes t})\right] \geq 1 - \mathsf{negl}(\lambda).$$

We now recall how $\mathsf{Tomography}$ is used to extract pseudorandom strings in [2].

Construction 1 ($\mathsf{Extract}$ [2]). *Let $\lambda \in \mathbb{N}$ be the security parameter. The algorithm $\mathsf{Extract}$ is defined as follows:*

- *Input: $t := 144\lambda d^8$ copies of a d-dimensional quantum state ρ.*
- *Perform $\mathsf{Tomography}(\rho^{\otimes t})$ with error tolerance $\delta := d^{-5/6}$ to obtain a classical matrix $M \in \mathbb{C}^{d\times d}$.*
- *Run $\mathsf{Round}(M)$ to get $y \in \{0,1\}^\ell$. Output y.*

$\mathsf{Round}(M)$: Input: Matrix $M \in \mathbb{C}^{d\times d}$.

- Define $k := d^{5/6}$, $r := d^{2/3}$, and $\ell := d^{1/6}$.
- Let $p_1, \ldots, p_d$ be the diagonal entries of M.
- For $i \in [\ell]$:
  1. Let $q_i := \sum_{j=1}^{r} p_{(i-1)r+j}$.
  2. Define $b_i := \begin{cases} 1 & \text{if } q_i > r/d \\ 0 & \text{if } q_i \leq r/d. \end{cases}$
- Output $b_1 \| \ldots \| b_\ell$.

This extraction procedure was shown to satisfy the following two lemmas.

Lemma 3 (Lemma 3.6 in [2]). *If $\mathsf{Extract}$ is run on a Haar random state, then $d_{\mathsf{TD}}((q_1, \ldots, q_\ell), Z/(2d)) \leq O(k/d) + O(\ell/\sqrt{r})$ where Z is a random variable in $\mathbb{R}^\ell$ with i.i.d. $\mathcal{N}(2r, 4r)$ entries. In other words, $Z/(2d)$ has an i.i.d. $\mathcal{N}(r/d, r/d^2)$ entries.*

Lemma 4 (Claim 3.7 in [2]). *Define*

$$\mathcal{G}_d \coloneqq \left\{ |\psi\rangle \in \mathcal{S}(\mathbb{C}^d) : \forall i \in [\ell], \left| q_i - \frac{r}{d} \right| > 2/d \right\}.$$

Then,

1. $\Pr\left[|\psi\rangle \in \mathcal{G}_d : |\psi\rangle \leftarrow \mathsf{Haar}(\mathbb{C}^d)\right] \geq 1 - O(d^{-1/6})$.
2. *There exists a negligible function* $\mathsf{negl}(n)$ *such that for any* $|\psi\rangle \in \mathcal{G}_d$*, there exists a string* y *such that* $\Pr\left[y \leftarrow \mathsf{Extract}\left(|\psi\rangle^{\otimes t}\right)\right] \geq 1 - \mathsf{negl}(\lambda)$.

2.5 Pseudodeterministic Primitives in MicroCrypt

The Extract procedure described in the previous section was used to construct a QPT algorithm termed pseudodeterministic PRGs from a SPRS. The non-determinism in these algorithms make their use in cryptography difficult. Hence, the follow-up work [4] converted pseudodeterministic PRGs into a notion known as $\perp$-PRGs.

We introduce $\perp$-PRGs now. Note that it is easy to distinguish $\perp$ evaluations from random. However, it is sufficient to only require indistinguishability for non-$\perp$ evaluations. This is incorporated in the security game by providing $\perp$ in the truly random case as well. See [4] for a more in-depth discussion.

Definition 10 (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 11 ($\perp$-Pseudorandom Generator). *Let* $\lambda \in \mathbb{N}$ *be the security parameter and let* $m = m(\lambda)$ *be polynomial in* λ*. A QPT algorithm* G *mapping* $\{0,1\}^\lambda$ *to* $\{0,1\}^m \cup \{\perp\}$*, is a* (μ, m)-$\perp$-pseudodeterministic pseudorandom generator ($\perp$-PRG) *if:*

1. (Expansion) $m(\lambda) > \lambda$ *for all* $\lambda \in \mathbb{N}$.
2. (Pseudodeterminism) *There exist a constant* $c > 0$ *such that* $\mu(\lambda) = O(\lambda^{-c})$ *and for sufficiently large* $\lambda \in \mathbb{N}$ *there exists a set* $\mathcal{G}_\lambda \subseteq \{0,1\}^\lambda$ *such that the following holds:*
   (a)
   $$\Pr_{x \leftarrow \{0,1\}^\lambda}[x \in \mathcal{G}_\lambda] \geq 1 - \mu(\lambda).$$
   (b) For every $x \in \mathcal{G}_\lambda$ *there exists a non-*$\perp$ *value* $y \in \{0,1\}^m$ *such that:*
   $$\Pr[G(x) = y] \geq 1 - \mathsf{negl}(\lambda). \tag{1}$$
   (c) For every $x \in \{0,1\}^\lambda$*, there exists a non-*$\perp$ *value* $y \in \{0,1\}^m$ *such that:*
   $$\Pr[G(x) \in \{y, \perp\}] \geq 1 - \mathsf{negl}(\lambda). \tag{2}$$

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

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

$\bot$-PRGs were constructed from SPRSs in [4].

Lemma 5 (Corollary 1 [4]). *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 Definitions: Cryptography with Quantum Input Sampling

We present definitions for various known cryptographic primitives but with quantum input sampling algorithms. First of all, we define PRS with quantum input sampling.

Definition 12 (Pseudorandom State Generator with Quantum Input Sampling). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ and $m = m(\lambda)$ be polynomials in λ. A pair of QPT algorithms $(\mathsf{QSamp}, \mathsf{StateGen})$ is called a (m, n)-PRS with quantum input sampling ($\mathsf{PRS}^{\mathrm{qs}}$), if the following conditions hold:*

- $\mathsf{QSamp}(1^\lambda)$: *Outputs a string $k \in \{0,1\}^m$.*
- $\mathsf{StateGen}(k)$: *Takes a m-bit string k and outputs a n-qubit pure-state.*
- (Determinism) *For every $k \in \{0,1\}^m$, there exists a n-qubit state $|\psi_k\rangle$, such that the following condition is satisfied over the distribution of inputs:*

$$\Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)}[\mathsf{StateGen}(k) = |\psi_k\rangle] \geq 1 - \mathsf{negl}(\lambda).$$

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

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

In the case where security only holds for $t \leq q$ for some polynomial $q = q(\lambda)$, then we call this (q, m, n)-bounded-copy $\mathsf{PRS}^{\mathrm{qs}}$ ($\mathsf{BC\text{-}PRS}^{\mathrm{qs}}$). If $n = c \cdot \log(\lambda)$ with $c > 1$, then we call this SPRS^{qs} and if $n = \Omega(\lambda)$, then we call this LPRS^{qs}.

We now introduce PRG with quantum input sampling and PRF with quantum key generation. As far as we know, these notions has not been defined previously.

Definition 13 (Pseudorandom Generator with Quantum Input Sampling). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ and $m = m(\lambda)$ be polynomials in λ. A pair of QPT algorithms (QSamp, G) is a (m,n)-*PRG with quantum input sampling ($\mathsf{PRG}^{\mathrm{qs}}$) *if:*

1. $\mathsf{QSamp}(1^\lambda)$: *Outputs a string $k \in \{0,1\}^m$.*
2. *$G(k)$: Takes an input $k \in \{0,1\}^m$ and outputs $y \in \{0,1\}^n$*
3. (Expansion) *$m(\lambda) < n(\lambda)$ for all $\lambda \in \mathbb{N}$.*
4. (Determinism) *For every $k \in \{0,1\}^m$, there exists a string $y_k \in \{0,1\}^n$, such that the following condition is satisfied over the distribution of inputs:*

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

5. (Security) *For any QPT distinguisher $\mathcal{A}$, there exists a negligible function ϵ such that:*

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

We say that G is a PRG^{qs} if for every QPT $\mathcal{A}$, ϵ is a negligible function. We say that G is a weak PRG^{qs} *if for every QPT $\mathcal{A}$, $\epsilon \leq \frac{1}{p}$ for some polynomial p.*

Definition 14 (Pseudorandom Function with Quantum Key Generation). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ and $m = m(\lambda)$ be polynomials in λ. A pair of QPT algorithms (QSamp, F) is called a (m,n)-*PRF with quantum key generation ($\mathsf{PRF}^{\mathrm{qs}}$)*, if:*

1. $\mathsf{QSamp}(1^\lambda)$: *Outputs a key $k \in \{0,1\}^m$.*
2. *$F_k(x)$: Takes a key $k \in \{0,1\}^m$ and an input $x \in \{0,1\}^n$ and outputs a string $y \in \{0,1\}^n$* [10].
3. (Determinism) *For every $k \in \{0,1\}^m$ and $x \in \{0,1\}^n$, there exists a string $y_{k,x} \in \{0,1\}^n$ such that for all $x \in \{0,1\}^n$, the following is satisfied over the distribution of keys:*

$$\Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)}[F_k(x) = y_{k,x}] \geq 1 - \mathsf{negl}(\lambda).$$

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

$$\left| \Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)}\left[\mathcal{A}^{F_k}(1^\lambda) = 1\right] - \Pr_{O \leftarrow \mathit{\Pi}_{n,n}}\left[\mathcal{A}^{O}(1^\lambda) = 1\right] \right| \leq \mathsf{negl}(\lambda).$$

We say a PRF^{qs} is quantum-query-secure *if the above holds even if $\mathcal{A}$ is given quantum-query access to F_k and O. Furthermore, in the case where security only holds for $t \leq q$ queries for some polynomial $q = q(\lambda)$, then we call this a q-query PRF^{qs}.*

[10] For simplicity, we only consider $\mathsf{PRF}^{\mathrm{qs}}$s with the same input and output lengths. All our results easily generalize to $\mathsf{PRF}^{\mathrm{qs}}$s with different input and output lengths.

4 Relations among Primitives with Quantum Input Sampling

In this section, we explore relations among MicroCrypt primitives with quantum input sampling algorithms. In summary, we find that there exists black-box constructions for the following:

1. $\mathsf{PRG}^{\mathsf{qs}}$ from $\perp$-PRG.
2. $\mathsf{PRG}^{\mathsf{qs}}$ from $\mathsf{BC\text{-}SPRS}^{\mathsf{qs}}$.
3. $\mathsf{SPRS}^{\mathsf{qs}}$ from $\mathsf{PRG}^{\mathsf{qs}}$.
4. $\mathsf{BQ\text{-}PRU}^{\mathsf{qs}}$ from $\mathsf{PRG}^{\mathsf{qs}}$ [11].

We also discuss how these results can be used to modify the output size of a SPRS in Section 4.3.

4.1 $\mathsf{PRG}^{\mathsf{qs}}$ from $\perp$-PRG

We show how to build $\mathsf{PRG}^{\mathsf{qs}}$s from $\perp$-PRGs. First, we build a weak $\mathsf{PRG}^{\mathsf{qs}}$, and then amplify security to achieve a standard $\mathsf{PRG}^{\mathsf{qs}}$.

Construction 2. *Let m be a polynomial on the security parameter $\lambda \in \mathbb{N}$ such that $m > \lambda$. Let $\mu = O(\lambda^{-c})$ for some constant $c > 0$. Let G be a (μ, m)-$\perp$-PRG. The construction for a weak (λ, m)-PRG^{qs} is as follows:*

- $\mathsf{QSamp}(1^\lambda)$: *For $i \in [\lambda]$:*
  1. *Sample $k_i \leftarrow \{0,1\}^\lambda$.*
  2. *For each $j \in [\lambda]$, compute $y_{i,j} \leftarrow G(k_i)$.*
  3. *If $\mathsf{vote}_\lambda(y_{i,1}, \ldots, y_{i,\lambda}) \neq \perp$ [12], then output k_i.*

  Otherwise, output $\perp$.
- $\mathsf{PRG}(k)$:
  1. *If $k = \perp$, output 0^m.*
  2. *For each $j \in [\lambda]$, compute $y_j \leftarrow G(k)$.*
  3. *If $y_j = \perp$ for all $j \in [\lambda]$, then output $\perp$.*
  4. *Otherwise, output the first most common non-$\perp$ value in $(y_1, \ldots, y_\lambda)$.*

Lemma 6. *Construction 2 is a weak (λ, m)-PRG^{qs} assuming the existence of a (μ, m)-$\perp$-PRG.*

Proof. We first show that the algorithms $(\mathsf{QSamp}, \mathsf{PRG})$ satisfy the determinism condition of $\mathsf{PRG}^{\mathsf{qs}}$s. By the pseudodeterminism condition of G, it is clear that there is negligible probability that $\mathsf{QSamp}(1^\lambda)$ outputs $\perp$.

For any $k \in \{0,1\}^\lambda$, if $\Pr[G(k) = \perp] \geq \frac{2}{3}$, then there is negligible probability that $\mathsf{vote}_\lambda(y_1, \ldots, y_\lambda) \neq \perp$, where $y_j \leftarrow G(k)$ for all $j \in [\lambda]$. By the union bound, there is negligible probability that $\mathsf{QSamp}(1^\lambda)$ outputs a string k such

[11] This part is shown in Appendix A. See Definition 16 for the definition of $\mathsf{BQ\text{-}PRU}^{\mathsf{qs}}$s.
[12] $\mathsf{vote}(a_1, \ldots, a_\lambda)$ outputs the first most common element in the tuple $(a_1, \ldots, a_\lambda)$.

that $\Pr[G(k) = \bot] \geq \frac{2}{3}$. Therefore, except with negligible probability, the output of $\mathsf{QSamp}(1^\lambda)$ is a non-$\bot$ string k such that $\Pr[G(k) = \bot] < \frac{2}{3}$.

By the pseudodeterminism of G, for any $k \in \{0,1\}^\lambda$, there exists an output $y_k \in \{0,1\}^m$ such that $\Pr[G(k) \in \{y_k, \bot\}] \geq 1 - \mathsf{negl}(\lambda)$. Therefore, if $\Pr[G(k) = \bot] < \frac{2}{3}$, then $\Pr[G(k) = y_k] > \frac{1}{3} - \mathsf{negl}(\lambda)$. Hence, it is clear that for $k \leftarrow \mathsf{QSamp}(1^\lambda)$, $\mathsf{PRG}(k)$ outputs y_k, except with negligible probability. To sum up,

$$\Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)}[\mathsf{PRG}(k) = y_k] \geq 1 - \mathsf{negl}(\lambda).$$

Next, we need to show that the security condition is satisfied. In other words, we need to show that for any QPT distinguisher $\mathcal{A}$

$$\left| \Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)}[\mathcal{A}(\mathsf{PRG}(k)) = 1] - \Pr_{y \leftarrow \{0,1\}^m}[\mathcal{A}(y) = 1] \right| \leq 1/\mathsf{poly}(\lambda).$$

We commence with a hybrid argument.

- Hybrid H_0: This is the output distribution of the generator.
  - $k \leftarrow \mathsf{QSamp}(1^\lambda)$.
  - For each $j \in [\lambda]$, compute $y_j \leftarrow G(k)$.
  - If $y_j = \bot$ for all $j \in [\lambda]$, then output $\bot$.
  - Otherwise, output the first most common non-$\bot$ value in $(y_1, \ldots, y_\lambda)$.
- Hybrid H_1: The same as hybrid H_0 except the input is sampled from the good set $\mathcal{G}_\lambda$ of G.
  - $k \leftarrow \mathcal{G}_\lambda$.
  - For each $j \in [\lambda]$, compute $y_j \leftarrow G(k)$.
  - If $y_j = \bot$ for all $j \in [\lambda]$, then output $\bot$.
  - Otherwise, output the first most common non-$\bot$ value in $(y_1, \ldots, y_\lambda)$.
- Hybrid H_2: The same as hybrid H_1 except the output is computed using a single evaluation of $G(k)$.
  - $k \leftarrow \mathcal{G}_\lambda$.
  - Output $G(k)$.
- Hybrid H_3: The same as hybrid H_2 except output is a $\bot$ random string.
  - $k \leftarrow \mathcal{G}_\lambda$.
  - $y \leftarrow \{0,1\}^m$.
  - Output $\mathsf{Is}\text{-}\bot(G(k), y)$.
- Hybrid H_4: The output is a random string.
  - $y \leftarrow \{0,1\}^m$.
  - Output y.

We now show that no QPT adversary can distinguish these hybrids, except with inverse polynomial advantage.

Claim 1. *For any QPT adversary $\mathcal{A}$:*

$$\left| \Pr_{y \leftarrow H_0}[\mathcal{A}(y) = 1] - \Pr_{y \leftarrow H_1}[\mathcal{A}(y) = 1] \right| \leq \mu + \mathsf{negl}(\lambda).$$

Proof. Recall the algorithm of $\mathsf{QSamp}(1^\lambda)$ is as follows:

$\mathsf{QSamp}(1^\lambda)$: For $i \in [\lambda]$:

1. Sample $k_i \leftarrow \{0,1\}^\lambda$.
2. For each $j \in [\lambda]$, compute $y_{i,j} \leftarrow G(k_i)$
3. If $\mathsf{vote}_\lambda(y_{i,1}, \ldots, y_{i,\lambda}) \neq \perp$, then output k_i.

Otherwise, output $\perp$.

