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 assumptions. First, we examine primitives such as pseudorandom generators (PRGs) and pseudorandom states (PRSs), extended with quantum input sampling, which we term $\mathsf{PRG}^{\mathsf{qs}}$ and $\mathsf{PRS}^{\mathsf{qs}}$. In these primitives, the input is sampled via a quantum algorithm rather than uniformly at random. The second relaxation, $\bot$-pseudodeterminism, allows the generator to output $\bot$ on an inverse-polynomial fraction of inputs.

We demonstrate an equivalence between (bounded-query) logarithmic-sized $\mathsf{PRS}^{\mathsf{qs}}$, logarithmic-sized $\mathsf{PRS}^{\mathsf{qs}}$, and $\mathsf{PRG}^{\mathsf{qs}}$. Notably, such an equivalence remains unknown for the uniform key sampling versions of these primitives. Furthermore, we establish that $\mathsf{PRG}^{\mathsf{qs}}$ can be constructed from $\bot$-pseudodeterministic PRGs ($\bot$-PRGs).

To further justify our exploration, we present two separation results. First, we examine the relationship between $\bot$-pseudodeterministic notions and their deterministic counterparts. We show that there does not exist a black-box construction of a one-way state generator (OWSG) from a $\bot$-PRG, indicating that $\bot$-pseudodeterministic primitives may be inherently weaker than their deterministic counterparts. Second, we explore the distinction between quantum and uniform input sampling. We prove that there does not exist a black-box construction of a $\bot$-psuedodeterministic OWSG from a $\mathsf{PRF}^{\mathsf{qs}}$, suggesting that primitives relying on quantum input sampling may be weaker than those using traditional uniform sampling. Given the broad cryptographic applicability of $\mathsf{PRF}^{\mathsf{qs}}$s and $\bot$-PRGs, these separation results yield numerous new insights into the 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) [16] may exist even when (classically-evaluable post-quantum) one-way functions (OWF) do not, relative to an oracle [3] [18]. 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 [23, 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 [21] and one-way state generators [23] yield quantum outputs, similar to PRSs. Additionally, assumptions such as $\perp$-PRG [4] and one-way puzzles [17] 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 (SPRS), 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 the for a fixed seed, the output is deterministic on $1 - 1/\mathsf{poly}(n)$ fraction of inputs. 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-time digital signatures and quantum public-key encryption with tamper-resilient keys, which had eluded MicroCrypt.

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

While these applications are powerful, $\perp$-PRGs and SPRSs have not been black-box separated from OWFs [4], which somewhat limits the significance of this result. In fact, most MicroCrypt assumptions, such as pseudorandom function-like states with proofs of destruction [5] and efficiently verifiable one-way puzzles (Ev-OWPuzzs) [17, 8], have been conjectured to be weaker than OWFs, but their separability has not been established. As a result, many applications built in MicroCrypt have only been realized from assumptions which have not been separated from OWFs. 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 $\mathsf{PRG}^{\mathsf{qs}}$s and $\mathsf{PRS}^{\mathsf{qs}}$s.

To clarify, in this work, by PRG we refer to a quantum-evaluable post-quantum-secure pseudorandom generator [5], where security is guaranteed when the input is sampled uniformly at random. Meanwhile, by $\mathsf{PRG}^{\mathsf{qs}}$, we refer to a quantum-evaluable post-quantum-secure pseudorandom generator, where security is guaranteed if the (classical) input is sampled using a specified QPT algorithm. We similarly define PRS and $\mathsf{PRS}^{\mathsf{qs}}$ (see Definitions 5 and 12).

While many MicroCrypt primitives have been previously defined with quantum input sampling [22, 17, 6], the fundamental distinction between primitives based on quantum versus uniform input sampling has not been previously recognized.

In the first part of this work, we introduce natural variants of MicroCrypt primitives that incorporate quantum input sampling and examine the relationships among these primitives, finding surprising results. Specifically, we show black-box constructions for the following:

1. $\mathsf{PRG}^{\mathsf{qs}}$ from $\perp$-$\mathsf{PRG}^{\mathsf{qs}}$.
2. $\mathsf{PRG}^{\mathsf{qs}}$ from bounded-copy $\mathsf{SPRS}^{\mathsf{qs}}$ (BC-$\mathsf{SPRS}^{\mathsf{qs}}$).
3. $\mathsf{SPRS}^{\mathsf{qs}}$ from $\mathsf{PRG}^{\mathsf{qs}}$.
4. BQ-$\mathsf{PRU}^{\mathsf{qs}}$ [6] from $\mathsf{PRG}^{\mathsf{qs}}$.
5. $\mathsf{PRU}^{\mathsf{qs}}$ from $\mathsf{PRF}^{\mathsf{qs}}$.

[4] Notably, the separation between PRS and OWFs [18] only applies to linear-sized PRSs and not to SPRS.

[5] This is different from classically-evaluable post-quantum-secure pseudorandom generators, as demonstrated very recently in [19].

[6] This stands for bounded-query pseudorandom unitaries with quantum key sampling, which is defined in Section 3.

As a direct application of these results, we obtain a method to decrease the output length of $\mathsf{SPRS}^{\mathsf{qs}}$ and to convert a SPRS into a $\mathsf{SPRS}^{\mathsf{qs}}$ of longer output length.

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

Our first separation shows that $\bot$-pseudodeterministic notions may be weaker than deterministic ones.

Theorem 1 (Informal Theorem 8). *There does not exist a black-box construction of a* OWSG *from a* $\bot$-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 PRSs are separated from PRGs [18]. Since $\bot$-PRGs have broad applicability [4], this result yields additional separations as corollaries.

Corollary 1 (Informal Corollary 6). OWSG*s are black-box separated from:*

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

Our second separation shows that primitives with quantum input sampling may be weaker than uniform input sampling, even for $\bot$-pseudodeterministic notions.

Theorem 2 (Informal Theorem 9). *There does not exist a black-box construction of a* $\bot$-OWSG *from a (quantum-query-secure)* PRF^{qs}*s.*

Given that $\mathsf{PRF}^{\mathsf{qs}}$s have many applications, we obtain many other separations as a corollary.

Corollary 2 (Informal Corollary 7). $\bot$-OWSG*s,* $\bot$-PRG*s, and* SPRS*s are black-box separated from:*

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

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

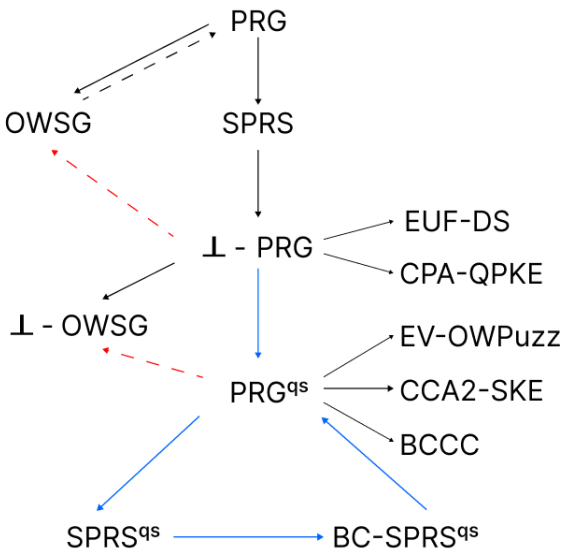

Fig. 1. The black straight arrows indicate implications that are trivial or from previous works [23, 2, 4]. The black dotted arrow indicates a separation from previous work [18]. The blue straight arrows are implications from this work and the dotted red arrows are separations from this work.

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 [22, 17]. Nevertheless, the relations among 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. 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. For a $\perp$-PRG, there exists a set of "good" inputs, that produce deterministic outputs, and a set of "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. It is straightforward to 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.

– 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, [8] use this property to show that uniform and quantum sampling versions of these puzzles are equivalent. Hence, quantum input sampling seems more relevant to deterministic notions such as OWFs.
– Very recently, [19] showed a separation between quantumly-evaluable OWFs and classically-evaluable OWFs, relative to a classical oracle. Our separations are not comparable with theirs. However, as a direct result, they find that classically-evaluable OWFs are separated from many cryptographic applications, such as EUF-DS and QPKE. Our separations imply that such applications are even separated from OWSGs.

1.3 Technical Overview

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* SPRS$^{\text{qs}}$ imply PRG$^{\text{qs}}$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 [16], but instantiated with a polynomial-domain PRF. Note that $\mathsf{PRF}^{\mathsf{qs}}$ with polynomial domain can be trivially derived from $\mathsf{PRG}^{\mathsf{qs}}$s [7].

Modifying the Size of SPRS. By leveraging the above equivalence, we obtain a way to decrease the output length of a $\mathsf{SPRS}^{\mathsf{qs}}$. Simply convert a $\mathsf{SPRS}^{\mathsf{qs}}$ into a $\mathsf{PRG}^{\mathsf{qs}}$, and then convert this back into a $\mathsf{SPRS}^{\mathsf{qs}}$. Due to the change in parameters during this conversion, the resulting $\mathsf{SPRS}^{\mathsf{qs}}$ 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_n is a $\perp$-PRG with expansion factor of 2, then the composition $G_{2n} \circ G_n$ is a $\perp$-OWSG with expansion factor 4. Hence, starting with a SPRS, building a $\perp$-PRG, extending the output length of the $\perp$-PRG sufficiently through composition, and finally converting this to a $\mathsf{SPRS}^{\mathsf{qs}}$, we obtain a method to convert a SPRS into a $\mathsf{SPRS}^{\mathsf{qs}}$ with larger output length.

Other Constructions. On the downside, we face an obstacle in the quantum input sampling regime when attempting to build a $\mathsf{PRF}^{\mathsf{qs}}$s from $\mathsf{PRG}^{\mathsf{qs}}$s. While a $\mathsf{PRG}^{\mathsf{qs}}$ with sufficient expansion easily implies a $\mathsf{PRF}^{\mathsf{qs}}$ with polynomial domain, it is not clear if a $\mathsf{PRG}^{\mathsf{qs}}$ can be used to build full-fledged $\mathsf{PRF}^{\mathsf{qs}}$ with exponential domain. Note that the standard conversion of a PRG to a PRF [14], and its quantum adaption [27], 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, $\mathsf{PRF}^{\mathsf{qs}}$ with polynomial domain can still be useful. Using domain extension techniques [11], we convert them into bounded-query $\mathsf{PRF}^{\mathsf{qs}}$s with exponential domain. These, in turn, enable the construction of bounded-copy linear-length $\mathsf{PRS}^{\mathsf{qs}}$ and bounded-query $\mathsf{PRU}^{\mathsf{qs}}$ following a similar approach outlined in [16, 20].

1.3.2 Separation Results We present two separation results in this work using quantum oracles. These separations complement the positive results discussed above.