Note that if $k_1 \in \mathcal{G}_\lambda$ in the algorithm of QSamp, then the output is k_1, except with negligible probability, by the correctness condition of $\perp$-PRG. In other words, if $k_1 \in \mathcal{G}_\lambda$, then the output of QSamp is statistically indistinguishable from sampling a random element from $\mathcal{G}_\lambda$. Note that $\Pr_{k \leftarrow \{0,1\}^\lambda}[k \in \mathcal{G}_\lambda] \geq 1 - \mu$, hence the probability $k_1 \notin \mathcal{G}_\lambda$ is at most μ. Therefore, we have:

$$\left| \Pr_{y \leftarrow \mathsf{H}_0}[\mathcal{A}(y) = 1] - \Pr_{y \leftarrow \mathsf{H}_1}[\mathcal{A}(y) = 1] \right| \leq \mu + \mathsf{negl}(\lambda).$$

□

Claim 2. *Hybrids H_1 and H_2 are statistically indistinguishable.*

Proof. In both hybrids, the first step is to sample a input from the good set. By the pseudodeterminism of $\perp$-PRGs, for any input $k \in \mathcal{G}_\lambda$, there is a string y satisfying $\Pr[y \leftarrow G(k)] \geq 1 - \mathsf{negl}(\lambda)$. In this case, the probability that $\mathsf{vote}_\lambda(y_1, \ldots, y_\lambda) = y$, where $y_i \leftarrow G(k)$ for $i \in [\lambda]$, is at least $1 - \mathsf{negl}(\lambda)$. Hence, both hybrids are statistically indistinguishable. □

Claim 3. *For any QPT adversary $\mathcal{A}$:*

$$\left| \Pr_{y \leftarrow \mathsf{H}_2}[\mathcal{A}(y) = 1] - \Pr_{y \leftarrow \mathsf{H}_3}[\mathcal{A}(y) = 1] \right| \leq 2\mu + \mathsf{negl}(\lambda).$$

Proof. By the pseudodeterminism property of G, $\Pr_{k \leftarrow \{0,1\}^\lambda}[k \in \mathcal{G}_\lambda] \geq 1 - \mu$ so

$$\left| \Pr_{y \leftarrow \mathsf{H}_2}[\mathcal{A}(y) = 1] - \Pr_{k \leftarrow \{0,1\}^\lambda}[\mathcal{A}(G(k)) = 1] \right| \leq \mu + \mathsf{negl}(\lambda).$$

Furthermore, by the security of G, there is negligible function such that

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

Finally, by the pseudodeterminism property of G,

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^\lambda \\ y \leftarrow \{0,1\}^m}}[\mathcal{A}(\mathsf{ls}\text{-}\perp(y, G(k))) = 1] - \Pr_{\substack{k \leftarrow \mathcal{G}_\lambda \\ y \leftarrow \{0,1\}^m}}[\mathcal{A}(\mathsf{ls}\text{-}\perp(y, G(k))) = 1] \right| \leq \mu + \mathsf{negl}(\lambda).$$

The second term on the left hand side of the equation above is hybrid H_3. All in all, the triangle inequality gives:

$$\left| \Pr_{y \leftarrow \mathsf{H}_2} [\mathcal{A}(y) = 1] - \Pr_{y \leftarrow \mathsf{H}_3} [\mathcal{A}(y) = 1] \right| \leq 2\mu + \mathsf{negl}(\lambda).$$

□

Claim 4. *Hybrid H_3 is statistically indistinguishable from H_4.*

Proof. This follows directly from the pseudodeterminism property of $\perp$-PRGs. □

By the triangle inequality, we get that for any QPT adversary $\mathcal{A}$ such that,

$$\left| \Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)} [\mathcal{A}(\mathsf{PRG}(k)) = 1] - \Pr_{y \leftarrow \{0,1\}^m} [\mathcal{A}(y) = 1] \right| =$$
$$\left| \Pr_{y \leftarrow \mathsf{H}_0} [\mathcal{A}(y) = 1] - \Pr_{y \leftarrow \mathsf{H}_4} [\mathcal{A}(y) = 1] \right| \leq 3\mu + \mathsf{negl}(\lambda).$$

Hence, Construction 2 is a weak $\mathsf{PRG}^{\mathrm{qs}}$. □

Next, it was shown in [17] that a weak PRG can be upgraded to a strong PRG through a standard amplification argument, increasing the input length from λ to λ^2. This argument applies to $\mathsf{PRG}^{\mathrm{qs}}$s, giving the following result.

Theorem 6. *If there exists (μ, m)-$\perp$-PRG where $m > \lambda^2$ and $\mu = O(\lambda^{-c})$ for some constant $c > 0$, then there exists a (λ^2, m)-PRG^{qs} satisfying strong security.*

4.2 $\mathsf{PRG}^{\mathrm{qs}}$ from $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$

In this part, we show that $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$ can be used to construct $\mathsf{PRG}^{\mathrm{qs}}$s using properties of pseudorandom states described in Section 2.4.

Construction 3. *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = c \cdot \log \lambda$, $d = \lceil \lambda^c \rceil$, and $m \coloneqq \lceil \lambda^{c/12} \rceil$, where $c > 24$ is a constant. Let $(\mathsf{QS}, \mathsf{PRS})$ be a $(500\lambda d^8, \lambda, n)$-$\mathsf{BC\text{-}SPRS}^{qs}$. The construction for a weak (λ, m)-PRG^{qs} is as follows:*

- $\mathsf{QSamp}(1^\lambda)$: *For each $i \in [\lambda]$:*
  1. *Sample $s_i \leftarrow \mathsf{QS}(1^\lambda)$.*
  2. *Extract $t \coloneqq 144\lambda d^8$ copies of pseudorandom state $\rho_i^{\otimes t}$ using $\mathsf{PRS}(s_i)$.*
  3. *Run $M_i \leftarrow \mathsf{Tomography}(\rho_i^{\otimes t})$.*
  4. *If $M_i \in \mathcal{G}_\lambda$ (as defined in Lemma 4), then output $k \coloneqq s_i$.*

  Otherwise, output $\perp$.
- $G(k)$:
  1. *If $k = \perp$, then output $\perp$.*
  2. *Extract $t \coloneqq 144\lambda d^8$ copies of pseudorandom state $\rho^{\otimes t}$ using $\mathsf{PRS}(k)$.*
  3. *Compute $y \leftarrow \mathsf{Extract}(\rho^{\otimes t})$.*

4. Output y.

Lemma 7. *Construction 3 is a weak (λ, m)-PRG^{qs} assuming the existence of a $(500\lambda d^8, \lambda, n)$-$\mathsf{BC\text{-}SPRS}^{qs}$.*

Proof. We first show that the determinism condition of a $\mathsf{PRG}^{\mathrm{qs}}$ is satisfied (see Definition 13).

Assume for contradiction that the determinism condition is not satisfied. Then, there is a non-negligible probability such that sampling a key $k \leftarrow \mathsf{QSamp}(1^\lambda)$ and evaluating $G(k)$ twice yields distinct values. We obtain a contradiction as follows.

By Lemma 4, there is negligible probability that $\mathsf{QSamp}(1^\lambda)$ outputs $\perp$. The other possibility is that $\mathsf{QSamp}(1^\lambda)$ outputs a key $k = s$ satisfying $M_s \in \mathcal{G}_\lambda$ where $M_s \leftarrow \mathsf{Tomography}(\rho^{\otimes t})$ and $\rho \leftarrow \mathsf{PRS}(s)$. Therefore, with non-negligible probability, $k = s$ satisfies $M_s \in \mathcal{G}_\lambda$ but two evaluations of $G(k)$ yield distinct values.

Then, we construct a distinguisher $\mathcal{D}$ that breaks the security of $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$ as follows. $\mathcal{D}$ receives $500\lambda d^8 > 3t$ copies of a state ρ and needs to distinguish whether $\rho \leftarrow \mathsf{PRS}(k)$ for $k \leftarrow \mathsf{QS}(1^\lambda)$ or $\rho \leftarrow \mathsf{Haar}(\mathbb{C}^n)$ is a Haar-random state. $\mathcal{D}$ computes $M \leftarrow \mathsf{Tomography}(\rho^{\otimes t})$ with the first t copies, $y_1 \leftarrow \mathsf{Extract}(\rho^{\otimes t})$ with the second t copies, and $y_2 \leftarrow \mathsf{Extract}(\rho^{\otimes t})$ with the last t copies. If $y_1 \neq y_2$ and $M \in \mathcal{G}_\lambda$, then $\mathcal{D}$ guesses that ρ is a $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$ i.e. outputs 1. Otherwise, $\mathcal{D}$ outputs 0. By our assumption, $y_1 \neq y_2$ and $M \in \mathcal{G}_\lambda$ occurs with non-negligible probability if ρ is generated using $(\mathsf{QS}, \mathsf{PRS})$. On the other hand, Lemma 4 states that this occurs with negligible probability when ρ is a Haar-random state. It is clear that $\mathcal{D}$ has non-negligible advantage in distinguishing $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$ from Haar-random states using only $3t$ copies of the state, which contradicts the security condition of $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$. Therefore, (QSamp, G) satisfies the correctness condition of a $\mathsf{PRG}^{\mathrm{qs}}$.

Next, we need to show security. Specifically, we need to show that for any QPT distinguisher $\mathcal{A}$:

$$\left| \Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)} [\mathcal{A}(G(k)) = 1] - \Pr_{y \leftarrow \{0,1\}^m} [\mathcal{A}(y) = 1] \right| \leq O(1/d^{1/6}).$$

The proof is through a hybrid argument:

- H_0:
  1. Sample $k \leftarrow \mathsf{QSamp}(1^\lambda)$.
  2. Compute $y \leftarrow G(k)$.
  3. Run $\mathcal{A}(y)$ and output the result.
- H_1: The same as H_0, except we modify the $\mathsf{PRG}^{\mathrm{qs}}$ algorithms by extracting a classical string from the pseudorandom states during input sampling instead of during evaluation.
  - $\mathsf{QSamp}_{\mathsf{H}_1}(1^\lambda)$: For each $i \in [\lambda]$:
    1. Sample $s_i \leftarrow \mathsf{QS}(1^\lambda)$.
    2. Extract $t := 144\lambda d^8$ copies of pseudorandom state $\rho_i^{\otimes t}$ using $\mathsf{PRS}(s_i)$.

3. Run $M_i \leftarrow \mathsf{Tomography}(\rho_i^{\otimes t})$.
4. If $M_i \in \mathcal{G}_\lambda$, then compute $y_i \leftarrow \mathsf{Round}(M_i)$.
5. Output $k := y_i$.

Otherwise, output $\perp$.

- $G_{\mathsf{H}_1}(k)$: Output k.

– H_2: Same as H_1, except we modify the $\mathsf{PRG}^{\mathrm{qs}}$ algorithms by using random Haar states instead of pseudorandom states.

- $\mathsf{QSamp}_{\mathsf{H}_2}(1^\lambda)$: For each $i \in [\lambda]$:
  1. Sample $t := 144\lambda d^8$ copies of a random Haar state $\rho_i^{\otimes t}$ from $\mathsf{Haar}(\mathbb{C}^d)$.
  2. Run $M_i \leftarrow \mathsf{Tomography}(\rho_i^{\otimes t})$.
  3. If $M_i \in \mathcal{G}_\lambda$, then compute $y_i \leftarrow \mathsf{Round}(M_i)$.
  4. Output $k := y_i$.

  Otherwise, output $\perp$.
- $G_{\mathsf{H}_2}(k)$: Output k.

– H_3: Same as H_2, except we modify the $\mathsf{PRG}^{\mathrm{qs}}$ algorithms by removing the condition on the random Haar states.

- $\mathsf{QSamp}_{\mathsf{H}_3}(1^\lambda)$:
  1. Sample $t := 144\lambda d^8$ copies of a random Haar state $\rho^{\otimes t}$ from $\mathsf{Haar}(\mathbb{C}^d)$.
  2. Run $M \leftarrow \mathsf{Tomography}(\rho^{\otimes t})$.
  3. Compute $y \leftarrow \mathsf{Round}(M)$.
  4. Output $k := y$.
- $G_{\mathsf{H}_3}(k)$: Output k.

– H_4: Same as H_3, except we sample a random string instead of extracting randomness from the Haar random states.

- $\mathsf{QSamp}_{\mathsf{H}_4}(1^\lambda)$: Sample a random string $y \leftarrow \{0,1\}^m$. Set $k = y$.
- $G_{\mathsf{H}_4}(k)$: Output k.

First of all, hybrid H_0 is statistically indistinguishable from hybrid H_1, since we just move a step from the evaluation phase to the input sampling phase.

H_1 is computationally indistinguishable from hybrid H_2 because any QPT adversary that can distinguish between these hybrids can be used to break $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$ security.

Next, if $M \in \mathcal{G}_\lambda$ in $\mathsf{QSamp}_{\mathsf{H}_3}$ in hybrid H_3 and $M_1 \in \mathcal{G}_\lambda$ in $\mathsf{QSamp}_{\mathsf{H}_2}(1^\lambda)$ in H_2, then it is clear that H_2 and H_3 are statistically indistinguishable. This scenario occurs with $O(d^{-1/6})$ probability by Lemma 4.

Finally, the variational distance between hybrids H_3 and H_4 is at most $O(d^{-1/6})$ by Lemma 3.

All in all, by the triangle inequality:

$$\begin{aligned}
&\left| \Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)} [\mathcal{A}(G(k)) = 1] - \Pr_{y \leftarrow \{0,1\}^m} [\mathcal{A}(y) = 1] \right| \\
&= \left| \Pr_{y \leftarrow \mathsf{H}_0} [\mathcal{A}(y) = 1] - \Pr_{y \leftarrow \mathsf{H}_4} [\mathcal{A}(y) = 1] \right| \\
&\leq O(d^{-1/6}).
\end{aligned}$$

□

Recall that weak $\mathsf{PRG}^{\mathrm{qs}}$ can be upgraded to a strong $\mathsf{PRG}^{\mathrm{qs}}$ following the same argument in [17]. Hence, we obtain the following result.

Theorem 7. *If there exists* $(500\lambda d, \lambda, n)$-$\mathsf{BC}$-$\mathsf{SPRS}^{qs}$*, then there exists a* (λ^2, m)-PRG^{qs} *satisfying strong security.*

4.3 $\mathsf{SPRS}^{\mathrm{qs}}$ from $\mathsf{PRG}^{\mathrm{qs}}$

In this section, we show that $\mathsf{PRG}^{\mathrm{qs}}$s can be used to build $\mathsf{SPRS}^{\mathrm{qs}}$.

Construction 4. *Let* $\lambda \in \mathbb{N}$ *be the security parameter and let* $c > 12$ *be a constant. Let* $m = m(\lambda)$ *be polynomial on* λ *such that* $m > \lambda^{2c+1}$*. Let* (QS, G) *be a* (λ, m)-PRG^{qs}*. Let* $N = \lceil \lambda^c \rceil$ *and* $\mathcal{X} = \{1, \ldots, N\}$*. The construction for a* $(\lambda, c \cdot \log \lambda)$-$\mathsf{SPRS}^{qs}$ *is as follows:*

- $\mathsf{QSamp}(1^\lambda)$: *Sample* $s \leftarrow \mathsf{QS}(1^\lambda)$*. Output* $k = s$*.*
- $\mathsf{StateGen}(k)$:
  1. *Compute* $y \leftarrow G(k)$*.*
  2. *Interpret* y *as a function* $f_y : \mathcal{X} \to \mathcal{X}$ [13]*.*
  3. *Output*

$$|\psi_k\rangle \coloneqq \frac{1}{\sqrt{N}} \sum_{x \in \mathcal{X}} \omega_N^{f_y(x)} |x\rangle.$$

Theorem 8. *Construction 4 is a* $(\lambda, c \cdot \log \lambda)$-$\mathsf{SPRS}^{qs}$ *assuming the existence of a* (λ, m)-PRG^{qs} *where* $m > \lambda^{2c+1}$*.*

Theorem 8 follows directly from [23]. Specifically, [23] shows that a n-PRS can be built from a PRF with domain size 2^n. This conversion applies to the quantum input sampling regime as well. Then, Theorem 8 follows by setting $n \coloneqq c \cdot \log \lambda$ and noting that f_y in Construction 4 is computationally indistinguishable from a random function by the security of $\mathsf{PRG}^{\mathrm{qs}}$s.

Theorem 7 states that we can construct a $\mathsf{PRG}^{\mathrm{qs}}$ from any $\mathsf{SPRS}^{\mathrm{qs}}$. On the other hand, Theorem 8 state that we can construct a $\mathsf{SPRS}^{\mathrm{qs}}$ from a $\mathsf{PRG}^{\mathrm{qs}}$. Applying these two results consecutively, we get a method to shrink the size of a $\mathsf{SPRS}^{\mathrm{qs}}$.

Corollary 4. *For any constant* $c > 36$*, and any constant* $0 < m < c/36$*,* $(\lambda, c \log \lambda)$-$\mathsf{SPRS}^{qs}$ *implies* $(\lambda, m \log \lambda)$-$\mathsf{SPRS}^{qs}$*.*

Furthermore, by starting with a SPRS, building a $\perp$-PRG (Lemma 5), amplifying the output length sufficiently [14], building a $\mathsf{PRG}^{\mathrm{qs}}$ (Theorem 6), and finally a $\mathsf{SPRS}^{\mathrm{qs}}$ (Theorem 8), we obtain a $\mathsf{SPRS}^{\mathrm{qs}}$ of larger size.

Corollary 5. *For any constant* $c > 36$*, and any constant* $m > c$*,* $(\lambda, c \log \lambda)$-$\mathsf{SPRS}$ *implies* $(\lambda, m \log \lambda)$-$\mathsf{SPRS}^{qs}$*.*

We also show how to build bounded-query $\mathsf{PRU}^{\mathrm{qs}}$ from $\mathsf{PRG}^{\mathrm{qs}}$ in Appendix A.