Note that the oracles considered in these separations are *completely-positive-trace-preserving* (CPTP) maps. Specifically, our separations only rule out black-box reductions that do not make inverse queries to the oracles and that do not purify the oracles. Note that CPTP oracle separations have been considered in previous works [12, 13].

Separating OWSG*s from* $\perp$-PRG*s.* Firstly, we show that there does not exist a black-box construction of a OWSG from a $\perp$-PRG. This result is somewhat surprising, as OWSGs are considered weaker than PRGs, as evidenced by existing

[7] The output of the $\mathsf{PRG}^{\mathsf{qs}}$ can be interpreted as a $\mathsf{PRF}^{\mathsf{qs}}$ with polynomial domain.

separations between them [18]. 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 $\frac{1}{\mathsf{poly}(n)}$ fraction of inputs, the algorithm may return $\perp$ sometimes.

The result leverages two oracles. The first oracle is for membership in a PSPACE-complete language, used to break any OWSG. The second oracle, $\mathcal{O}$, is a modified quantum random oracle with an abort mechanism. This oracle exhibits inherent $\perp$-pseudodeterminism, rendering it unsuitable for constructing deterministic primitives like OWSGs.

The oracle $\mathcal{O}$ operates as follows: on input x, 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. Clearly, $\mathcal{O}$ functions as a $\perp$-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. In the case x is bad, the generator, running in polynomial-time, cannot distinguish between small variations in determinism errors. In other words, the output of the generator should remain constant under small perturbations of the determinism error of x. Generalizing this argument, we show that the generator's output must remain invariant even when the determinism error probability is 1, i.e. if $\Pr[\perp \leftarrow \mathcal{O}(x)] = 1$. Informally, this implies that the generator is independent of $\mathcal{O}$. This independence allows us to construct a OWSG that does not use the oracle $\mathcal{O}$, which should be impossible as OWSGs cannot exist in the presence of a PSPACE-oracle [7].

This separation is quite significant since $\perp$-PRGs have numerous strong applications [4], including (1) (many-time) digital signatures of classical messages and (2) quantum public-key encryption with tamper-resilient keys, showing that these primitives are separated from OWSGs as well.

Separating $\perp$-OWSGs from PRF^{qs}s Our second result separates $\perp$-OWSGs from $\mathsf{PRF}^{\mathrm{qs}}$s. This separation targets the distinction in quantum versus classical input sampling procedures.

The construction involves three oracles:

- $\mathcal{C}$: An oracle for membership in a PSPACE-complete language, independent of the other oracles.
- $\mathcal{R}$: A restricted-access random oracle, that can only be accessed with a valid key as a part of the input.
- σ: A quantum oracle that generates random keys. Each distinct key gives access to a distinct function in $\mathcal{R}$.

It is straightforward to construct a $\mathsf{PRF}^{\mathrm{qs}}$ relative to these oracles. The quantum key generation algorithm of the $\mathsf{PRF}^{\mathrm{qs}}$ samples a key by querying σ

and setting the key as the response. Meanwhile, the evaluation algorithm of the $\mathsf{PRF}^{\mathrm{qs}}$, on input x, uses the key to evaluate the random oracle $\mathcal{R}$ on x and returns the result. The same output of σ is used in each evaluation, so this yields deterministic evaluations from $\mathcal{R}$. Given that a quantum-accessible random oracle acts as a (quantum-query-secure) PRF [25], it is not difficult to show that this construction constitutes a $\mathsf{PRF}^{\mathrm{qs}}$.

The more difficult part is showing that $\perp$-OWSGs cannot exist relative to these oracles. Assume G is a $\perp$-OWSG relative to these oracles, for the purpose of obtaining a contradiction. The input sampled uniformly at random for G is independent of the oracles. Consequently, any evaluation of G receives distinct keys from queries to σ and, thus, distinct evaluations from $\mathcal{R}$.

To formalize this result, we simulate access to $(\sigma, \mathcal{R})$ in G using uniformly sampled random strings that are correlated in the same way as σ and $\mathcal{R}$. Specifically, since each evaluation of G receives different responses from σ and $\mathcal{R}$, this implies that the generator could simulate its computation by sampling random strings directly, rather than querying the oracles. As a result, there exists a $\perp$-OWSG without oracle access that is secure against adversaries with access to a PSPACE oracle. However, this contradicts a result from [7], which shows that there does not exist $\perp$-OWSG if PSPACE = BQP.

1.4 Organization

We briefly outline the structure of the paper. Section 2 reviews the required background material. Section 3 introduces MicroCrypt notions with quantum input sampling. Section 4 explores implications among various MicroCrypt primitives with quantum input sampling. Section 5 presents the separation results.

2 Preliminaries

2.1 Notations

We refer the reader to [24] for a detailed exposition to preliminary quantum information. We let $\mathcal{S}(\mathcal{H})$, $\mathcal{D}(\mathcal{H})$, and $\mathcal{U}(\mathcal{H})$ denote the set of unit vectors, density matrices, 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 $x \leftarrow X$ 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 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 $p(\lambda)$. We say that a quantum algorithm is *deterministic* if every input is mapped to a fixed output except with negligible probability with respect to the input length. We call a QPT algorithm A *non-uniform* if it consists of a

family $\{A_\lambda, \rho_\lambda\}_\lambda$ where $\{A_\lambda\}_\lambda$ are algorithms of polynomial run-time (not necessarily uniformly generated), such that for each λ, there is additionally a subset of input qubits for A_λ that are designated to be initialized with the density matrix ρ_λ of polynomial length. All the algorithms used in the security definitions are uniform unless mentioned otherwise. We also avoid using the λ subscript in algorithms to avoid excessive notation.

If A is quantum oracle algorithm and P is an algorithm with classical input space, then A^P means that A is given access to a polynomial number of classical oracle queries to P. Furthermore, we say A has *quantum-query access to* P to mean that it is given polynomial quantum queries to the unitary map $U_P : |x\rangle \to |x\rangle|P(x)\rangle$.

We let $[n] = \{1, 2, \ldots, n\}$, $[n : n + k] = \{n, n + 1, \ldots, n + k\}$, and $y_{[1:n]} = y_1 y_2 \ldots y_n$ for every $n, k \in \mathbb{N}$ and string y of length at least n. Furthermore, we let $\mathsf{negl}(x)$ denote any function that is asymptotically smaller than the inverse of any polynomial. 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 denote the density matrix of a quantum state in a register E as ρ_E and we let d_{TD} denote the total trace distance between two density matrices or two distributions.

2.2 Black-Box Separation

The notion of oracle black-box separations was first considered in [15] and later formalized in the quantum setting in [9]. We refer the reader to [9] for an exposition to this topic. Here, we just recall the definitions relevant for this work from [9], but adapted to CPTP oracles.

Definition 1. *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.

Let U be a CPTP implementation of $G \in P$. Then, we say that U is a CPTP implementation of P. We also say that quantum channel $\mathcal{A}$ P-breaks U to mean that $\mathcal{A}$ P-breaks G.

We now formalize the notion of constructions relative to a CPTP oracles.

[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.

Definition 2. *Let $\mathcal{O}$ be a CPTP oracle. An efficient implementation of primitive P relative to $\mathcal{O}$ is a QPT oracle algorithm $G^{(\cdot)}$ such that $G^{\mathcal{O}} \in P$. Let U be a CPTP implementation of $G^{\mathcal{O}}$. Then, we say that U is a CPTP implementation of P relative to $\mathcal{O}$.*

We are now ready to define the notion of black-box construction under CPTP access. All the black-box separations in this paper disprove the existence of black-box constructions under CPTP access.

Definition 3. *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 U of P, G^U is an implementation of Q.*
2. *There is a QPT algorithm $S(\cdot)$ such that, for every CPTP implementation U of P, every adversary $\mathcal{A}$ that Q-breaks G^U , and every CPTP implementation $\tilde{\mathcal{A}}$ of $\mathcal{A}$, it holds that $S^{\tilde{\mathcal{A}}}$ P-breaks U.*

2.3 MicroCrypt Primitives

We recall several MicroCrypt assumptions relevant to this work. First, we define 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 with high probability. This encompasses some earlier definitions, such as in [7], but is less general than other variants, such as in [23]. Note that our separation result only shows that there are no black-box constructions of a OWSG satisfying our definition from a $\perp$-PRG. In other words, we explicitly use this notion of correctness and the result does not hold for the more generalized notions, such as in [23]. We do not believe this to be a major issue since, to our knowledge, all known constructions of OWSGs satisfy our correctness condition.

Definition 4 (One-Way State Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be polynomial in λ. A QPT algorithm G is called a* n-one-way state generator (OWSG)*, if the following holds:*

- (Correctness): *There exists a negligible function $\delta = \delta(\lambda)$ such that for any $k \in \{0,1\}^\lambda$, there exists a n-qubit pure-state $|\phi_k\rangle$ such that $\Pr[G(k) = |\phi_k\rangle] \geq 1 - \delta$.*
- (Security) *For any polynomial $t = t(\lambda)$ and QPT distinguisher $\mathcal{A}$, there exists a function $\epsilon(\cdot)$ such that:*

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

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.

We define pseudorandom states (PRSs) from [16]. There are three size regimes for PRSs that are considered separately. Firstly, if the size of the PRS is $n = c \cdot \log(\lambda)$ with $c \ll 1$, then this can be constructed information-theoretically [6] and, therefore, is not very interesting cryptographically. Secondly, if $n = c \cdot \log(\lambda)$ with $c \geq 1$, then tomography can be efficiently performed to extract classical pseudorandomness, allowing for powerful applications [2]. Finally, larger states $n = \Omega(\lambda)$, where extracting a classical description is no longer feasible, are the final regime. This is the only regime that was separated from PRGs in previous work [18].

Definition 5 (Pseudorandom State Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be polynomial in λ. A QPT algorithm PRS is called a* n-pseudorandom state generator (PRS)*, if the following holds:*

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

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

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 (post-quantum) PRGs and PRFs. We consider only the versions of PRG and PRF with quantum evaluation algorithms in this work.

Definition 6 (Pseudorandom Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be polynomial in λ. A 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 $\mathcal{A}$:*

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

Definition 7 (Pseudorandom Function). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $m = m(\lambda)$ be polynomial in λ. A QPT deterministic algorithm F taking a key in $\{0,1\}^\lambda$ and mapping $\{0,1\}^m$ to $\{0,1\}^m$ is called a* m-pseudorandom

function (PRF), *if for any QPT distinguisher* $\mathcal{A}$*:*

$$\left| \Pr_{k\leftarrow\{0,1\}^\lambda}[\mathcal{A}^{F_k}(0^\lambda)=1] - \Pr_{O\leftarrow\Pi_{m,m}}[\mathcal{A}^{O}(0^\lambda)=1] \right| \leq \mathsf{negl}(\lambda).$$

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

Pseudorandom states can be built from any (quantum-query-secure) PRF [16].