[13] For $i \in \mathcal{X}$, $f_z(i) \coloneqq z[it : (i+1)t]$ where $t \coloneqq \lceil \log N \rceil$.

[14] It is easy to amplify the output length of a $\perp$-PRG by re-applying the algorithm on the output, at the cost of increasing the pseudodeterminism error.

5 Separations

5.1 Separating PRG from $\mathsf{PRF}^{\mathsf{qs}}$

We show a distinction between uniform input sampling and quantum input sampling by demonstrating that there do not exist fully black-box constructions of a PRG from a $\mathsf{PRF}^{\mathsf{qs}}$ with inverse access.

We only state the result here and give the proof in Section 5.1.1.

Theorem 9. *There does not exist a fully black-box construction of a* PRG *from a (quantum-query-secure)* PRF^{qs} *with inverse access.*

The following lemma generalizes the applications of PRFs in cryptography [23, 2, 4] to $\mathsf{PRF}^{\mathsf{qs}}$s. These follow in the same way as using quantum key generation rather than uniform key generation does not affect the proofs.

Lemma 8. *There exists fully black-box constructions of the following primitives from (quantum-query-secure)* PRF^{qs}*s:*

1. PRG^{qs}, SPRS^{qs}, LPRS^{qs}, *and* PRU^{qs}.
2. *Statistically-binding computationally hiding bit commitments with classical communication.*
3. *Message authentication codes of classical messages with classical communication.*
4. *CCA2-secure symmetric encryption for classical messages with classical keys and ciphertexts.*
5. *EV-OWPuzzs.*

As a result, of the applications of $\mathsf{PRF}^{\mathsf{qs}}$, we obtain the following corollary.

Corollary 6. *There is no fully black-box construction for a* PRG *from the following primitives with inverse access:*

1. PRG^{qs}, SPRS^{qs}, LPRS^{qs}, *and* PRU^{qs}.
2. *Statistically-binding computationally hiding bit commitments with classical communication.*
3. *Message authentication codes of classical messages with classical communication.*
4. *CCA2-secure symmetric encryption of classical messages with classical keys and ciphertexts.*
5. *EV-OWPuzzs.*

5.1.1 Separation Proof. The idea of the separation proof is to consider three oracles. The first is a PSPACE oracle. The second is a restricted-access random oracle $\mathcal{O}$, that can only be accessed with a key as input. This oracle will acts as a $\mathsf{PRF}^{\mathsf{qs}}$. The third oracle σ produces random keys such that each distinct key gives access to a different function in $\mathcal{O}$. Note that this is not an issue for the

$\mathsf{PRF}^{\mathrm{qs}}$, as a key from σ can be sampled during key generation and, then, reused during evaluation to obtain deterministic outputs from $\mathcal{O}$.

On the other hand, a uniform key generation algorithm is incapable of accessing σ during key generation. Consequently, a PRG is incapable of obtaining deterministic evaluations from $\mathcal{O}$. This informally means that any PRG cannot depend on $(\sigma, \mathcal{O})$ and, thus, cannot exist in the presence of a PSPACE oracle. This implies that there does not exist a fully black-box construction of a PRG from a $\mathsf{PRF}^{\mathrm{qs}}$.

We first introduce some preliminary information. We present s-PRG security relative to an oracle $\mathcal{T}$ in the form of an experiment to simplify notation later on.

$\mathsf{Exp}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}},G^{\mathcal{T}}}(1^\lambda)$:

1. Sample $k \leftarrow \{0,1\}^\lambda$ and $b \leftarrow \{0,1\}$.
2. If $b = 0$, generate $y \leftarrow G^{\mathcal{T}}(k)$. Else, sample $y \leftarrow \{0,1\}^s$.
3. $b' \leftarrow \mathcal{A}^{\mathcal{T}}(y)$.
4. If $b' = b$, output 1. Otherwise, output 0.

We define the advantage of the adversary in this experiment as follows. It is clear that PRG security implies that this advantage should be negligible.

$$\mathsf{Adtg}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}},G^{\mathcal{T}}}(1^\lambda) \coloneqq \Pr\left[\mathsf{Exp}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}},G^{\mathcal{T}}}(1^\lambda) = 1\right] - \frac{1}{2}.$$

Let $U_{|\psi\rangle}$ be a unitary that flips $|0^n\rangle$ with $|\psi\rangle$ and acts as the identity on all other orthogonal states. We will use the following result from [14] regarding the simulation of this oracle with polynomial copies of $|\psi\rangle$.

Theorem 10 (Theorem 2.6 in [14]). *Let $T, n \in \mathbb{N}$, $\epsilon > 0$, and $|\psi\rangle$ be a real-valued n-qubit state orthogonal to $|0^n\rangle$. Let $U_{|\psi\rangle}$ be the unitary defined above. For any oracle algorithm $\mathcal{A}$ making T queries to $U_{|\psi\rangle}$, there exists an algorithm $\tilde{\mathcal{A}}$ with access to $O(\frac{T^2}{\epsilon^2})$ copies of $|\psi\rangle$ that outputs a state that is ϵ-close in terms of trace distance.*

Note that Theorem 2.6 in [14] is more involved than Theorem 10 since it deals with *complex-valued* quantum states. Our version is limited to real-valued states.

Also, note that [38] shows that any computationally unbounded adversaries cannot distinguish quantum-query-access to a a randomly sampled polynomial of degree $(2r-1)$ and a random function given only r quantum queries. Given that this result holds for computationally unbounded adversaries, it holds for adversaries with access to a PSPACE-oracle. We now describe the oracles used in the separation.

Construction 5. *For $n \in \mathbb{N}$, let $O_n \leftarrow \Pi_{n,8n}$ and $P_n \leftarrow \Pi_{2n,n}$ be random functions. Let $\mathcal{T} \coloneqq (\sigma, \mathcal{O}, \mathcal{C})$ be a tuple of oracles, where $\sigma = \{\sigma_n\}_{n\in\mathbb{N}}$, $\mathcal{O} \coloneqq \{\mathcal{O}_n\}_{n\in\mathbb{N}}$, and $\mathcal{C} \coloneqq \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ are defined as follows:*

1. *$\mathcal{C}$ is for membership in a PSPACE-complete language.*
2. *σ_n: Unitary that swaps $|0^{9n+1}\rangle$ with the state $|\psi_n\rangle \coloneqq \frac{1}{\sqrt{2^n}}|1\rangle \sum_{x\in\{0,1\}^n} |x\rangle|O_n(x)\rangle$ and acts as the identity on all other orthogonal states.*
3. *$\mathcal{O}_n$: Unitary of the classical function that maps (x, y, a), where $x, a \in \{0,1\}^n$ and $y \in \{0,1\}^{8n}$, to $P_n(x, a)$ if $O_n(x) = y$ and to $\perp$ otherwise.*

Notice that the unitary oracles above are self-inverse, so our separation is relative to a unitary oracle with access to the inverse.

We introduce some notation for the proof. Let $\mathbb{T}$ denote the set of all possible oracles and let $\mathcal{T} \leftarrow \mathbb{T}$ denote sampling an oracle in the way described in Construction 5. For any oracle $\mathcal{T}$ and integer $m \in \mathbb{N}$, let $\mathcal{T}_{\leq m}$ denote the sequence of oracles $(\sigma_n, \mathcal{O}_n)_{n\leq m}$ and let $\mathbb{T}[\mathcal{T}_{\leq m}]$ denote the set

$$\mathbb{T}[\mathcal{T}_{\leq m}] \coloneqq \{\tilde{\mathcal{T}} \in \mathbb{T} : \tilde{\mathcal{T}}_{\leq m} = \mathcal{T}_{\leq m}\}.$$

Theorem 11. *There does not exist a fully black-box construction of a* PRG *from a (quantum-query-secure)* PRF^{qs} *with inverse access.*

Proof. For simplicity, we only prove the theorem for $(9n, n)$-$\mathsf{PRF}^{\mathrm{qs}}$ i.e. for $\mathsf{PRF}^{\mathrm{qs}}$s with $9n$-bit keys and n-bit inputs. However, the proof easily generalizes to other parameters by modifying the parameters of the oracles.

Assume, for the purpose of obtaining a contradiction, that $\tilde{G}^{F,F^{-1}}$ is a black-box construction of a PRG, from any $(9n, n)$-$\mathsf{PRF}^{\mathrm{qs}}$ F. First, the following result states that there exists a $\mathsf{PRF}^{\mathrm{qs}}$ relative to $\mathcal{T}$.

Lemma 9. *There exists a (quantum-query-secure)* $(9n, n)$-PRF^{qs} *relative to $\mathcal{T}$ for any O and with probability 1 over the distribution of P. Furthermore, correctness is satisfied for any oracle $\mathcal{T}$.*

Proof. **Construction 6.** *The algorithms of a* PRF^{qs} *with oracle access to $\mathcal{T}$ is as follows:*

- $\mathsf{QSamp}^{\mathcal{T}}(1^n)$:
  1. *Query $\sigma_n(|0^{9n+1}\rangle)$ and measure the result in the computational basis to obtain (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}^{8n}$.*
  2. *Output $k \coloneqq (x, y)$.*
- $F_k^{\mathcal{T}}(a)$: *Interpret k as (x, y). Output $\mathcal{O}_n(x, y, a)$* [15].

It is clear that $(\mathsf{QSamp}^{\mathcal{T}}, F^{\mathcal{T}})$ satisfies the correctness condition of $\mathsf{PRF}^{\mathrm{qs}}$ for any oracle $\mathcal{T}$.

For security, note that for any $(x, y) \leftarrow \mathsf{QSamp}^{\mathcal{T}}(1^n)$, $F^{\mathcal{T}}_{(x,y)}(\cdot) = P_n(x, \cdot)$, where $P_n(x, \cdot)$ is a random function independent of σ. Lemma 2.2 from [33] states that a random oracle acts as a PRF i.e. for any QPT adversary $\mathcal{A}$ that makes $p(n)$ oracle queries:

$$\mathop{\mathbb{E}}_{P_n\leftarrow\Pi_{2n,n}}\left[\left|\Pr_{x\leftarrow\{0,1\}^n}\left[\mathcal{A}^{P_n,P_n(x,\cdot)}(1^n)=1\right]-\Pr_{\tilde{P}_n\leftarrow\Pi_{n,n}}\left[\mathcal{A}^{P_n,\tilde{P}_n}(1^n)=1\right]\right|\right]\leq\frac{2p(n)}{2^n}<\frac{1}{2^{n/4}}.$$

[15] This is a slight abuse of notation, as $\mathcal{O}$ is a unitary but we interpret it as a classical function here.

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 QPT adversaries with access to a PSPACE-oracle:

$$\mathop{\mathbb{E}}_{P_n \leftarrow \Pi_{2n,n}} \left[\left| \Pr_{x \leftarrow \{0,1\}^n} \left[\mathcal{A}^{P_n, P_n(x,\cdot), \mathcal{C}}(1^n) = 1 \right] - \Pr_{\tilde{P}_n \leftarrow \Pi_{n,n}} \left[\mathcal{A}^{P_n, \tilde{P}_n, \mathcal{C}}(1^n) = 1 \right] \right| \right] \leq \frac{1}{2^{n/4}}.$$

By Markov inequality, we get that

$$\Pr_{P_n \leftarrow \Pi_{2n,n}} \left[\left| \Pr_{x \leftarrow \{0,1\}^n} \left[\mathcal{A}^{P_n, P_n(x,\cdot), \mathcal{C}}(1^n) = 1 \right] - \Pr_{\tilde{P}_n \leftarrow \Pi_{n,n}} \left[\mathcal{A}^{P_n, \tilde{P}_n, \mathcal{C}}(1^n) = 1 \right] \right| \geq 2^{-n/8} \right] \leq 2^{-n/8}$$

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

$$\left| \Pr_{x \leftarrow \{0,1\}^n} \left[\mathcal{A}^{P_n, P_n(x,\cdot), \mathcal{C}}(1^n) = 1 \right] - \Pr_{\tilde{P}_n \leftarrow \Pi_{n,n}} \left[\mathcal{A}^{P_n, \tilde{P}_n, \mathcal{C}}(1^n) = 1 \right] \right| \leq 2^{-n/8},$$

except for finitely many $n \in \mathbb{N}$. There are countable number of quantum algorithms $\mathcal{A}$ making polynomial queries to $\mathcal{T}$, so this bound holds for every such adversary. □

By Lemma 9 and the assumed existence of a black-box construction $\tilde{G}$, there exists an algorithm $G^{\mathcal{T}}$ that is a PRG with probability 1 over the oracles $\mathcal{T}$ and satisfies correctness for any oracle $\mathcal{T} \in \mathbb{T}$. Let $s = s(\lambda)$ be a polynomial denoting the output length of G.

Claim 5. *For any QPT adversary $\mathcal{A}$:*

$$\Pr_{\mathcal{T} \leftarrow \mathbb{T}} \left[\mathit{Adtg}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}}, G^{\mathcal{T}}}(1^\lambda) \leq O\left(\frac{1}{\lambda^4}\right) \right] \geq \frac{3}{4}.$$

Proof. If this does not hold, then there exists a QPT adversary $\mathcal{A}$ such that

$$\Pr_{\mathcal{T} \leftarrow \mathbb{T}} \left[\mathsf{Adtg}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}}, G^{\mathcal{T}}}(1^\lambda) > \frac{1}{\lambda^5} \right] > \frac{1}{4}$$

for infinitely many $\lambda \in \mathbb{N}$.

By a variant of Borel-Cantelli Lemma (Lemma 2.9 in [28]), this means $\mathcal{A}$ is successful in breaking the security of $G^{\mathcal{T}}$ with probability $\frac{1}{4}$ over the oracle distribution. Therefore, $G^{\mathcal{T}}$ is not a PRG with probability $\frac{1}{4}$ over the oracle distribution, giving a contradiction. □

Let $r = r(\lambda)$ denote the maximum run-time of G and $m = m(\lambda) \coloneqq 10(r\lambda)^4 + \lambda$. Hence, G makes at most r queries to the oracles.

Fix an oracle $\mathcal{T}$. We will need to show the following lemma.

Lemma 10. *For large enough λ and any $k \in \{0,1\}^\lambda$,*

$$\Pr_{\mathcal{T}',\mathcal{T}'' \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]} \left[G^{\mathcal{T}'}(k) = G^{\mathcal{T}''}(k) \right] \geq 1 - \frac{r}{m}.$$

Proof. Let $\mathcal{T}' \coloneqq (\sigma', \mathcal{O}', \mathcal{C})$ and $\mathcal{T}'' \coloneqq (\sigma'', \mathcal{O}'', \mathcal{C})$ be two oracles sampled from $\mathbb{T}[\mathcal{T}_{\leq \log(2m)}]$ and determined by the functions (P', O') and (P'', O''), respectively, as described in Construction 5. Fix $k \in \{0,1\}^\lambda$.

We will now describe how to construct a sequence of oracles $\mathcal{T}^i = (\sigma^i, \mathcal{O}^i, \mathcal{C})$, starting with $\mathcal{T}'$ and ending with $\mathcal{T}''$, such that $G^{\mathcal{T}^i}(k)$ is invariant (to some degree) as we move along the sequence. Note that $\mathcal{T}'_n$ and $\mathcal{T}''_n$ only differ for $\log(2m) < n < r$. For such a value of n, let $x^* \in \{0,1\}^n$ be an arbitrary string.

- $\mathcal{T}^1$: This is the same as $\mathcal{T}'$. Define

$$\left|\psi_n^1\right\rangle \coloneqq \frac{1}{\sqrt{2^n}} |1\rangle \sum_{x \in \{0,1\}^n} |x\rangle |O'_n(x)\rangle.$$

- $\mathcal{T}^2$: Same as $\mathcal{T}^1$, except we change σ_n^2 to be the unitary that swaps $|0^{9n+1}\rangle$ with

$$\left|\psi_n^2\right\rangle \coloneqq \frac{1}{\sqrt{2^n - 1}} |1\rangle \sum_{x \in \{0,1\}^n \setminus \{x^*\}} |x\rangle |O'_n(x)\rangle.$$

- $\mathcal{T}^3$: Same as $\mathcal{T}^2$, except for any $a \in \{0,1\}^n$, we set $\mathcal{O}_n^3(x^*, O'_n(x^*), a)$ to $\perp$ and, then, we set $\mathcal{O}_n^3(x^*, O''_n(x^*), a)$ to $P''_n(x^*, a)$.
- $\mathcal{T}^4$: Same as $\mathcal{T}^3$, except σ_n^4 is the unitary that swaps $|0^{9n+1}\rangle$ with

$$\left|\psi_n^4\right\rangle \coloneqq \frac{1}{\sqrt{2^n}} \left(|1\rangle \sum_{x \in \{0,1\}^n \setminus \{x^*\}} |x\rangle |O_n(x)\rangle + |1\rangle |x^*\rangle |O''_n(x^*)\rangle \right).$$

We perform these modifications for every input x^* of length n for every $\log(2m) < n < r$, one input at a time, until the oracle $\mathcal{T}'$ is completely replaced with $\mathcal{T}''$ on inputs of length less than r. What remains to show is that G is invariant to some degree under these modifications.