Lemma 1 (Theorem 1 [16]). *For any (quantum-query-secure)* $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ *where* $\mathcal{X} : \{1,\ldots,N\}$*, the family of states* $\{|\phi\rangle\}_{k\in\mathcal{K}}$ *defined as follows is a* $(\log N)$-PRS*:*

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

where $\omega_N \coloneqq e^{-i2\pi/N}$*.*

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* $t = t(\lambda) \coloneqq 36\lambda d^3/\delta$ *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*

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

We now recall how Tomography is used to extract pseudorandom strings in [2].

Construction 1 (Extract [2]). *Let* $\lambda \in \mathbb{N}$ *be the security parameter, and let* $d \coloneqq d(\lambda)$ *and* $t \coloneqq t(\lambda)$ *be polynomials on* λ*. The algorithm* Extract *is defined as follows:*

- *Input:* $t \coloneqq 144\lambda d^8$ *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* $y \in \{0,1\}^{\ell}$.

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

– Define $k(\lambda) := d^{5/6}$, $r(\lambda) := d^{2/3}$, and $\ell(\lambda) := d^{1/6}$.
– 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 [2]). *Define*

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

Then,

1. $\Pr[|\psi\rangle \in \mathcal{G}_d : |\psi\rangle \leftarrow \mathsf{Haar}(\mathbb{C}^d)] \geq 1 - O(d^{-1/6})$.
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* $\Pr[y \leftarrow \mathsf{Extract}(|\psi\rangle)] \geq 1 - \mathsf{negl}(\lambda)$.

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 recall this notion and introduce a similar relaxation to OWSGs, which we term $\perp$-OWSGs.

Definition 8 ($\perp$ One-Way State Generator). *Let* $\lambda \in \mathbb{N}$ *be the security parameter and let* $n = n(\lambda)$ *be a polynomial in* λ. *A QPT algorithm* G *is called a* (μ, n)-$\perp$-pseudodeterministic OWSG ($\perp$-OWSG), *if the following holds:*

– (Correctness):

1. *For any $x \in \{0,1\}^\lambda$, there exists a non-$\perp$ n-qubit pure-state $|\psi_x\rangle$ such that:*

$$\Pr[G(x) \in \{|\psi_x\rangle, \perp\}] \geq 1 - \mathsf{negl}(\lambda).$$

2. *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-$\perp$ n-qubit state $|\psi_x\rangle$ such that:
   $$\Pr[G(x) = |\psi_x\rangle] \geq 1 - \mathsf{negl}(\lambda). \quad (1)$$

– (Security) *For any polynomial $t = t(\lambda)$ and QPT distinguisher $\mathcal{A}$, there exists a function $\epsilon(\cdot)$ such that:*

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

We say that G is a $\perp$-OWSG if for every QPT $\mathcal{A}$, ϵ is a negligible function. We say that G is a weak $\perp$-OWSG *if for every QPT $\mathcal{A}$, $\epsilon \leq \frac{1}{p}$ for some polynomial p. Furthermore, in the case where security only holds for $t \leq q$ for some polynomial q, then we call G a q-copy $\perp$-OWSG.*

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

1. Sample $k \leftarrow \{0,1\}^\lambda$.
2. For each $i \in [t+1]$ generate $|\psi_i\rangle \leftarrow G(k)$ where $|\psi_i\rangle \in \{\perp, |\phi_k\rangle\}$.
3. $k' \leftarrow \mathcal{A}(\otimes_{i\in[t]}|\psi_i\rangle)$. Let $|\phi_{k'}\rangle \leftarrow G(k')$.
4. If $|\psi_{t+1}\rangle = \perp$ or $|\phi_{k'}\rangle = \perp$, then output $b = 0$.
5. 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.

We will also need a weaker notion which we call *psuedodeterministic* OWSG.

Definition 9 (Pseudodeterministic One-Way State Generator). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n = n(\lambda)$ be a polynomial in λ. A QPT algorithm G is called a* (μ, n)-pseudodeterministic OWSG (PD-OWSG)*, if the following holds:*

– (Correctness): *$G(x)$ generates a n-qubit pure-state on any input $x \in \{0,1\}^\lambda$. Furthermore, 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:*
   1.
   $$\Pr_{x \leftarrow \{0,1\}^\lambda}[x \in \mathcal{G}_\lambda] \geq 1 - \mu(\lambda).$$

2. For every $x \in \mathcal{G}_\lambda$, there exists a non-$\bot$ n-qubit state $|\psi_x\rangle$ such that:

$$\Pr\left[G(x) = |\psi_x\rangle\right] \geq 1 - \mu(\lambda). \tag{2}$$

– (Security) *For any polynomial $t = t(\lambda)$ and QPT distinguisher $\mathcal{A}$, there exists a function $\epsilon(\cdot)$ such that:*

$$\mathsf{Adtg}^{\mathsf{PD\text{-}OWSG}}_{\mathcal{A},G}(1^\lambda, 1^t) \coloneqq \Pr[\mathsf{Exp}^{\mathsf{PD\text{-}OWSG}}_{\mathcal{A},G}(1^\lambda, 1^t) = 1] \leq \epsilon(\lambda).$$

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

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

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.

We prove the following lemmas regarding PD-OWSGs that will be useful in our separation results. These are largely derived from previously known results on OWSGs, so we only give sketches of the proofs.

Lemma 5. *Let n, q, and p be polynomials in the security parameter $\lambda \in \mathbb{N}$. Let $\mu = O(\lambda^{-c})$ for some constant c such that $\mu < \frac{1}{\lambda^2 q}$. Let $p > 2 \cdot q \cdot \lambda$ be a polynomial in λ. If $(1/q)$-weak p-copy (μ, n)-PD-OWSGs exist, then $(\frac{p}{q\lambda})$-copy $(\mu \cdot q \cdot \lambda, n \cdot q \cdot \lambda)$-PD-OWSGs exist.*

Proof. [sketch] Security follows in the same way as the proof given in [22], which converts weak OWSGs into OWSGs. For the pseudodeterminism error and the number of queries, note that the construction of the OWSG in this proof queries the weak OWSG $q \cdot \lambda$ times. Hence, the pseudodeterminism error is now $\mu \cdot q \cdot \lambda$ and the generator is only secure given $\frac{p}{q\lambda}$ queries. □

Lemma 6. *Let n and p be polynomials in the security parameter $\lambda \in \mathbb{N}$ and let $\mu = O(\lambda^{-c})$ for some constant $c > 1$. If kn-copy (μ, n)-PD-OWSGs exist for sufficiently large constant $k > 0$, then OWPuzzs exist.*

Proof. [sketch] Theorem 2.6 in [17] constructs OWPuzzs from $(k'n)$-copy OWSGs for some $k' > 0$. For a PD-OWSG G, the conversion only satisfies correctness when applied on a good key $x \in \mathcal{G}_\lambda$. However, by standard amplification arguments through repetition, this correctness error can be reduced to negligible. □

[7] show that if PSPACE = BQP, then OWPuzzs do not exist. In fact, the attack presented succeeds in finding a solution to a puzzle with probability 1.

Lemma 7 ([7]). OWPuzzs *do not exist if* PSPACE = BQP.

We introduce $\perp$-PRGs now. Note that it is easy to distinguish these functions from random by simply looking for any $\perp$ evaluations. 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:
   $$\Pr[G(x) = y] \geq 1 - \mathsf{negl}(\lambda). \tag{3}$$
   (c) For every $x \in \{0,1\}^\lambda$, there exists a non-$\perp$ value $y \in \{0,1\}^m$ such that:
   $$\Pr[G(x) \in \{y, \perp\}] \geq 1 - \mathsf{negl}(\lambda). \tag{4}$$
3. (Security) *For every QPT distinguisher $\mathcal{A}$, there exists a negligible function ϵ such that:*

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

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

Lemma 8 (Corollary 1 [4]). *If there exists $(c \log \lambda)$-*SPRS *for some constant $c > 12$, then there exists a $(O(\lambda^{-c/12+1}), \lambda^{c/12})$ - $\perp$-*PRG.

We note that [4] also constructed a $\perp$-pseudodeterministic pseudorandom function ($\perp$-PRF) from a $\perp$-PRG. We refer the reader to [4] for the explicit definition of $\perp$-PRFs as it is not required in this work.

3 Definitions: Cryptography with Quantum Input Sampling

We present definitions for various known cryptographic primitives but with quantum input sampling algorithms.

Note that any classical input sampling algorithm can be derandomized and replaced with uniform key sampling. However, a quantum procedure cannot be derandomized and might include non-deterministic quantum computations. Delaying these computations to the evaluation or state generation step may lead to a different outcome each time, which is problematic for deterministic notions such as PRSs or OWFs. Hence, we include a QPT algorithm QSamp to sample inputs.

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

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

*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 $\mathsf{PRS}^{\mathrm{qs}}$ ($\mathsf{BC\text{-}PRS}^{\mathrm{qs}}$).

We now introduce PRG with quantum input sampling. As far as we know, this notion 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 (n, m)-*PRG with quantum input sampling ($\mathsf{PRG}^{\mathrm{qs}}$) *if:*

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

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

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

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

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

We now introduce PRFs with quantum key generation.

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 $y\in\{0,1\}^n$.*
3. (Security) *For any QPT distinguisher $\mathcal{A}$:*

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

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}.*

Definition 15 (Pseudorandom Permutation with Quantum Key Generation). *Let $\lambda\in\mathbb{N}$ be the security parameter and let $n=n(\lambda)$ be polynomial in λ. A tuple of QPT algorithms $(\mathsf{QSamp},F,F^{-1})$ is called a* (m,n)-pseudorandom permutation with quantum key generation ($\mathsf{PRP}^{\mathrm{qs}}$)*, 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 $x,y\in\{0,1\}^n$*

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

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 $\mathcal{A}$:*

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

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 $\mathcal{A}$,*

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

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 [16], 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.