Claim 6. *For large enough λ,*

$$d_{TD}\left(G^{\mathcal{T}^1}(k), G^{\mathcal{T}^2}(k)\right) \leq \frac{3}{\lambda}.$$

Proof. In the oracles $\mathcal{T}^1$ and $\mathcal{T}^2$, only σ_n^1 and σ_n^2 differ. Both these oracles are unitaries that swap two states and, thus, have the structure required to apply Theorem 10. Specifically, setting $T = r$ and $\epsilon = \lambda$ in Theorem 10, we get that there exists an algorithm $\tilde{G}^{\mathcal{T}^1 \setminus \sigma_n^1}(k, |\psi_n^1\rangle^{\otimes q})$ that simulates $G^{\mathcal{T}^1}(k)$ without oracle access to σ_n^1, given $q \coloneqq O(\frac{T^2}{\epsilon^2}) = O(r^2\lambda^2)$ copies of $|\psi_n^1\rangle$, yielding an output that is $(\frac{1}{\lambda})$-close in trace distance. Similarly, $\tilde{G}^{\mathcal{T}^2 \setminus \sigma_n^2}(k, |\psi_n^2\rangle^{\otimes q})$ simulates

$G^{\mathcal{T}^2}(k)$ without oracle access to σ_n^2, given q copies of $|\psi_n^2\rangle$, yielding an output that is $(\frac{1}{\lambda})$-close in trace distance.

Given that, for large enough λ,

$$d_{\mathsf{TD}}\left(\left|\psi_n^2\right\rangle^{\otimes q}, \left|\psi_n^1\right\rangle^{\otimes q}\right) \leq \frac{q}{m} < \frac{1}{\lambda},$$

and noting that $\mathcal{T}^1 \setminus \sigma_n^1$ is equivalent to $\mathcal{T}^2 \setminus \sigma_n^2$, we get

$$d_{\mathsf{TD}}\left(\tilde{G}^{\mathcal{T}^1 \setminus \sigma_n^1}\left(k, \left|\psi_n^1\right\rangle^{\otimes q}\right), \tilde{G}^{\mathcal{T}^2 \setminus \sigma_n^2}\left(k, \left|\psi_n^2\right\rangle^{\otimes q}\right)\right) \leq \frac{1}{\lambda}.$$

Therefore, by the triangle inequality, we have that for large enough λ,

$$d_{\mathsf{TD}}\left(G^{\mathcal{T}^1}(k), G^{\mathcal{T}^2}(k)\right) \leq \frac{3}{\lambda}.$$

□

Claim 7. *For large enough λ,*

$$\Pr_{\mathcal{T}', \mathcal{T}'' \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]}\left[d_{\mathsf{TD}}\left(G^{\mathcal{T}^2}(k), G^{\mathcal{T}^3}(k)\right) \geq \frac{1}{2^{2n}}\right] \leq 1/2^{2n}.$$

Proof. Notice $\sigma^2 = \sigma^3$ and $\mathcal{O}^2$ only differs from $\mathcal{O}^3$ on inputs starting with $(x^*, O'(x^*))$ or $(x^*, O''(x^*))$. Crucially, $O'(x^*)$ and $O''(x^*)$ are distributed uniformly at random and are of length $8n$. Therefore, to distinguish these two oracles, G needs to do unstructured search for an input starting with x^* that maps to a non-$\perp$ element.

The lower bound for unstructured search [36, 10] states that G cannot distinguish these two oracles with better than $O(\frac{r^2}{2^{8n}}) \leq \frac{1}{2^{4n}}$ probability for large enough λ. This probability is over the oracle distributions of $\mathcal{T}', \mathcal{T}''$ as well. The Markov inequality then gives for large enough λ:

$$\Pr_{\mathcal{T}', \mathcal{T}'' \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]}\left[d_{\mathsf{TD}}\left(G^{\mathcal{T}^2}(k), G^{\mathcal{T}^3}(k)\right) \geq \frac{1}{2^{2n}}\right] \leq 1/2^{2n}.$$

□

Claim 8. *For large enough λ,*

$$d_{\mathsf{TD}}\left(G^{\mathcal{T}^3}(k), G^{\mathcal{T}^4}(k)\right) \leq \frac{3}{\lambda}.$$

Proof. This follows in the same way as Claim 6. □

As a result of the past three claims and the triangle inequality, with probability $1 - \frac{1}{2^{2n}}$ over the distributions $\mathcal{T}', \mathcal{T}''$, we have for large enough λ,

$$\Pr\left[G^{\mathcal{T}^1}(k) = G^{\mathcal{T}^4}(k)\right] \geq 1 - \frac{7}{\lambda}. \tag{3}$$

We now argue that this inverse-polynomial difference must in fact be negligible. Notice that $\mathcal{T}^4 \in \mathbb{T}$ so by Lemma 9, $G^{\mathcal{T}^4}$ satisfies the correctness condition of a PRG. In other words, it is almost-deterministic, meaning it must output a fixed value except with negligible probability. Combining this with Eq. (3), we get that, with probability $1 - \frac{1}{2^{2n}}$ over the oracle distributions, $G^{\mathcal{T}^1}(k)$ and $G^{\mathcal{T}^4}(k)$ agree except with negligible probability.

Performing the same changes on every $\log(2m) < n < r$ and $x^* \in \{0,1\}^n$, we reach $\mathcal{T}''$. The union bound gives us that with probability

$$1 - \sum_{\log(2m)<n<r} \left(\sum_{x^* \in \{0,1\}^n} \frac{1}{2^{2n}} \right) > 1 - \sum_{\log(2m)<n<r} \frac{1}{2^n} > 1 - \frac{r}{2m}.$$

over the distribution of oracles $\mathcal{T}', \mathcal{T}''$, $G^{\mathcal{T}'}(k)$ and $G^{\mathcal{T}''}(k)$ agree except with negligible probability. More formally,

$$\Pr_{\mathcal{T}', \mathcal{T}'' \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]} \left[G^{\mathcal{T}'}(k) = G^{\mathcal{T}''}(k) \right] \geq \left(1 - \frac{r}{2m}\right) (1 - \mathsf{negl}(\lambda)) \geq 1 - \frac{r}{m}.$$

Remark 1. It may seem problematic that our argument involves exponentially many hybrids, where each differs from its neighbor with some negligible error. Why don't the errors accumulate and become non-negligible? This issue is mitigated by the almost-determinism property of PRGs, which guarantees that the most likely output must be produced with $1 - \mathsf{negl}(\lambda)$ probability. As a result, the output cannot be gradually altered across the hybrids. □

By Lemma 10, for any $k \in \{0,1\}^\lambda$, there exists a string $y_k^{\mathcal{T}_{\leq \log(2m)}}$, such that

$$\Pr_{\tilde{\mathcal{T}} \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]} \left[G^{\tilde{\mathcal{T}}}(k) = y_k^{\mathcal{T}_{\leq \log(2m)}} \right] \geq 1 - \frac{r}{m} \geq 1 - \frac{1}{\lambda^4}. \tag{4}$$

Furthermore, note that G should not depend on $\mathcal{C}$, since the $\mathsf{PRF}^{\mathrm{qs}}$ given in Lemma 9 does not rely on $\mathcal{C}$ (only on $\mathcal{O}$ and σ) and G is based on a black-box construction from this $\mathsf{PRF}^{\mathrm{qs}}$.

We now consider a generator $\overline{G}$ that does not depend on the oracles, and is defined as follows on inputs of length $\lambda + 16m^3$:

$\overline{G}(k)$:

- Parse k as (k_1, k_2) where $k_1 \in \{0,1\}^\lambda$ and $k_2 \in \{0,1\}^{16m^3}$.
- Construct functions $(\sigma_n^{k_2}, \mathcal{O}_n^{k_2})_{n \leq \log(2m)}$ in the same way as Construction 5 but with the randomness determined by k_2.
- For $j \in [\lambda]$:
  1. For each $\log(2m) \leq i \leq r$, sample uniformly at random two $2r$-degree polynomials $\tilde{O}_i : \mathbb{F}_{2^i} \to \mathbb{F}_{2^{8i}}$ and $\tilde{P}_i : \mathbb{F}_{2^{2i}} \to \mathbb{F}_{2^i}$. Let $(\tilde{\sigma}_i, \tilde{\mathcal{O}}_i)$ be the resulting oracles.
  2. Run $G(k_1)$ and answer the queries as follows:
     (a) For a query x of length $n \leq \log(2m)$, respond using $(\sigma_n^{k_2}, \mathcal{O}_n^{k_2})$.
     (b) For a query x of length $n > \log(2m)$, respond using $(\tilde{\sigma}_n, \tilde{\mathcal{O}}_n)$.
  3. Let y_j be the result of $G(k_1)$.
- Set $y = \mathsf{vote}_\lambda(y_1, \ldots, y_\lambda)$.
- Output (y, k_2).

Now consider the following variants of PRG security experiment.

- $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$:
  1. Sample oracle $\mathcal{T}$ as in Construction 5.
  2. $b \leftarrow \mathsf{Exp}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}}, G^{\mathcal{T}}}(1^\lambda)$.
  3. Output b.
- $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$:
  1. Sample oracle $\mathcal{T}$ as in Construction 5.
  2. $b \leftarrow \mathsf{Exp}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}_{\leq \log(2m)}, \mathcal{C}}, G^{\mathcal{T}}}(1^\lambda)$. *Notice that $\mathcal{A}$ only has access to $\mathcal{T}_{\leq \log(2m)}$ in this experiment.*
  3. Output b.
- $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$:
  1. $b \leftarrow \mathsf{Exp}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{C}}, \overline{G}}(1^\lambda)$.
  2. Output b.

By Claim 5, for any QPT adversary $\mathcal{A}$,

$$\Pr_{\mathcal{T} \leftarrow \mathbb{T}}\left[\mathsf{Adtg}^{\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}}, G^{\mathcal{T}}}(1^\lambda) \leq O\left(\frac{1}{\lambda^4}\right)\right] \geq \frac{3}{4}.$$

Therefore, for large enough λ,

$$\Pr\left[\mathsf{Exp}_1^{\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \leq \frac{1}{4} \cdot \frac{1}{2} + \frac{3}{4} \cdot \frac{1}{\lambda^3} \leq \frac{1}{4} + \frac{1}{\lambda^2}.$$

Next, the only difference between $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$ and $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$ is that $\mathcal{A}$'s oracle access to $\mathcal{T}$ is restricted. Hence, for any QPT adversary $\mathcal{A}$, there exists a QPT $\mathcal{B}$ such that $\Pr\left[\mathsf{Exp}_2^{\mathcal{A}}(\lambda) = 1\right] \leq \Pr\left[\mathsf{Exp}_1^{\mathcal{B}}(\lambda) = 1\right]$.

Finally, we need to relate the success probabilities of $\mathsf{Exp}_2^{\mathcal{A}}$ and $\mathsf{Exp}_3^{\mathcal{A}}$.

Claim 9. *For any QPT adversary $\mathcal{A}$ and large enough λ, there exists a QPT algorithm $\mathcal{B}$ such that*

$$\Pr\left[\mathit{Exp}_3^{\mathcal{A}}(\lambda)=1\right] \leq \Pr\left[\mathit{Exp}_2^{\mathcal{B}}(\lambda)=1\right] + \frac{1}{\lambda^4}.$$

Proof. Fix an adversary $\mathcal{A}$ in $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$. We construct an algorithm $\mathcal{B}$ in $\mathsf{Exp}_2^{\mathcal{B}}(\lambda)$ as follows.

In $\mathsf{Exp}_2^{\mathcal{B}}(\lambda)$, a random input $k_1 \leftarrow \{0,1\}^\lambda$ and a random bit $b \leftarrow \{0,1\}$ are sampled. Then, let $y_0 \leftarrow G^{\mathcal{T}}(k_1)$ and $y_1 \leftarrow \{0,1\}^s$. $\mathcal{B}^{\mathcal{T}_{\leq \log(2m)},\mathcal{C}}$ receives y_b and must guess b.

$\mathcal{B}^{\mathcal{T}_{\leq \log(2m)},\mathcal{C}}$ commences as follows. For every $n \leq \log(2m)$, it queries $\sigma_n(|0^{9n+1}\rangle)$ $4m^2$ times and measures the result in the computational basis. This allows $\mathcal{B}$ to obtain all the evaluations of O_n and learn the function entirely, except with negligible probability. Next, for each $n \leq \log(2m)$, $\mathcal{B}$ uses O_n and the oracle $\mathcal{O}_n$ to learn P_n entirely, which requires at most $2m$ queries.

$\mathcal{B}$ encodes $(P_n, O_n)_{n\leq \log(2m)}$ into a string, say k_2, of length less than $16m^3$. $\mathcal{B}$ runs $\mathcal{A}^{\mathcal{C}}$ on (y_b, k_2) and receives a response b'. $\mathcal{B}$ outputs b'.

As long as $\mathcal{B}$ encodes $(P_n, O_n)_{n\leq \log(2m)}$ correctly (which occurs with $1-\mathsf{negl}(\lambda)$ probability) and $\overline{G}(k_1,k_2) = y_0$ (which occurs with probability $1-2r/m$ by Eq. (4)), then it is clear that $\Pr\left[\mathsf{Exp}_2^{\mathcal{B}}(\lambda)=1\right]$ is at least $\Pr\left[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1\right]$. Therefore,

$$\Pr\left[\mathsf{Exp}_2^{\mathcal{B}}(\lambda)=1\right] \geq \Pr\left[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1\right](1-\mathsf{negl}(\lambda))\left(1-\frac{2r}{m}\right),$$

which implies, for large enough λ,

$$\Pr\left[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1\right] \leq \Pr\left[\mathsf{Exp}_2^{\mathcal{B}}(\lambda)=1\right] + \frac{1}{\lambda^4}.$$

□

To sum up, for any QPT adversary $\mathcal{A}$, there exists a QPT algorithm $\mathcal{B}$ and large enough λ such that

$$\Pr\left[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1\right] \leq \Pr\left[\mathsf{Exp}_2^{\mathcal{B}}(\lambda)=1\right] + \frac{1}{\lambda^4} \leq \frac{3}{4} + \frac{1}{\lambda^2} + \frac{1}{\lambda^4} \leq \frac{3}{4} + \frac{1}{\lambda} \tag{5}$$

Notice that $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$ is just the PRG security experiment for $\overline{G}$ against $\mathcal{A}^{\mathcal{C}}$. On the other hand, there exists a trivial search attack, using a PSPACE oracle, against PRG security, given that any polynomial-space quantum computations with classical inputs can be simulated using a PSPACE oracle. In particular, there exists an adversary $\overline{\mathcal{A}}$ such that

$$\Pr\left[\mathsf{Exp}_3^{\overline{\mathcal{A}}}(\lambda)=1\right] \geq 1-\mathsf{negl}(\lambda).$$

contradicting Eq. (5) above.

Therefore, there does not exist a fully black-box construction of a PRG from a $\mathsf{PRF}^{\mathrm{qs}}$. □

5.2 Separating OWSG from $\perp$-PRG

We show that there does not exist a fully black-box construction of a OWSGs from a $\perp$-PRG with CPTP access. The proof is given in Section 5.2.1, but the result and implications are discussed below.

Theorem 12. *Let $\lambda \in \mathbb{N}$ be the security parameter. For any polynomial $m(\lambda) > \lambda$ and pseudodeterminism error $\mu(\lambda) = O(\lambda^{-c})$ for $c > 0$, there does not exist a fully black-box construction of a* OWSG *from CPTP access to a (μ, m)-$\perp$-*PRG.

This separation is significant because there are multiple MicroCrypt primitives that can be built from $\perp$-PRGs, thus yielding new separations in MicroCrypt.

First of all, we note that $\perp$-PRGs can be used to build the following primitives [4].

Lemma 11. *There exists black-box constructions of the following primitives from $\perp$-PRGs:*

1. *$\perp$-PRFs.*
2. *(Many-time) digital signatures of classical messages with classical keys and signatures.*
3. *Quantum public-key encryption of classical messages with tamper-resilient keys and classical ciphertexts.*

As a result of Theorem 12 and Lemma 11, we obtain a separation between OWSGs and some cryptographic applications.

Corollary 7. *There does not exist a fully black-box construction of a* OWSG *from CPTP access to:*

1. *$\perp$-PRFs.*
2. *(Many-time) digital signatures of classical messages with classical keys and signatures.*
3. *Quantum public-key encryption of classical messages with tamper-resilient keys and classical ciphertexts.*

5.2.1 Separation Proof. To demonstrate the separation between OWSGs and $\perp$-PRGs, we will use two independent oracles: an oracle for a PSPACE-complete language and a $\perp$-pseudodeterministic random oracle. We show that the random oracle already acts as a $\perp$-PRG, noting that the $\perp$-pseudodeterminism is not a problem given the nature of $\perp$-PRGs. On the other hand, the unpredictability of the $\perp$-pseudodeterminism in the random oracle prevents its use in the construction of almost-deterministic primitives such as a OWSG. Specifically, through a careful argument, we show that any OWSG must exist independently of the random oracle and, thus, cannot exist in the presence of a PSPACE oracle. Given this distinction, there cannot exist a fully black-box construction of a OWSG from a $\perp$-PRG.

Let $\lambda, n \in \mathbb{N}$ be security parameters. Let $c > 0$ be a constant. We define a sequence of CPTP oracles $\mathcal{O} := \{\mathcal{O}_n\}_{n\in\mathbb{N}}$ as follows.

Construction 7. *Fix a pseudodeterminism error $\mu(n) = n^{-c}$. Let $w = w(n)$ be any function such that $2^{-w} \in [\mu/16, \mu/4]$ and let m be polynomial such that $m(n) > n$. Sample a random permutation $P_n \leftarrow \Pi_n$ and random functions $Q_n \leftarrow \Pi_{n,n}$ and $O_n \leftarrow \Pi_{n,m}$. The quantum channel $\mathcal{O}_n := \mathcal{O}[P_n, Q_n, O_n]$ on n-qubit input ρ does as follows:*

- *Measure ρ in the computational basis and let x denote the result.*
- *Compute $y = O_n(x)$.*
- *Compute $q = Q_n(x)$ and let $p_x := q/2^n$, where q is interpreted as an integer in $[0 : 2^n]$.*
- *Compute $z = P_n(x)$. If the first w-bits of z are 0^w, then let $|\phi_x\rangle := \sqrt{p_x}|\bot\rangle + \sqrt{1-p_x}|y\rangle$.*
- *Otherwise, let $|\phi_x\rangle := |y\rangle$.*
- *Measure $|\phi_x\rangle$ in the computational basis and output the result.*

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

$$\mathcal{G}_n^{\mathcal{O}} := \{x \in \{0,1\}^n : P_n(x)_{[1:w]} \neq 0^w\},$$

where $P_n(x)_{[1:w]}$ denotes the first w-bits of $P_n(x)$.