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. $\mathsf{PRG}^{\mathrm{qs}}$ from $\perp$-PRG.
2. $\mathsf{PRG}^{\mathrm{qs}}$ from BC-$\mathsf{SPRS}^{\mathrm{qs}}$.
3. $\mathsf{SPRS}^{\mathrm{qs}}$ from $\mathsf{PRG}^{\mathrm{qs}}$.
4. BQ-$\mathsf{PRU}^{\mathrm{qs}}$ from $\mathsf{PRG}^{\mathrm{qs}}$

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 $\bot$-PRG

We show how to build $\mathsf{PRG}^{\mathsf{qs}}$s from $\bot$-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 $\lambda < m$. Let $\mu = O(\lambda^{-c})$ for some constant $c > 0$. Let G be a (μ, m)-$\bot$-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 $\mathsf{vote}(y_{i,1}, \ldots, y_{i,\lambda}) \neq \bot$ [9], then output k_i.*

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

Lemma 9. *Construction 2 is a weak (λ, m)-PRG^{qs} assuming the existence of a (μ, m)-$\bot$-PRG.*

Proof. We first show that the algorithm PRG satisfies the determinism condition of $\mathsf{PRG}^{\mathsf{qs}}$s. Firstly, by the pseudodeterminism condition of the $\bot$-PRG G, there is negligible probability that $\mathsf{QSamp}(1^\lambda)$ outputs $\bot$. Furthermore, for any $k \in \{0,1\}^\lambda$, if $\Pr[G(k) = \bot] \geq \frac{2}{3}$, then there is negligible probability that $\mathsf{vote}(y_1, \ldots, y_\lambda) \neq \bot$, 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 key k such that $\Pr[G(k) = \bot] \geq \frac{2}{3}$. Therefore, except with negligible probability, the output of $\mathsf{QSamp}(1^\lambda)$ is a non-$\bot$ key k such that $\Pr[G(k) = \bot] < \frac{2}{3}$. It is clear that for such a key, $\mathsf{PRG}(k)$ is deterministic by the pseudodeterminism of G.

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.
  - $k \leftarrow \mathcal{G}_\lambda$.

[9] $\mathsf{vote}(a_1, \ldots, a_n)$ outputs the first most common element in the tuple $(a_1, \ldots, a_n)$.

- For each $j \in [\lambda]$, compute $y_j \leftarrow G(k)$.
- If $y_j = \perp$ for all $j \in [\lambda]$, then output $\perp$.
- Otherwise, output the first most common non-$\perp$ 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 $\perp$ random string.
- $k \leftarrow \mathcal{G}_\lambda$.
- $y \leftarrow \{0,1\}^m$.
- Output $\mathsf{ls}\text{-}\perp(G(k), y)$.

– 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:*

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

Proof. Recall the algorithm of $\mathsf{QSamp}(1^\lambda)$ is as follows:
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 $\mathsf{vote}(y_{i,1}, \ldots, y_{i,\lambda}) \neq \perp$, then output k_i.

Note that any input in the good set $\mathcal{G}_\lambda$ of G, yields a deterministic output with high probability, so if $k_1 \in \mathcal{G}_\lambda$ in the algorithm of QSamp, then the output is k_1 with high probability. 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 $\mathsf{vote}(y_1, \ldots, y_\lambda) = y$, 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 $\mathcal{A}$:*

$$\left|\Pr_{y\leftarrow H_2}[\mathcal{A}(y)=1]-\Pr_{y\leftarrow H_3}[\mathcal{A}(y)=1]\right|\leq 2\mu+\mathsf{negl}(\lambda).$$

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

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

Finally, by the pseudodeterminism property of G,

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

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 $\bot$-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 [10] that a weak PRG can be upgraded to a strong PRG through a standard amplification argument. This argument applies to $\mathsf{PRG}^{\mathrm{qs}}$s, giving the following result.

Theorem 3. *If there exists (μ,m)-$\bot$-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 $c > 12$ is a constant. Let $(\mathsf{QS}, \mathsf{PRS})$ be a $(200\lambda d^8, \lambda, n)$-$\mathsf{BC\text{-}SPRS}^{qs}$. The construction for a (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. *Compute $y \leftarrow \mathsf{Extract}(k)$.*
  3. *Output y.*

Theorem 4. *Construction 3 satisfies weak (λ, m)-PRG^{qs} security assuming the existence of a $(200\lambda d^8, \lambda, n)$-$\mathsf{BC\text{-}SPRS}^{qs}$.*

Proof. We first show that the algorithm G satisfies the determinism condition of a $\mathsf{PRG}^{\mathrm{qs}}$. Firstly, by the pseudodeterminism condition of the $\perp$-PRG G, there is negligible probability that $\mathsf{QSamp}(1^\lambda)$ outputs $\perp$.

The other possibility is that $\mathsf{QSamp}(1^\lambda)$ outputs a key s satisfying $M_s \in \mathcal{G}_\lambda$ where $M_s \leftarrow \mathsf{Tomography}(\rho_i^{\otimes t})$ and $\rho_i \leftarrow \mathsf{PRS}(s)$ for all $i \in [t]$. In this case, [4] argues using Lemmas 2 and 4 that $G(s)$ yields a unique output except with negligible probability. However, this was only shown for SPRSs, or random Haar states, and not for $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$s. The quantum input sampling in $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$ does not affect this argument, however, the limited copies may affect it.

It is easy to argue that the limited copies have no affect on this determinism property. By our assumption, $\mathsf{PRS}(s)$ is indistinguishable from a random Haar state even given $200\lambda d^8 > 2t$ copies. In other words, if s satisfies $M_s \in \mathcal{G}_\lambda$ in one computation but two computations of $G(s)$ do not agree with non-negligible probability, then PRS can be distinguished from a random Haar states as follows. The distinguisher uses $2t$ copies of the state $\rho_i^{\otimes 2t}$, where $\rho_i \leftarrow \mathsf{PRS}(s)$ for all $i \in [2t]$, to compute $G(s)$ twice. If $M_s \in \mathcal{G}_\lambda$ in the first computation but the two evaluations do not agree, then the distinguisher can distinguish the states from random Haar states and break $\mathsf{BC\text{-}SPRS}^{\mathrm{qs}}$ security. Therefore, except with negligible probability, if $M_s \in \mathcal{G}_\lambda$, then two evaluations of $G(s)$ need to yield a unique output. So G satisfies the determinism condition of $\mathsf{PRG}^{\mathrm{qs}}$s.