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

Lemma 12. *$\mathcal{O}_n$ has the following properties:*

- $\Pr_{x \leftarrow \{0,1\}^n}\left[x \in \mathcal{G}_n^{\mathcal{O}}\right] \geq 1 - \frac{\mu}{4}$.
- *For every $x \in \mathcal{G}_n^{\mathcal{O}}$, there exists a non-$\bot$ value $y \in \{0,1\}^m$ such that:*

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

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

Theorem 13. *Let $\lambda \in \mathbb{N}$ be the security parameter. For any polynomial $m(\lambda) > \lambda$ and pseudodeterminism error $\mu(\lambda) = O(\lambda^{-c})$ for $c > 0$, there does not exist a fully black-box construction of a* OWSG *from CPTP access to a (μ, m)-$\bot$-*PRG.

Proof. Assume for contradiction that there exists a black-box construction of a OWSG $\tilde{G}^F$ from CPTP access to a (μ, m)-$\bot$-PRG F. First, we show that there exists a (μ, m)-$\bot$-PRG relative to the oracles $(\mathcal{O}, \mathcal{C})$.

Claim 10. *Under security parameter $n \in \mathbb{N}$, the sequence of functions $\{\mathcal{O}_n[P_n, Q_n, O_n]\}_{n\in\mathbb{N}}$ is a $(\mu(n), m(n))$-$\bot$-*PRG *for all possible sequences P and Q and with probability 1 over the distribution of O. Furthermore, correctness is satisfied for all possible oracles.*

Proof. By Lemma 12, $\mathcal{O}$ satisfies the correctness/pseudodeterminism condition of a (μ, m)-$\bot$-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 $q = q(n)$:

$$\left| \Pr \begin{bmatrix} k \leftarrow \{0,1\}^n \\ y_1 \leftarrow \mathcal{O}_n(k) \\ \vdots \\ y_q \leftarrow \mathcal{O}_n(k) \end{bmatrix} : \mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1, ..., y_q) = 1 \right] - \Pr \begin{bmatrix} k \leftarrow \{0,1\}^n \\ y \leftarrow \{0,1\}^m \\ y_1 \leftarrow \mathsf{ls}\text{-}\bot(\mathcal{O}_n(k), y) \\ \vdots \\ y_q \leftarrow \mathsf{ls}\text{-}\bot(\mathcal{O}_n(k), y) \end{bmatrix} : \mathcal{A}^{\mathcal{O},\mathcal{C}}(y_1, \ldots, y_q) = 1 \right] \Bigg| \leq \mathsf{negl}(n)$$

Let Z_n be the function that outputs 0^m on any input and let $\mathcal{Z}_n \coloneqq \mathcal{O}[P_n, Q_n, Z_n]$. Note that

- $\mathcal{Z}_n$ is independent of O_n,
- $\mathcal{O}_n(k) = \mathsf{ls}\text{-}\bot(\mathcal{O}_n(k), O_n(k)) = \mathsf{ls}\text{-}\bot(\mathcal{Z}_n(k), O_n(k))$,
- $\mathsf{ls}\text{-}\bot(\mathcal{O}_n(k), y) = \mathsf{ls}\text{-}\bot(\mathcal{Z}_n(k), y)$.

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

Lemma 2.2 from [33] states that a random oracle acts as a PRG i.e.:

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

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:

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

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

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

where,

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

By Markov inequality, we get that

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

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

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

except for finitely many $n\in\mathbb{N}$. There are countable number of 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 $\perp$-PRG for any P,Q and with probability 1 over the distribution of O. □

By our assumption, the above claim implies the existence of a OWSG $G^{\mathcal{O}[P,Q,O]}$, by the assumed existence of the black-box construction $\tilde{G}$, for any P,Q and with probability 1 over the distribution of O.

Claim 11. *For any QPT adversary $\mathcal{A}$ and polynomial $t=t(\lambda)$:*

$$\Pr_{\mathcal{O}}\left[\mathsf{Adtg}^{\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{O},\mathcal{C}},G^{\mathcal{O}}}(1^\lambda,1^t)\leq O\left(\frac{1}{\lambda^4}\right)\right]\geq\frac{3}{4}.$$

where the probability is taken over the oracle distribution.

Proof. Same as the proof of Claim 5. □

Let $r=r(\lambda)$ be a polynomial denoting the maximum run-time of G on any input and let $m=r^4+\lambda$.

Intuitively, we will argue that G cannot depend on $\mathcal{O}_n$ for large n, due to the deterministic nature of G, and thus cannot exist in the presence of a PSPACE oracle.

We use the notation $G\simeq G'$ to mean that there exists negligible function $\epsilon=\epsilon(\lambda)$ such that for every $k\in\{0,1\}^\lambda$, there exists a pure-state $|\psi_k\rangle$, such that $\Pr[G(k)=|\psi_k\rangle]$ and $\Pr[G'(k)=|\psi_k\rangle]$ are both at least $1-\epsilon$.

Claim 12. *Let $\mathcal{O}':=\mathcal{O}[P',Q',O']$ and $\mathcal{O}'':=\mathcal{O}[P'',Q'',O'']$ be two oracles such that $(P'_n,Q'_n,O'_n)=(P''_n,Q''_n,O''_n)$ for all $n\leq\log(2m)$ and $n\geq r$. Then, $G^{\mathcal{O}'}\simeq G^{\mathcal{O}''}$.*

Proof. We will first show that there exists a function $\ell=\ell(\lambda)$ and sequences $P^1,P^2,\ldots,P^\ell,Q^1,Q^2,\ldots,Q^\ell$, and $O^1,O^2,\ldots,O^\ell$, where $P^i:=\{P^i_n\}_{n\in\mathbb{N}}$, $Q^i:=\{Q^i_n\}_{n\in\mathbb{N}}$, and $O^i:=\{O^i_n\}_{n\in\mathbb{N}}$, such that:

1. $P_n^i \in \Pi_n$, $Q_n^i \in \Pi_{n,n}$, and $O_n^i \in \Pi_{n,m}$ for any $i \in [\ell]$ and $n \in \mathbb{N}$.
2. $P' = P^1$ and $P'' = P^\ell$.
3. $Q' = Q^1$ and $Q'' = Q^\ell$.
4. $O' = O^1$ and $O'' = O^\ell$.
5. For any $i \in [\ell]$,

$$d_{\mathsf{TD}}(\mathcal{O}^i, \mathcal{O}^{i+1}) \coloneqq \sum_{n\in\mathbb{N}} \sum_{x\in\{0,1\}^n} d_{\mathsf{TD}}(\mathcal{O}_n^i(x), \mathcal{O}_n^{i+1}(x)) \leq \frac{1}{m},$$

where $\mathcal{O}^i \coloneqq \mathcal{O}[P^i, Q^i, O^i]$.

We will now describe how to construct such a sequence. Note that $\mathcal{O}'_n$ and $\mathcal{O}''_n$ only differ for $\log(2m) < n < r$. For such values of n, if we set $Q_n^2(x)$ to $Q_n^1(x) + 1$ or $Q_n^1(x) - 1$, while keeping the other functions fixed, the resulting oracles satisfy:

$$d_{\mathsf{TD}}(\mathcal{O}^1, \mathcal{O}^2) \leq \frac{1}{2^{\log(2m)}} \leq \frac{1}{m}$$

It is not difficult to see that this allows constructing the sequence of functions described. Specifically, for any $\log(2m) < n < r$, we perform small changes to Q'_n until we reach a function, say Q_n^j, that sends all values to 1^n, while keeping all other functions fixed. Then, we set $O_n^{j+1}(x) = O''_n(x)$ for all x such that $P_n(x)_{[1:w]} = 0^w$ and keep $O_n^{j+1}(x) = O_n^j(x)$ otherwise. This step does not change the oracle, i.e. $\mathcal{O}^j = \mathcal{O}^{j+1}$, because Q_n^j and Q_n^{j+1} return 1^n on these inputs so both oracles return $\perp$.

Next, we perform small changes to Q_n^{j+1} until we reach a function, say Q_n^t for some $t > j$, that sends all values to 0^n. Then, we set P_n^{t+1} to any function in Π_n. Again, this step does not change the oracle, i.e. $\mathcal{O}^t = \mathcal{O}^{t+1}$, because Q_n^t and Q_n^{t+1} return 0^n on any input. The new function P_n^{t+1} allows us to perform the first step on a new set of inputs i.e. we can modify O_n^{t+1} on all inputs such that $P_n^{t+1}(x)_{[1:w]} = 0^w$. Iteratively applying these modifications allows us to reach the required functions O''_n and P''_n. Finally, we perform small changes to Q_n while keeping the other functions fixed to obtain Q''_n. These steps are performed for all $n \in [\log(2m) : r]$ to build the sequence described.

Since $P_n^i \in \Pi_n$, $Q_n^i \in \Pi_{n,n}$, and $O_n^i \in \Pi_{n,m}$ for any $n \in \mathbb{N}$, by Claim 10, $G^{\mathcal{O}^i}$ is almost-deterministic for any $i \in [\ell]$.

Note that for any $i \in [\ell]$, $\mathcal{O}_i$ is an oracle with classical output and $d_{\mathsf{TD}}(\mathcal{O}^i, \mathcal{O}^{i+1}) \leq \frac{1}{m}$. Therefore, with probability at least $(1 - \frac{1}{m})^r \geq 1 - \frac{r}{m}$, the responses that G receives from $\mathcal{O}_i$ and $\mathcal{O}_{i+1}$ are indistinguishable. This means that for any $k \in \{0,1\}^\lambda$, with probability at least $1 - \frac{r}{m} - \mathsf{negl}(\lambda)$, $G^{\mathcal{O}^{i+1}}(k)$ outputs the same state generated by $G^{\mathcal{O}^i}(k)$. In order to satisfy the determinism property of OWSGs, this must mean that $G^{\mathcal{O}^i} \simeq G^{\mathcal{O}^{i+1}}$ for all $i \in [\ell]$. By induction, we obtain $G^{\mathcal{O}'} \simeq G^{\mathcal{O}''}$. □

For any oracle $\mathcal{O} = \mathcal{O}[P, Q, O]$, G is independent of $(P_n, Q_n, O_n)_{n\geq r}$. So, by the above claim, for any $(P_n, Q_n, O_n)_{n\leq\log(2m)}$ and $k \in \{0,1\}^\lambda$, there exists a

state $|\psi_k^{\mathcal{O}_{\leq \log(2m)}}\rangle$, such that for any $(\tilde{P}, \tilde{Q}, \tilde{O})$ satisfying $(\tilde{P}_n, \tilde{Q}_n, \tilde{O}_n)_{n \leq \log(2m)} = (P_n, Q_n, O_n)_{n \leq \log(2m)}$,

$$\Pr\left[G^{\mathcal{O}[\tilde{P},\tilde{Q},\tilde{O}]}(k) = \left|\psi_k^{\mathcal{O}_{\leq \log(2m)}}\right\rangle\right] \geq 1 - \mathsf{negl}(\lambda). \tag{6}$$

We now consider a generator $\overline{G}$ that does not depend on the oracle and is defined as follows on inputs of length $\lambda + 16m^3$:

$\overline{G}(k)$:

- Parse k as (k_1, k_2) where $k_1 \in \{0,1\}^\lambda$ and $k_2 \in \{0,1\}^{16m^3}$.
- Construct functions $(\mathcal{O}_n^{k_2})_{n \leq \log(2m)}$ in the same way as Construction 7 but with the randomness determined by k_2.
- Initiate an empty memory $\mathbf{M}$.
- Run $G(k_1)$ and answer the queries as follows:
  1. For a query x of length $n \leq \log(2m)$, respond with $\mathcal{O}_n^{k_2}(x)$.
  2. For a query x of length $n > \log(2m)$, if $(x, y) \in \mathbf{M}$ for some y, respond with y. Otherwise, sample $y \leftarrow \{0,1\}^m$, store (x, y) in $\mathbf{M}$, and respond with y.
- Let $|\psi\rangle$ be the result of $G(k_1)$.
- Output $|\psi\rangle \otimes |k_2\rangle$.

Consider the following experiment variants of OWSG security with some polynomial $t = t(\lambda)$.

- $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$:
  1. Sample oracle $\mathcal{O}$ as in Construction 7.
  2. $b \leftarrow \mathsf{Exp}^{\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{O},\mathcal{C}},G^{\mathcal{O}}}(1^\lambda, 1^t)$.
  3. Output b.
- $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$:
  1. Sample oracle $\mathcal{O}$ as in Construction 7.
  2. $b \leftarrow \mathsf{Exp}^{\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{O}_{\leq \log(2m)},\mathcal{C}},G^{\mathcal{O}}}(1^\lambda, 1^t)$. *Notice that $\mathcal{A}$ only has access to $\mathcal{O}_{\leq \log(2m)}$ in this experiment.*
  3. Output b.
- $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$:
  1. $b \leftarrow \mathsf{Exp}^{\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{C}},\overline{G}}(1^\lambda, 1^t)$.
  2. Output b.

By Claim 11, for any QPT adversary $\mathcal{A}$,

$$\Pr_{\mathcal{O}}\left[\mathsf{Adtg}^{\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{O},\mathcal{C}},G^{\mathcal{O}}}(1^\lambda, 1^t) \leq O\left(\frac{1}{\lambda^4}\right)\right] \geq \frac{3}{4}.$$

Therefore, for large enough λ

$$\Pr\left[\mathsf{Exp}_1^{\mathcal{A}}(\lambda) = 1\right] \leq \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{\lambda^3}.$$

Next, the only difference between $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$ and $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$ is that $\mathcal{A}$'s oracle access to $\mathcal{O}$ is restricted. Hence, for any QPT adversary $\mathcal{A}$, there exists a QPT $\mathcal{B}$ such that $\Pr\left[\mathsf{Exp}_2^{\mathcal{A}}(\lambda)=1\right] \leq \Pr\left[\mathsf{Exp}_1^{\mathcal{B}}(\lambda)=1\right]$.

Finally, we need to relate $\mathsf{Exp}_2^{\mathcal{A}}$ and $\mathsf{Exp}_3^{\mathcal{A}}$.

Claim 13. *For any QPT adversary $\mathcal{A}$, there exists a QPT algorithm $\mathcal{B}$ such that*

$$\Pr\left[\mathit{Exp}_3^{\mathcal{A}}(\lambda)=1\right] \leq \Pr\left[\mathit{Exp}_2^{\mathcal{B}}(\lambda)=1\right] + \mathsf{negl}(\lambda).$$

Proof. Fix an adversary $\mathcal{A}$ in $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$. We construct an algorithm $\mathcal{B}$ in $\mathsf{Exp}_2^{\mathcal{B}}(\lambda)$ as follows.

In $\mathsf{Exp}_2^{\mathcal{B}}(\lambda)$, a random input $k_1 \leftarrow \{0,1\}^\lambda$ is sampled and $t+1$ evaluations are generated $|\psi_i\rangle \leftarrow G^{\mathcal{O}}(k_1)$ for $i \in [t+1]$. Then, $\mathcal{B}^{\mathcal{O}_{\leq \log(2m)},\mathcal{C}}$ receives $\bigotimes_{i\in[t]} |\psi_i\rangle$.

$\mathcal{B}$ commences as follows. It queries the oracle $\mathcal{O}_{\leq\log(2m)}$ $4m^2$ times on every input of length less than $\log(2m)$. This allows $\mathcal{B}$ to describe $\mathcal{O}_{\leq\log(2m)}$ in a string, say k_2, of length less than $16m^3$. Specifically, k_2 is used to describe the randomness used in constructing $\mathcal{O}_{\leq\log(2m)}$ (see Construction 7). It is not difficult to see that $\mathcal{B}$ can learn $\mathcal{O}_{\leq\log(2m)}$ exactly, except with negligible probability. $\mathcal{B}^{\mathcal{O}_{\leq\log(2m)},\mathcal{C}}$ runs $\mathcal{A}^{\mathcal{C}}$ on $\bigotimes_{i\in[t]}(|\psi_i\rangle \otimes |k_2\rangle)$ and receives a response (k_1', k_2'). $\mathcal{B}$ outputs k_1'.

Next, the experiment in $\mathsf{Exp}_2^{\mathcal{B}}(\lambda)$ computes $|\psi_{k_1'}\rangle \leftarrow G^{\mathcal{O}}(k_1')$ and measures $|\psi_{t+1}\rangle$ with $\{|\psi_{k_1'}\rangle\langle\psi_{k_1'}|, I - |\psi_{k_1'}\rangle\langle\psi_{k_1'}|\}$. If the result is $|\psi_{k'}\rangle\langle\psi_{k'}|$, then the output is $b=1$, and the output is $b=0$ otherwise.

Notice that the input $\mathcal{A}$ receives and needs to invert from $\mathcal{B}$ has the same distribution as the input it receives and needs to invert in $\mathsf{Exp}_3^{\mathcal{A}}$. Moreover, as long as $\mathcal{B}$ encodes $\mathcal{O}_{\leq\log(2m)}$ correctly (which occurs with $1-\mathsf{negl}(\lambda)$ probability), it is clear that $\Pr\left[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1\right]$ is at most $\Pr\left[\mathsf{Exp}_2^{\mathcal{B}}(\lambda)=1\right]$, since the inversion task in the former is at least as hard as in the latter. Therefore,

$$\Pr\left[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1\right] \leq \Pr\left[\mathsf{Exp}_2^{\mathcal{B}}(\lambda)=1\right] + \mathsf{negl}(\lambda).$$

□

To sum up, for large enough λ, for any QPT adversary $\mathcal{A}$, there exists a QPT algorithm $\mathcal{B}$ such that

$$\Pr\left[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1\right] \leq \Pr\left[\mathsf{Exp}_2^{\mathcal{B}}(\lambda)=1\right] + \mathsf{negl}(\lambda) \tag{7}$$

$$\leq \frac{1}{4} + \frac{3}{4\lambda^3} + \mathsf{negl}(\lambda) \tag{8}$$

Notice that $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$ is just the OWSG security experiment for $\overline{G}$ against $\mathcal{A}^{\mathcal{C}}$. On the other hand, by Lemma 1, there exists an attack against any OWSG

using a PSPACE oracle. In particular, there exists an adversary $\overline{\mathcal{A}}$ such that

$$\Pr\left[\mathsf{Exp}_3^{\overline{\mathcal{A}}}(\lambda) = 1\right] \geq \frac{1}{2}.$$

contradicting Eq. (8) above.

Therefore, there does not exist a fully black-box construction of a OWSG from a (μ, m)-$\perp$-PRG. □

5.3 Separating $\perp$-PRG from $\mathsf{PRF}^{\mathrm{qs}}$

We strengthen the first separation (Theorem 9) but in the more restricted setting of CPTP access. Specifically, we show that there does not exist a fully black-box construction of a $\perp$-PRG from CPTP access to a $\mathsf{PRF}^{\mathrm{qs}}$.

The proof is given in Section 5.3.1, but the result and its implications are discussed below.

Theorem 14. *Let $\lambda, n \in \mathbb{N}$ be security parameters and let $s = s(\lambda) > 3\lambda$ and $\mu = \mu(\lambda) \leq \lambda^{-c}$ be functions where $c > 0$ is a constant. There does not exist a fully black-box construction of a (μ, s)-$\perp$-PRG from CPTP access to a PRF^{qs}.*

As a result of Theorem 14, we obtain a separation between $\perp$-PRGs and many other MicroCrypt primitives since $\mathsf{PRF}^{\mathrm{qs}}$ inherit many of the same applications as PRFs. We note that this separation extends to SPRSs since they imply $\perp$-PRGs [4, 31].

Corollary 8. *There does not exist a fully black-box construction of a $\perp$-PRGs or a SPRSs from CPTP access to:*

1. PRG^{qs}, SPRS^{qs}, LPRS^{qs}, *and* PRU^{qs}.
2. *Statistically-binding computationally hiding bit commitments with classical communication.*
3. *Message authentication codes of classical messages with classical communication.*
4. *CCA2-secure symmetric encryption of classical messages with classical keys and ciphertexts.*
5. *EV-OWPuzzs.*

5.3.1 Separation Proof.

We prove that there does not exist a fully black-box construction of a $\perp$-PRG from a $\mathsf{PRF}^{\mathrm{qs}}$.

We present (μ, s)-$\perp$-PRG security relative to an oracle $\mathcal{T}$ in the form of an experiment to simplify notation later on.

$\mathsf{Exp}^{\perp\text{-PRG}}_{\mathcal{A}^{\mathcal{T}}, G^{\mathcal{T}}}(1^\lambda, 1^q)$:

1. Sample $k \leftarrow \{0,1\}^\lambda$, $b \leftarrow \{0,1\}$, and $y \leftarrow \{0,1\}^s$.
2. If $b = 0$, for each $i \in [q]$, generate $y_i \leftarrow G^{\mathcal{T}}(k)$.
3. If $b = 1$, for each $i \in [q]$, generate $y_i \leftarrow \mathsf{Is}\text{-}\perp(G^{\mathcal{T}}(k), y)$.
4. $b' \leftarrow \mathcal{A}^{\mathcal{T}}(y_1, \ldots, y_q)$.
5. If $b' = b$, output 1. Otherwise, output 0.

We define the advantage of the adversary in this experiment as follows.

$$\mathsf{Adtg}^{\perp\text{-}\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}},G^{\mathcal{T}}}(1^\lambda) := \Pr\left[\mathsf{Exp}^{\perp\text{-}\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}},G^{\mathcal{T}}}(1^\lambda) = 1\right] - \frac{1}{2}.$$

We now describe the oracles used in the separation. These are similar to the oracles given in Construction 5, which were used to separate PRG and $\mathsf{PRF}^{\mathrm{qs}}$. However, in this case, σ is no longer a unitary, which aids in preventing its use in a wider variety of primitives, including $\perp$-PRGs.

Construction 8. *For $n \in \mathbb{N}$, let $O_n \leftarrow \Pi_{n,n}$ and $P_n \leftarrow \Pi_{2n,n}$ be random functions. Let $\mathcal{T} := (\sigma, \mathcal{O}, \mathcal{C})$ be a tuple of oracles, where $\sigma = \{\sigma_n\}_{n\in\mathbb{N}}$, $\mathcal{O} := \{\mathcal{O}_n\}_{n\in\mathbb{N}}$, and $\mathcal{C} := \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ are defined as follows:*

1. *$\mathcal{C}$ is for membership in a PSPACE-complete language.*
2. *$\sigma_n(1^n)$:*
   (a) *Sample $x \leftarrow \{0,1\}^n$.*
   (b) *Output $|x, O_n(x)\rangle$.*
3. *$\mathcal{O}_n$: Unitary of the classical function that maps (x, y, a), where $x, y, a \in \{0,1\}^n$, to $P_n(x, a)$ if $O_n(x) = y$ and to $\perp$ otherwise.*

We first introduce some notation for the proof, similar to Section 5.1. Let $\mathbb{T}$ denote the set of all possible oracles and let $\mathcal{T} \leftarrow \mathbb{T}$ denote sampling an oracle in the way given in Construction 8. For any oracle $\mathcal{T}$ and integer $m \in \mathbb{N}$, let $\mathcal{T}_{\leq m}$ denote the sequence of oracles $(\sigma_n, \mathcal{O}_n)_{n\leq m}$ and let $\mathbb{T}[\mathcal{T}_{\leq m}]$ denote the set

$$\mathbb{T}[\mathcal{T}_{\leq m}] := \{\tilde{\mathcal{T}} \in \mathbb{T} : \tilde{\mathcal{T}}_{\leq m} = \mathcal{T}_{\leq m}\}.$$

Theorem 15. *Let $\lambda, n \in \mathbb{N}$ be security parameters and let $s = s(\lambda) > 3\lambda$ and $\mu = \mu(\lambda) \leq \lambda^{-c}$ be functions, where $c > 0$ is a constant. There does not exist a fully black-box construction of a (μ, s)-$\perp$-PRG from CPTP access to a PRF^{qs}.*

Proof. For simplicity, we only prove the theorem for $(9n, n)$-$\mathsf{PRF}^{\mathrm{qs}}$, but the proof easily generalizes to other parameters by modifying the parameters of the oracles.

Assume, for the purpose of obtaining a contradiction, that $\tilde{G}^F$ is a fully black-box construction of a (μ, s)-$\perp$-PRG, from CPTP access to a $(9n, n)$-$\mathsf{PRF}^{\mathrm{qs}}$ F. We first show that there exists a $\mathsf{PRF}^{\mathrm{qs}}$ relative to $\mathcal{T}$.

Claim 14. *There exists a (quantum-query-secure) $(9n, n)$-PRF^{qs} relative to $\mathcal{T}$ for any O and with probability 1 over the distribution of P. Furthermore, correctness is satisfied for any oracle $\mathcal{T}$.*

Proof. Similar to proof of Lemma 9. □

Given that there exists a $\mathsf{PRF}^{\mathrm{qs}}$ relative to $\mathcal{T}$ with probability 1 over the oracle distribution and correctness is satisfied for all oracles, there exists a $\perp$-PRG $G^{\mathcal{T}}$, from the assumed existence of the black-box construction $\tilde{G}$, with probability 1 over the distribution of P and correctness is satisfied for any oracle $\mathcal{T} \in \mathbb{T}$.

Claim 15. *For any QPT adversary $\mathcal{A}$:*

$$\Pr_{\mathcal{T}\leftarrow\mathbb{T}}\left[\mathsf{Adtg}^{\perp\text{-}\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}},G^{\mathcal{T}}}(1^\lambda)\leq O\left(\frac{1}{\lambda^4}\right)\right]\geq\frac{3}{4}.$$

Proof. Same as the proof of Claim 5. □

Let $r=r(\lambda)$ denote the maximum run-time of G and $m\coloneqq 10(\frac{r\lambda}{\mu})^4+\lambda$. Hence, G makes at most r queries to the oracles.

Fix an oracle $\mathcal{T}$. We will need to show the following lemma.

Lemma 13. *There exists a set $\mathbb{G}_\lambda^{\mathcal{T}_{\leq\log(2m)}}\subseteq\{0,1\}^\lambda$ such that:*

1. $\Pr_{k\leftarrow\{0,1\}^\lambda}\left[k\in\mathbb{G}_\lambda^{\mathcal{T}_{\leq\log(2m)}}\right]\geq 1-\sqrt{\mu}$.
2. *If $k\in\mathbb{G}_\lambda^{\mathcal{T}_{\leq\log(2m)}}$, then there exists a string $y_k^{\mathcal{T}_{\leq\log(2m)}}$ such that:*

$$\Pr\left[G_\lambda^{\tilde{\mathcal{T}}}(k)=y_k^{\mathcal{T}_{\leq\log(2m)}}:\tilde{\mathcal{T}}\leftarrow\mathbb{T}\left[\mathcal{T}_{\leq\log(2m)}\right]\right]\geq 1-3\sqrt{\mu},$$

where the probability is taken over the distribution of oracles $\tilde{\mathcal{T}}$ satisfying $\tilde{\mathcal{T}}_{\leq\log(2m)}=\mathcal{T}_{\leq\log(2m)}$ and the resulting distribution of $G_\lambda^{\tilde{\mathcal{T}}}(k)$.

Proof. Define $\mathbb{G}_\lambda^{\mathcal{T}_{\leq\log(2m)}}\subseteq\{0,1\}^\lambda$ as the set of inputs that are in the good set $\mathcal{G}_\lambda^{\tilde{\mathcal{T}}}$ with at least $1-\sqrt{\mu}$ probability over the distribution $\tilde{\mathcal{T}}\leftarrow\mathbb{T}\left[\mathcal{T}_{\leq\log(2m)}\right]$.

We first show that at least $1-\sqrt{\mu}$ fraction of inputs are in this set i.e. $\Pr_{k\leftarrow\{0,1\}^\lambda}\left[k\in\mathbb{G}_\lambda^{\mathcal{T}_{\leq\log(2m)}}\right]\geq 1-\sqrt{\mu}$. Otherwise, we have that at least $\sqrt{\mu}$ fraction of inputs are in the good set with probability less than $1-\sqrt{\mu}$ over the oracle distribution. In this case, even if the rest of the inputs are in the good set for any oracle, the average size of the good set is smaller than $1-\mu$. More explicitly, we get

$$\mathop{\mathbb{E}}_{\mathcal{T}\leftarrow\mathbb{T}}\left[|\mathcal{G}_\lambda^{\mathcal{T}}|\right]<(1-\sqrt{\mu})\cdot 1+\sqrt{\mu}\cdot(1-\sqrt{\mu})=1-\mu.$$

This gives a contradiction since $\mathop{\mathbb{E}}_{\mathcal{T}\leftarrow\mathbb{T}}\left[|\mathcal{G}_\lambda^{\mathcal{T}}|\right]\geq 1-\mu$ by the pseudodeterminism of $\perp$-PRGs. Hence, we must have $\Pr_{k\leftarrow\{0,1\}^\lambda}\left[k\in\mathbb{G}_\lambda^{\mathcal{T}_{\leq\log(2m)}}\right]\geq 1-\sqrt{\mu}$.

Now let $k\in\mathbb{G}_\lambda^{\mathcal{T}_{\leq\log(2m)}}$. By the definition of this set,

$$\Pr\left[G_\lambda^{\tilde{\mathcal{T}}}(k)\neq\perp:\tilde{\mathcal{T}}\leftarrow\mathbb{T}[\mathcal{T}_{\leq\log(2m)}]\right]\geq 1-2\sqrt{\mu}. \tag{9}$$

It is sufficient to show that

$$\Pr\left[\perp\neq y_1\neq y_2\neq\perp\;\middle|\;\begin{array}{c}\mathcal{T}',\mathcal{T}''\leftarrow\mathbb{T}\left[\mathcal{T}_{\leq\log(2m)}\right]\\ y_1\leftarrow G^{\mathcal{T}'}(k)\\ y_2\leftarrow G^{\mathcal{T}''}(k)\end{array}\right]\leq\sqrt{\mu}. \tag{10}$$

If this holds, then combining this with Eq. (9) implies that there exists a unique output $y_k^{\mathcal{T}_{\leq \log(2m)}}$ such that

$$\Pr\left[G_\lambda^{\tilde{\mathcal{T}}}(k) = y_k^{\mathcal{T}_{\leq \log(2m)}} : \tilde{\mathcal{T}} \leftarrow \mathbb{T}\left[\mathcal{T}_{\leq \log(2m)}\right]\right] \geq 1 - 3\sqrt{\mu}$$

which is the result we need to show. Assume that Eq. (10) does not hold. To show a contradiction, we commence with a hybrid argument.

- Hybrid H_0:
  1. Sample an oracle $\mathcal{T}' \leftarrow \mathbb{T}\left[\mathcal{T}_{\leq \log(2m)}\right]$ and let O' and P' denote the functions encoded in $\mathcal{T}'$.
  2. Sample $k \leftarrow \{0,1\}^\lambda$.
  3. Compute $y_1 \leftarrow G^{\mathcal{T}'}(k)$.
  4. Compute $y_2 \leftarrow G^{\mathcal{T}'}(k)$.
  5. Output (y_1, y_2).
- Hybrid H_1:
  1. Sample oracle $\mathcal{T}' \leftarrow \mathbb{T}\left[\mathcal{T}_{\leq \log(2m)}\right]$.
  2. Sample $k \leftarrow \{0,1\}^\lambda$.
  3. Initiate empty memory $\mathbf{M}$.
  4. Compute $y_1 \leftarrow G^{\mathcal{T}'_{\mathbf{M}}}(k)$.
  5. Reset $\mathbf{M}$ to empty.
  6. Compute $y_2 \leftarrow G^{\mathcal{T}'_{\mathbf{M}}}(k)$.
  7. Output (y_1, y_2).

  Here, $\mathcal{T}'_{\mathbf{M}} := (\sigma'_{\mathbf{M}}, \mathcal{O}'_{\mathbf{M}})$ is defined as follows.
  - $\sigma'_{\mathbf{M}}(1^n)$:
    1. If $n \leq \log(2m)$, then output $\sigma(1^n)$.
    2. Otherwise, sample $x \leftarrow \{0,1\}^n$.
    3. Store x in memory $\mathbf{M}$.
    4. Output $|x, O'_n(x)\rangle$.
  - $\mathcal{O}'_{\mathbf{M}}$:
    1. If the input is of size $3n$ and $n \leq \log(2m)$, then apply $\mathcal{O}_n$.
    2. Otherwise, apply the unitary of the classical function which on input (x, y, a), outputs $P'_n(x, a)$ if $x \in \mathbf{M}$ and $y = O'_n(x)$, and outputs $\perp$ otherwise.
- Hybrid H_2:
  1. Sample oracles $\mathcal{T}', \mathcal{T}'' \leftarrow \mathbb{T}\left[\mathcal{T}_{\leq \log(2m)}\right]$.
  2. Sample $k \leftarrow \{0,1\}^\lambda$.
  3. Initiate empty memory $\mathbf{M}$.
  4. Compute $y_1 \leftarrow G^{\mathcal{T}'\mathbf{M}}(k)$.
  5. Reset $\mathbf{M}$ to empty.
  6. Compute $y_2 \leftarrow G^{\mathcal{T}''\mathbf{M}}(k)$.
  7. Output (y_1, y_2).

  where $\mathcal{T}'_{\mathbf{M}}$ and $\mathcal{T}''_{\mathbf{M}}$ are defined in the same way as in H_1.
- Hybrid H_3:
  1. Sample $\mathcal{T}', \mathcal{T}'' \leftarrow \mathbb{T}\left[\mathcal{T}_{\leq \log(2m)}\right]$.
  2. Sample $k \leftarrow \{0,1\}^\lambda$.

3. Compute $y_1 \leftarrow G^{\mathcal{T}'}(k)$.
4. Compute $y_2 \leftarrow G^{\mathcal{T}''}(k)$.
5. Output (y_1, y_2).

Claim 16. *With probability at least $1 - \mu/8$, hybrids H_0 and H_1 are indistinguishable.*

Proof. The oracles in these two hybrids only differ on inputs starting with $(x, O'(x))$ such that $(x, O'(x))$ was not the response of a query to σ by G in the same evaluation. Crucially, O' is a random function. Therefore, to distinguish these two oracles, G needs to do unstructured search for an input starting with $x \notin \mathbf{M}$ that maps to a non-$\perp$ element.

The lower bound for unstructured search [36, 10] states that G cannot distinguish these two hybrids with better than $O(\frac{r^2}{2^n}) \leq \mu/8$ probability. □

Claim 17. *With probability at least $1 - \mu/8$, hybrids H_1 and H_2 are indistinguishable.*

Proof. Consider the two evaluations in H_1. Let $(x_i^1, O'(x_i^1))_{i\in[r]}$ and $(x_j^2, O'(x_j^2))_{j\in[r]}$ denote the responses of oracles $(\sigma'_n)_{n>\log(2m)}$ to the queries of G in the first and second evaluation, respectively.

If $\{x_i^1\}_{i\in[r]} \cap \{x_j^2\}_{j\in[r]} = \emptyset$, then these two hybrids are indistinguishable, since O', O'', P', P'' are random functions. This scenario occurs with at least $1 - \frac{(2r)^2}{m} \geq 1 - \mu/8$ probability by the birthday problem. □

Claim 18. *With probability at least $1 - \mu/8$, hybrids H_2 and H_3 are indistinguishable.*

Proof. This follows in the same way as Claim 16. □

By the above three claims and the triangle inequality, we have that with probability at least $1 - \mu/2$, hybrids H_0 and H_3 are indistinguishable.

Notice that in H_3, by our assumption, the probability that (y_1, y_2) are non-$\perp$ distinct strings is at least $\sqrt{\mu}$. Therefore, the probability that the two strings generated in hybrid H_0 also are non-$\perp$ distinct strings is $\sqrt{\mu} - \mu/2 > \mu/2$. However, this contradicts the pseudodeterminism condition of G, since for a fixed oracle, two evaluations should yield the same string or $\perp$ except with negligible probability. □

We are now ready to prove the main result (Theorem 14) using a hybrid argument. But first, we consider a generator $\tilde{G}$ that only depends on $\mathcal{T}_{\leq\log(2m)}$ and is defined as follows.

$\tilde{G}^{\mathcal{T}_{\leq \log(2m)}}(k)$:

- For $j \in [\lambda]$:
  1. For each $\log(2m) \leq i \leq r$, sample uniformly at random two $2r$-degree polynomials $\tilde{O}_i : \mathbb{F}_{2^i} \to \mathbb{F}_{2^i}$ and $\tilde{P}_i : \mathbb{F}_{2^{2i}} \to \mathbb{F}_{2^i}$. Let $(\tilde{\sigma}_i, \tilde{\mathcal{O}}_i)$ be the resulting oracles.
  2. Run $G(k)$ and answer the queries as follows:
     (a) For a query x of length $n \leq \log(2m)$, respond using $(\sigma_n, \mathcal{O}_n)$.
     (b) For a query x of length $n > \log(2m)$, respond using $(\tilde{\sigma}_n, \tilde{\mathcal{O}}_n)$.
  3. Let y_j be the result of $G(k_1)$.
- Set $y = \mathsf{vote}_\lambda(y_1, \ldots, y_\lambda)$.
- Output y.

We also consider a generator $\overline{G}$ that does not depend on the oracles, and is defined as follows on inputs of length $\lambda + 16m^3$:

$\overline{G}(k)$:

- Parse k as (k_1, k_2) where $k_1 \in \{0,1\}^\lambda$ and $k_2 \in \{0,1\}^{16m^3}$.
- Construct functions $(\sigma_n^{k_2}, \mathcal{O}_n^{k_2})_{n \leq \log(2m)}$ in the same way as Construction 8 but with the randomness determined by k_2.
- For $j \in [\lambda]$:
  1. For each $\log(2m) \leq i \leq r$, sample uniformly at random two $2r$-degree polynomials $\tilde{O}_i : \mathbb{F}_{2^i} \to \mathbb{F}_{2^i}$ and $\tilde{P}_i : \mathbb{F}_{2^{2i}} \to \mathbb{F}_{2^i}$. Let $(\tilde{\sigma}_i, \tilde{\mathcal{O}}_i)$ be the resulting oracles.
  2. Run $G(k_1)$ and answer the queries as follows:
     (a) For a query x of length $n \leq \log(2m)$, respond using $(\sigma_n^{k_2}, \mathcal{O}_n^{k_2})$.
     (b) For a query x of length $n > \log(2m)$, respond using $(\tilde{\sigma}_n, \tilde{\mathcal{O}}_n)$.
  3. Let y_j be the result of $G(k_1)$.
- Set $y = \mathsf{vote}_\lambda(y_1, \ldots, y_\lambda)$.
- Output (y, k_2).

Now consider the following variants of $\perp$-PRG security experiment, where we set $q(\lambda) = 1$ (see Definition 11).

- $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$:
  1. Sample oracle $\mathcal{T}$ as in Construction 7.
  2. $b \leftarrow \mathsf{Exp}^{\perp\text{-PRG}}_{\mathcal{A}^{\mathcal{T}}, G^{\mathcal{T}}}(1^\lambda, 1^q)$.
  3. Output b.
- $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$:
  1. Sample oracle $\mathcal{T}$ as in Construction 8.
  2. $b \leftarrow \mathsf{Exp}^{\perp\text{-PRG}}_{\mathcal{A}^{\mathcal{T}_{\leq \log(2m)}, \mathcal{C}}, G^{\mathcal{T}}}(1^\lambda, 1^q)$. *Notice that $\mathcal{A}$ only has access to $\mathcal{T}_{\leq \log(2m)}$ and $\mathcal{C}$ in this experiment.*
  3. Output b.
- $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$:
  1. Sample oracle $\mathcal{T}$ as in Construction 8.

2. $b \leftarrow \mathsf{Exp}^{\perp\text{-}\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}_{\leq \log(2m)}, \mathcal{C}}, \tilde{G}^{\mathcal{T}_{\leq \log(2m)}}}(1^\lambda, 1^q)$.
3. Output b.

– $\mathsf{Exp}^{\mathcal{A}}_4(\lambda)$:
1. $b \leftarrow \mathsf{Exp}^{\perp\text{-}\mathsf{PRG}}_{\mathcal{A}^{\mathcal{C}}, \overline{G}}(1^\lambda, 1^q)$.
2. Output b.

Remark 2. Technically, $\overline{G}$ and $\tilde{G}$ do not satisfy the pseudodeterminism conditions of a $\perp$-PRG, but we can still run the $\perp$-PRG security experiment on them.

Claim 19. *For any QPT adversary $\mathcal{A}$, there exists a QPT adversary $\mathcal{B}$ such that*

$$\Pr\left[\mathit{Exp}^{\mathcal{A}}_2(\lambda) = 1\right] \leq \Pr\left[\mathit{Exp}^{\mathcal{B}}_1(\lambda) = 1\right].$$

Proof. This is clear because the only difference between theses experiments is that the adversary's access to the oracle in Exp_2 is restricted. □

Claim 20. *For any QPT adversary $\mathcal{A}$ and large enough λ,*

$$d_{TD}(\mathit{Exp}^{\mathcal{A}}_2(\lambda), \mathit{Exp}^{\mathcal{A}}_3(\lambda)) \leq 1/\lambda^4.$$

Proof. By Lemma 13, for any oracle $\mathcal{T}$, there exists a set $\mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}$ such that:

1. $\Pr_{k \leftarrow \{0,1\}^\lambda}\left[k \in \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}\right] \geq 1 - \sqrt{\mu}$.
2. If $k \in \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}$, then there exists a string $y_k^{\mathcal{T}_{\leq \log(2m)}}$ such that:

$$\Pr\left[G_\lambda^{\tilde{\mathcal{T}}}(k) = y_k^{\mathcal{T}_{\leq \log(2m)}} : \tilde{\mathcal{T}} \leftarrow \mathbb{T}\left[\mathcal{T}_{\leq \log(2m)}\right]\right] \geq 1 - 3\sqrt{\mu}. \tag{11}$$

By definition of $\perp$-PRGs, for an input $k \in \{0,1\}^\lambda$, there exists a string $y_k^{\mathcal{T}}$ such that $\Pr\left[G_\lambda^{\mathcal{T}}(k) \in \{y_k^{\mathcal{T}}, \perp\}\right] \geq 1 - \mathsf{negl}(\lambda)$. As usual, $\mathcal{G}_\lambda^{\mathcal{T}}$ denotes the good set of inputs for $G_\lambda^{\mathcal{T}}$ (see Definition 11).

Let B denote the event that the key k and oracle $\mathcal{T}$ sampled in $\mathsf{Exp}^{\mathcal{A}}_2(\lambda)$ or $\mathsf{Exp}^{\mathcal{A}}_3(\lambda)$ satisfy the following conditions: $k \in \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}} \cap \mathcal{G}_\lambda^{\mathcal{T}}$ and $y_k^{\mathcal{T}_{\leq \log(2m)}} = y_k^{\mathcal{T}}$, where $y_k^{\mathcal{T}_{\leq \log(2m)}}$ is the string that satisfies Eq. (11).

We now show that event B occurs with probability at least $1 - 6\sqrt{\mu}$. Note that

$$\Pr\left[k \in \mathcal{G}_\lambda^{\mathcal{T}} : \begin{matrix} k \leftarrow \{0,1\}^\lambda \\ \mathcal{T} \leftarrow \mathbb{T} \end{matrix}\right] \geq 1 - \mu$$

$$\Pr\left[k \in \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}} : \begin{matrix} k \leftarrow \{0,1\}^\lambda \\ \mathcal{T} \leftarrow \mathbb{T} \end{matrix}\right] \geq 1 - \sqrt{\mu}.$$

Therefore, $\Pr\left[k \in \mathcal{G}_\lambda^{\mathcal{T}} \cap \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}} : k \leftarrow \{0,1\}^\lambda, \mathcal{T} \leftarrow \mathbb{T}\right] \geq 1 - 2\sqrt{\mu}$. Given this, to show that event B occurs with at least $1 - 6\sqrt{\mu}$ probability, it is sufficient to show that $y_k^{\mathcal{T}_{\leq \log(2m)}} = y_k^{\mathcal{T}}$ occurs with at least $1 - 4\sqrt{\mu}$ probability when

$k \leftarrow \mathcal{G}_\lambda^{\mathcal{T}} \cap \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}$ and $\mathcal{T} \leftarrow \mathbb{T}$. If this does not hold, then by an averaging argument, we obtain

$$\Pr\left[G_\lambda^{\tilde{\mathcal{T}}}(k) = y_k^{\mathcal{T}_{\leq \log(2m)}} : \begin{array}{c}\mathcal{T} \leftarrow \mathbb{T} \\ \tilde{\mathcal{T}} \leftarrow \mathbb{T}\left[\mathcal{T}_{\leq \log(2m)}\right] \\ k \leftarrow \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}\end{array}\right] =$$

$$\Pr\left[G_\lambda^{\mathcal{T}}(k) = y_k^{\mathcal{T}_{\leq \log(2m)}} : \begin{array}{c}\mathcal{T} \leftarrow \mathbb{T} \\ k \leftarrow \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}\end{array}\right] \leq$$

$$(1-2\sqrt{\mu})\Pr\left[G_\lambda^{\mathcal{T}}(k) = y_k^{\mathcal{T}_{\leq \log(2m)}} : \begin{array}{c}\mathcal{T} \leftarrow \mathbb{T} \\ k \leftarrow \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}} \cap \mathcal{G}_\lambda^{\mathcal{T}}\end{array}\right] + 2\sqrt{\mu} \leq$$

$$(1-2\sqrt{\mu})(1-4\sqrt{\mu}) + 2\sqrt{\mu} < 1 - 3\sqrt{\mu}$$

which contradicts Eq. (11), as this equation states that the first probability should be at least $1 - 3\sqrt{\mu}$. Therefore, event B occurs with at least $1 - 6\sqrt{\mu}$ probability.

If event B occurs, then in $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$, the generator $G_\lambda^{\mathcal{T}}(k)$ returns $y_k^{\mathcal{T}_{\leq \log(2m)}}$ with probability $1 - \mathsf{negl}(\lambda)$ for every query. While, in $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$, $\overline{G}_\lambda(k)$ returns $y_k^{\mathcal{T}_{\leq \log(2m)}}$ with probability $1 - \mathsf{negl}(\lambda)$.

Overall, if event B occurs, then the two experiments can only be distinguished with at most negligible probability. Given that event B occurs with probability at least $1 - 6\sqrt{\mu}$, for large enough λ these two experiments can be distinguished with at most $6\sqrt{\mu} + \mathsf{negl}(\lambda) < \frac{1}{\lambda^4}$ probability. □

Claim 21. *For any QPT adversary $\mathcal{A}$, there exists a QPT algorithm $\mathcal{B}$ such that*

$$\Pr\left[\mathit{Exp}_4^{\mathcal{A}}(\lambda) = 1\right] \leq \Pr\left[\mathit{Exp}_3^{\mathcal{B}}(\lambda) = 1\right] + 3\sqrt{\mu}.$$

Proof. Fix an adversary $\mathcal{A}$ in $\mathsf{Exp}_4^{\mathcal{A}}(\lambda)$. We construct an algorithm $\mathcal{B}$ in $\mathsf{Exp}_3^{\mathcal{B}}(\lambda)$ as follows.

In $\mathsf{Exp}_3^{\mathcal{B}}(\lambda)$, an input $k_1 \leftarrow \{0,1\}^\lambda$, a bit $b \leftarrow \{0,1\}$, and a string $y \leftarrow \{0,1\}^s$ are sampled. Then, let $y_0 \leftarrow \tilde{G}^{\mathcal{T}_{\leq \log(2m)}}(k_1)$ and $y_1 \leftarrow \mathsf{ls}\text{-}\perp(\tilde{G}^{\mathcal{T}_{\leq \log(2m)}}(k_1), y)$. $\mathcal{B}^{\mathcal{T}_{\leq \log(2m)}, \mathcal{C}}$ receives y_b and must guess b.

$\mathcal{B}^{\mathcal{T}_{\leq \log(2m)}, \mathcal{C}}$ commences as follows. For every $n \leq \log(2m)$, it queries $\sigma_n(1^n)$ $4m^2$ times. This allows $\mathcal{B}$ to obtain all the evaluations of O_n and learn the function entirely, except with negligible probability. Next, for each $n \leq \log(2m)$, $\mathcal{B}$ uses O_n and the oracle $\mathcal{O}_n$ to learn P_n entirely, which requires at most $2m$ queries.

$\mathcal{B}$ encodes $(P_n, O_n)_{n \leq \log(2m)}$ into a string, say k_2, of length less than $16m^3$. $\mathcal{B}$ runs $\mathcal{A}^{\mathcal{C}}$ on (y_b, k_2) and receives a response b'. $\mathcal{B}$ outputs b'.

As long as $\mathcal{B}$ encodes $(P_n, O_n)_{n \leq \log(2m)}$ correctly (which occurs with $1 - \mathsf{negl}(\lambda)$ probability) and $\Pr[\overline{G}(k_1, k_2) = (y_0, k_2)] \geq 1 - \mathsf{negl}(\lambda)$, then it is clear that $\Pr\left[\mathsf{Exp}_3^{\mathcal{B}}(\lambda) = 1\right]$ is at least $\Pr\left[\mathsf{Exp}_4^{\mathcal{A}}(\lambda) = 1\right] - \mathsf{negl}(\lambda)$. If $k_1 \in \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}$,

then these conditions occur with $1 - \mathsf{negl}(\lambda)$ probability. Recall $k_1 \in \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}$ occurs with probability at least $1 - \sqrt{\mu}$ by Lemma 13. All together, this means

$$\Pr\left[\mathsf{Exp}_3^{\mathcal{B}}(\lambda) = 1\right] \geq \Pr\left[\mathsf{Exp}_4^{\mathcal{A}}(\lambda) = 1\right](1 - \mathsf{negl}(\lambda))(1 - 2\sqrt{\mu}),$$

which implies, for large enough λ,

$$\Pr\left[\mathsf{Exp}_3^{\mathcal{A}}(\lambda) = 1\right] \leq \Pr\left[\mathsf{Exp}_2^{\mathcal{B}}(\lambda) = 1\right] + 3\sqrt{\mu}.$$

□

By the triangle inequality, for any QPT adversary $\mathcal{A}$ and large enough λ, there exists a QPT adversary $\mathcal{B}$ such that

$$\left|\Pr\left[\mathsf{Exp}_4^{\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}_0^{\mathcal{B}}(\lambda) = 1\right]\right| \leq \frac{1}{\lambda^4} + 3\sqrt{\mu} + \mathsf{negl}(\lambda). \tag{12}$$

By Claim 15, for any QPT adversary $\mathcal{A}$, there exists a negligible function δ such that:

$$\Pr_{\mathcal{T} \leftarrow \mathbb{T}}\left[\mathsf{Adtg}^{\perp\text{-}\mathsf{PRG}}_{\mathcal{A}^{\mathcal{T}}, G^{\mathcal{T}}}(1^\lambda) \leq O\left(\frac{1}{\lambda^4}\right)\right] \geq \frac{3}{4}.$$

Therefore, for any QPT adversary $\mathcal{A}$ and large enough λ,

$$\Pr\left[\mathsf{Exp}_0^{\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \leq \frac{3}{4} \cdot \frac{1}{\lambda^3} + \frac{1}{4} \cdot \frac{1}{2}.$$

By Eq. (12), for any QPT adversary $\mathcal{A}$ and large enough λ,

$$\Pr\left[\mathsf{Exp}_4^{\mathcal{A}}(\lambda) = 1\right] \leq \frac{3}{4} + 3\sqrt{\mu} + \frac{1}{\lambda^2}. \tag{13}$$

Notice that $\overline{G}$ does not use any oracle access.

On the other hand, we will show that there exists an adversary that contradicts Eq. (13).

Claim. There exists a QPT algorithm $\overline{\mathcal{A}}$ such that

$$\Pr\left[\mathsf{Exp}_4^{\overline{\mathcal{A}}}(\lambda) = 1\right] \geq 1 - 2\sqrt{\mu}.$$

Proof. In the security experiment, $\mathsf{Exp}_4^{\overline{\mathcal{A}}}(\lambda)$, an input $k \leftarrow \{0,1\}^{\lambda + 16m^3}$, a bit $b \leftarrow \{0,1\}$ and string $y \leftarrow \{0,1\}^\ell$ are sampled, where $\ell := s + 16m^3$. Then, sample $y_0 \leftarrow \overline{G}(k)$ and $y_1 \leftarrow \mathsf{ls}\text{-}\perp(\overline{G}(k), y)$. $\overline{\mathcal{A}}$ receives y_b and needs to guess b. Interpret y_b as (y'_b, k_2), where $y'_b \in \{0,1\}^s$ and $k_2 \in \{0,1\}^{16m^3}$.

$\overline{\mathcal{A}}$ uses oracle access to $\mathcal{C}$ to run the algorithm which computes $\overline{G}(k, k_2)$ on every input $k \in \{0,1\}^\lambda$ and if any computation yields y_b, then it outputs 0 and otherwise outputs 1. $\overline{\mathcal{A}}$ returns the output of this algorithm.

Let $I_\lambda := \{y : \exists k \in \{0,1\}^\lambda \text{ s.t } \Pr[\overline{G}(k,k_2) = y] \geq \frac{1}{2^{3\lambda/2}}\}$. Notice that $|I_\lambda| \leq 2^{5\lambda/2}$.

In the case $b = 1$, note that $\Pr[y_1 \in I_\lambda] \leq \frac{2^{5\lambda/2}}{2^s} \leq \frac{1}{2^{\lambda/2}}$ is negligible. This implies that there is a negligible probability that $\overline{\mathcal{A}}^{\mathcal{C}}(y_1)$ outputs 0 by the union bound. Therefore, $\overline{\mathcal{A}}$ guesses $b' = b$ correctly except with negligible probability when $b = 1$.

Meanwhile, if $b = 0$, then $y_0 \leftarrow \overline{G}(k)$. Since k is sampled uniformly at random, by Lemma 13,

$$\Pr[\overline{G}(k) = y_0] \geq (1 - \mathsf{negl}(\lambda))(1 - \sqrt{\mu}) \geq 1 - 2\sqrt{\mu}$$

In other words, $\overline{\mathcal{A}}^{\mathcal{C}}$ outputs $b' = b$ with probability $1 - 2\sqrt{\mu}$ when $b = 0$.

All in all,

$$\Pr\left[\mathsf{Exp}^{\perp\text{-}\mathsf{PRG}}_{\overline{\mathcal{A}}^{\mathcal{C}},\overline{G}}(1^\lambda) = 1\right] \geq (1 - \mathsf{negl}(\lambda)) \cdot \frac{1}{2} + (1 - 2\sqrt{\mu}) \cdot \frac{1}{2} \geq 1 - 2\sqrt{\mu}.$$

□

The claim above contradicts Eq. (13). So there does not exist a fully black-box construction of a (μ, s)-$\perp$-PRG from a $\mathsf{PRF}^{\mathrm{qs}}$. □

References

[1] Prabhanjan Ananth, Aditya Gulati, Luowen Qian, and Henry Yuen. "Pseudorandom (Function-Like) Quantum State Generators: New Definitions and Applications". In: *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part I*. Springer. 2022, pp. 237–265.

[2] Prabhanjan Ananth, Yao-Ting Lin, and Henry Yuen. "Pseudorandom Strings from Pseudorandom Quantum States". In: *arXiv preprint arXiv:2306.05613* (2023).

[3] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. "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*. Springer. 2022, pp. 208–236.

[4] Mohammed Barhoush, Amit Behera, Lior Ozer, Louis Salvail, and Or Sattath. *Signatures From Pseudorandom States via $\perp$-PRFs*. 2024. arXiv: 2311.00847 [cs.CR]. URL: https://arxiv.org/abs/2311.00847.

[5] James Bartusek. *Secure Quantum Computation with Classical Communication*. Cryptology ePrint Archive, Paper 2021/964. 2021. URL: https://eprint.iacr.org/2021/964.

[6] James Bartusek, Zvika Brakerski, and Vinod Vaikuntanathan. "Quantum state obfuscation from classical oracles". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 1009–1017.

[7] James Bartusek, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. "Obfuscation of pseudo-deterministic quantum circuits". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 2023, pp. 1567–1578.

[8] Amit Behera, Zvika Brakerski, Or Sattath, and Omri Shmueli. "Pseudorandomness with proof of destruction and applications". In: *Cryptology ePrint Archive* (2023).

[9] Amit Behera, Giulio Malavolta, Tomoyuki Morimae, Tamer Mour, and Takashi Yamakawa. "A new world in the depths of microcrypt: Separating OWSGs and quantum money from QEFID". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2025, pp. 23–52.

[10] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. "Strengths and weaknesses of quantum computing". In: *SIAM journal on Computing* 26.5 (1997), pp. 1510–1523.

[11] John Bostanci, Boyang Chen, and Barak Nehoran. "Oracle separation between quantum commitments and quantum one-wayness". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2025, pp. 3–22.

[12] Zvika Brakerski and Omri Shmueli. "Scalable pseudorandom quantum states". In: *Annual International Cryptology Conference*. Springer. 2020, pp. 417–440.

[13] Bruno Cavalar, Eli Goldin, Matthew Gray, Peter Hall, Yanyi Liu, and Angelos Pelecanos. "On the computational hardness of quantum one-wayness". In: *arXiv preprint arXiv:2312.08363* (2023).

[14] Boyang Chen, Andrea Coladangelo, and Or Sattath. "The power of a single Haar random state: constructing and separating quantum pseudorandomness". In: *arXiv preprint arXiv:2404.03295* (2024).

[15] Kai-Min Chung, Eli Goldin, and Matthew Gray. "On central primitives for quantum cryptography with classical communication". In: *Annual International Cryptology Conference*. Springer. 2024, pp. 215–248.

[16] Andrea Coladangelo and Saachi Mutreja. "On black-box separations of quantum digital signatures from pseudorandom states". In: *arXiv preprint arXiv:2402.08194* (2024).

[17] Yevgeniy Dodis, Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. "Security amplification for interactive cryptographic primitives". In: *Theory of Cryptography: 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings 6*. Springer. 2009, pp. 128–145.

[18] Nico Döttling, Giulio Malavolta, and Sihang Pu. "A combinatorial approach to quantum random functions". In: *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*. Springer. 2020, pp. 614–632.

[19] Bill Fefferman and Shelby Kimmel. "Quantum vs classical proofs and subset verification". In: *arXiv preprint arXiv:1510.06750* (2015).

[20] Eli Goldin, Tomoyuki Morimae, Saachi Mutreja, and Takashi Yamakawa. "CountCrypt: Quantum Cryptography between QCMA and PP". In: *arXiv preprint arXiv:2410.14792* (2024).

[21] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to construct random functions". In: *J. ACM* 33.4 (1986), pp. 792–807. DOI: 10.1145/6490.6503. URL: https://doi.org/10.1145/6490.6503.

[22] Russell Impagliazzo and Steven Rudich. "Limits on the provable consequences of one-way permutations". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. 1989, pp. 44–61.

[23] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. "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*. Springer. 2018, pp. 126–152.

[24] Dakshita Khurana and Kabir Tomer. "Commitments from quantum one-wayness". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 968–978.

[25] William Kretschmer. "Quantum Pseudorandomness and Classical Complexity". In: *16th Conference on the Theory of Quantum Computation, Communication and Cryptography*. 2021.

[26] William Kretschmer, Luowen Qian, and Avishay Tal. "Quantum-Computable One-Way Functions without One-Way Functions". In: *arXiv preprint arXiv:2411.02554* (2024).

[27] Fermi Ma and Hsin-Yuan Huang. "How to construct random unitaries". In: *arXiv preprint arXiv:2410.10116* (2024).

[28] Mohammad Mahmoody, Ameer Mohammed, Soheil Nematihaji, Rafael Pass, et al. "A note on black-box separations for indistinguishability obfuscation". In: *Cryptology ePrint Archive* (2016).

[29] Tomoyuki Morimae, Shogo Yamada, and Takashi Yamakawa. *Quantum Unpredictability*. Cryptology ePrint Archive, Paper 2024/701. 2024. URL: https://eprint.iacr.org/2024/701.

[30] Tomoyuki Morimae and Takashi Yamakawa. "One-wayness in quantum cryptography". In: *arXiv preprint arXiv:2210.03394* (2022).

[31] Tomoyuki Morimae and Takashi Yamakawa. "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*. Springer. 2022, pp. 269–295.

[32] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. New York, NY, USA: Cambridge University Press, 2000. ISBN: 0-521-63503-9. DOI: 10.1017/CBO9780511976667.

[33] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. "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. Springer. 2018, pp. 520–551.

[34] Dominique Unruh. "Computationally binding quantum commitments". In: *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 35*. Springer. 2016, pp. 497–527.

[35] Dominique Unruh. "Quantum proofs of knowledge". In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2012, pp. 135–152.

[36] Christof Zalka. "Grover's quantum searching algorithm is optimal". In: *Physical Review A* 60.4 (1999), p. 2746.

[37] Mark Zhandry. "A note on quantum-secure PRPs". In: *arXiv preprint arXiv:1611.05564* (2016).

[38] Mark Zhandry. "How to construct quantum random functions". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE. 2012, pp. 679–687.

[39] Mark Zhandry. "Schrödinger's pirate: How to trace a quantum decoder". In: *Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part III 18*. Springer. 2020, pp. 61–91.

A BQ-PRU^{qs} from PRG^{qs}

In this section, we show to build BQ-PRU^{qs}s from PRG^{qs}s. We first introduce some further definitions in the quantum input sampling regime.

A.1 Definitions: PRP^{qs} and BQ-PRU^{qs}

We introduce pseudorandom permutations with quantum input sampling.

Definition 15 (Pseudorandom Permutation with Quantum Key Generation). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ and $m = m(\lambda)$ be polynomials in λ. A tuple of QPT algorithms $(\mathsf{QSamp}, F, F^{-1})$ is called a* (m, n)-pseudorandom permutation with quantum key generation (PRP^{qs})*, if:*

1. $\mathsf{QSamp}(1^\lambda)$: *Outputs a string $k \in \{0,1\}^m$.*
2. $F_k(x)$*: Takes a key $k \in \{0,1\}^m$ and an input $x \in \{0,1\}^n$ and outputs a string $y \in \{0,1\}^n$.*
3. $F_k^{-1}(y)$*: Takes a key $k \in \{0,1\}^m$ and an input $y \in \{0,1\}^n$ and outputs a string $x \in \{0,1\}^n$.*
4. *(*Inverse Relation*) For every $k \in \{0,1\}^m$, there exists a permutation π_k over $\{0,1\}^n$ such that for all $x, y \in \{0,1\}^n$, the following conditions are satisfied:*

$$\Pr_{k \leftarrow \mathsf{QSamp}(1^\lambda)}[F_k(x) = \pi_k(x)] \geq 1 - \mathsf{negl}(\lambda).$$

and

$$\Pr_{k\leftarrow \mathsf{QSamp}(1^\lambda)}\left[F_k^{-1}(y)=\pi_k^{-1}(y)\right]\geq 1-\mathsf{negl}(\lambda).$$

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

$$\left|\Pr_{k\leftarrow \mathsf{QSamp}(1^\lambda)}\left[\mathcal{A}^{F_k,F_k^{-1}}(1^\lambda)=1\right]-\Pr_{O\leftarrow \Pi_n}\left[\mathcal{A}^{O,O^{-1}}(1^\lambda)=1\right]\right|\leq \mathsf{negl}(\lambda).$$

where Π_n is the set of permutations on $\{0,1\}^n$. We say a PRP^{qs} is quantum-query-secure *if the above holds even if $\mathcal{A}$ is given quantum-query-access. Furthermore, in the case where security only holds for $t\leq q$ queries for some polynomial $q=q(\lambda)$, then we call this q-query PRP^{qs}.*

We also define pseudorandom unitaries with quantum input sampling.

Definition 16 (Pseudorandom Unitaries with Quantum Input Sampling). *Let $m=m(\lambda)$ and $n=n(\lambda)$ be polynomials in the security parameter $\lambda\in\mathbb{N}$. A pair of QPT algorithms (QSamp,U) is a* (m,n)-pseudorandom unitary with quantum input sampling ($\mathsf{PRU}^{\mathrm{qs}}$) *if the following holds:*

1. *$\mathsf{QSamp}(1^\lambda)$: Outputs a m-bit key k.*
2. *U_k: Quantum channel that takes an m-bit key k and acts on n-qubit states.*
3. *For any QPT adversary $\mathcal{A}$,*

$$\left|\Pr_{k\leftarrow \mathsf{QSamp}(1^\lambda)}\left[\mathcal{A}^{U_k}(1^\lambda)=1\right]-\Pr_{U\leftarrow \mu}\left[\mathcal{A}^{U}(1^\lambda)=1\right]\right|\leq \mathsf{negl}(\lambda).$$

where μ denotes the Haar measure on the unitary group $\mathcal{U}(\mathbb{C}^n)$. If $\mathcal{A}$ is restricted to only $q=q(\lambda)$ queries to the unitary, then this is denoted as (q,m,n)-BQ-PRU^{qs}.

Note that our definition of is weaker than earlier definitions of PRU [23], as we do not require that the pseudorandom unitary to be a unitary map. We only require that it is indistinguishable from a Haar random unitary. Unfortunately, due to the negligible error inherent in $\mathsf{PRP}^{\mathrm{qs}}$ and $\mathsf{PRF}^{\mathrm{qs}}$, our construction of a $\mathsf{PRU}^{\mathrm{qs}}$ from these primitives is not guaranteed to be a unitary map.

A.2 Result

Note that a $\mathsf{PRG}^{\mathrm{qs}}$ with sufficient expansion easily implies a $\mathsf{PRF}^{\mathrm{qs}}$ with *polynomial* domain through interpreting the output string as a function. However, it is not clear if a $\mathsf{PRG}^{\mathrm{qs}}$ can be used to build full-fledged $\mathsf{PRF}^{\mathrm{qs}}$ with *exponential* domain since the standard construction converting a PRG to a PRF [21] and its quantum adaption [38] both implicitly use the uniform input sampling property of PRGs. Hence, adapting this conversion to the quantum input sampling setting is an interesting open question.

Fortunately, $\mathsf{PRF}^{\mathrm{qs}}$ with polynomial domain can still be useful for applications by converting them to bound-query $\mathsf{PRF}^{\mathrm{qs}}$s with exponential domain using Lemma 14. Specifically, the paper [18] shows how to expand the domain size of a PRF, and the same construction and result apply to $\mathsf{PRF}^{\mathrm{qs}}$s as well.

Lemma 14 (Theorem 7 [18]). *Let $\lambda \in \mathbb{N}$ be the security parameter and q and m be polynomials in λ. Let (QSamp, F) be a (quantum-query-secure) PRF^{qs} with key space $\mathcal{K}_q$, domain $\mathcal{X} : \{0,1\}^\ell$ where $\ell = O(\log(\lambda))$, and co-domain $\mathcal{Z} : \{0,1\}^m$. Then, there exists a q-query (quantum-query-secure) PRF^{qs} (QSamp, F'_q) with the same key sampling algorithm and key space $\mathcal{K}_q$, and with domain and co-domain $\mathcal{Z}$.*

We will use this result to build $\mathsf{BQ\text{-}PRU}^{\mathrm{qs}}$ and $\mathsf{BC\text{-}LPRS}^{\mathrm{qs}}$ from $\mathsf{PRG}^{\mathrm{qs}}$s. First, [37] shows how to build quantum-query-secure pseudorandom permutations from quantum-query-secure PRFs. This conversion queries the PRF a polynomial number of times with respect to the security parameter λ and input length n. Hence, the same proof can be used to show that for any $q' \in \mathsf{poly}(\lambda)$, there exists a $q \in \mathsf{poly}(\lambda, n)$ such that q-query pseudorandom functions imply q'-query pseudorandom permutations.

Corollary 9. *Let $\lambda \in \mathbb{N}$ be the security parameter and $q = q(\lambda)$ and $n = n(\lambda)$ be polynomials in λ. There exists a polynomial $\ell = \ell(\lambda)$, such that (λ, ℓ)-PRG^{qs}s imply (q, λ, n)-$\mathsf{BQ\text{-}PRP}^{qs}$s.*

Recently, [27] showed how to build a PRU from PRPs and PRFs. Notably, each unitary evaluation uses a single quantum query to the PRP and to the PRF. Therefore, we obtain bounded-copy $\mathsf{PRU}^{\mathrm{qs}}$s from bounded-query $\mathsf{PRF}^{\mathrm{qs}}$s and bounded-query $\mathsf{PRP}^{\mathrm{qs}}$s. Furthermore, $\mathsf{BQ\text{-}PRU}^{\mathrm{qs}}$ imply $\mathsf{BC\text{-}LPRS}^{\mathrm{qs}}$ given that LPRS can be viewed as a special case of PRUs, where the unitary can only be queried on the state $|0^n\rangle$.

Theorem 16. *Let $\lambda \in \mathbb{N}$ be the security parameter and q and n be polynomials in λ. There exists a polynomial ℓ in λ such that (λ, ℓ)-PRG^{qs}s imply (q, λ, n)-$\mathsf{BQ\text{-}PRU}^{qs}$s and (q, λ, n)-$\mathsf{BC\text{-}LPRS}^{qs}$.*