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

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

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 $k := y$.

    Otherwise, output $\perp$.
  - $G_{\mathsf{H}_3}(k)$: 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, $\mathsf{QSamp}_{\mathsf{H}_2}$ in hybrid H_2 samples a Haar random state that is in the set $\mathcal{G}_\lambda$ or it returns $\perp$. However, the latter occurs with negligible probability, so it is indistinguishable from sampling a Haar random state in $\mathcal{G}_\lambda$. Therefore, by Lemma 4, H_2 and H_3 can only be distinguished with $O(d^{-1/6})$ probability.

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:

$$\begin{aligned}
&\left|\Pr_{k\leftarrow\mathsf{QSamp}(1^\lambda)}[\mathcal{A}(G(k))=1]-\Pr_{y\leftarrow\{0,1\}^m}[\mathcal{A}(y)=1]\right|\\
&=\left|\Pr_{y\leftarrow\mathsf{H}_0}[\mathcal{A}(y)=1]-\Pr_{y\leftarrow\mathsf{H}_1}[\mathcal{A}(y)=1]\right|\\
&\leq O(d^{-1/6}).
\end{aligned}$$

□

Recall that weak $\mathsf{PRG}^{\mathrm{qs}}$ can be upgraded to a strong $\mathsf{PRG}^{\mathrm{qs}}$ following the same argument in [10]. Hence, we obtain the following result.

Theorem 5. *If there exists* $(200\lambda d, \lambda, n)$-$\mathsf{BC\text{-}SPRS}^{qs}$*, then there exists a* (λ^2, m)-PRG^{qs} *satisfying strong security.*

Note that $\perp$-PRGs were used in [4] to construct many-time digital signatures and quantum public-key encryption with tamper-resilient public-keys. However, it is not clear if these applications can be achieved from $\mathsf{PRG}^{\mathrm{qs}}$s since the arguments used intricately rely on the distribution of inputs for the $\perp$-PRG. Our quantum input sampling samples inputs in a more complex manner that may affect these results. Therefore, $\mathsf{PRG}^{\mathrm{qs}}$ may be more difficult to use than $\perp$-PRGs in certain cryptographic applications. However, in the next section, we show that $\mathsf{PRG}^{\mathrm{qs}}$s can still be used for $\mathsf{SPRS}^{\mathrm{qs}}$.

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* $f_y : \mathcal{X} \to \mathcal{X}$ [10]*.*
  3. *Output*

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

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

Theorem 6. *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 6 follows directly from Lemma 1. Specifically, Lemma 1 states that a n-SPRS can be built from a PRF with domain size 2^n. This conversion applies to the quantum input sampling regime as well. Then, Theorem 6 follows by setting $n := c \cdot \log \lambda$ and noting that f_y in Construction 4 is a PRF.

Theorem 5 states that we can construct a $\mathsf{PRG}^{\mathrm{qs}}$ from any $\mathsf{SPRS}^{\mathrm{qs}}$. On the other hand, Theorem 6 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 SPRS.

Corollary 3. *For any sufficiently large* $c \in \mathbb{N}$ *and any* $m < c/36$, $(\lambda, c \log \lambda)$-$\mathsf{SPRS}^{qs}$ *implies* $(\lambda, m \log \lambda)$-$\mathsf{SPRS}^{qs}$.

Furthermore, by starting with a SPRS, building a $\perp$-PRG [4], amplifying the output length sufficiently [11], building a $\mathsf{PRG}^{\mathrm{qs}}$ (Theorem 3), and finally a $\mathsf{SPRS}^{\mathrm{qs}}$, we obtain a $\mathsf{SPRS}^{\mathrm{qs}}$ of larger size.

Corollary 4. *For any sufficiently large* $c \in \mathbb{N}$ *and any* $m > c$, $(\lambda, c \log \lambda)$-SPRS *implies* $(\lambda, m \log \lambda)$-$\mathsf{SPRS}^{qs}$.

4.4 BQ-PRU$^{\mathrm{qs}}$ from PRG$^{\mathrm{qs}}$

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 [14] and its quantum adaption [27] 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 10. Specifically, the paper [11] 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 10 (Theorem 7 [11]). *Let* $\lambda \in \mathbb{N}$ *be the security parameter and* q *and* m *be polynomials in* λ. *Let* (QSamp, F) *be a* 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, [26] 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 and input length.

[11] It is easy to amplify the output length of a $\perp$-PRG by re-applying the algorithm on the output.

Hence, the same proof can be used to show that for any $q' \in \mathsf{poly}(\lambda)$, there exists a $q \in \mathsf{poly}(\lambda)$ such that q-query pseudorandom functions imply q'-query pseudorandom permutations.

Corollary 5. *Let $\lambda \in \mathbb{N}$ be the security parameter and $q = q(\lambda)$ and $m = m(\lambda)$ be polynomials in λ. There exists a polynomial $\ell = \ell(\lambda)$, such that (λ, ℓ)-PRG^{qs}s imply (q, λ, m)-$\mathsf{BQ\text{-}PRP}^{qs}$s.*

Recently, [20] 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 7. *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)-$\mathsf{BC\text{-}LPRS}^{qs}$.*

5 Black-Box Separations Among MicroCrypt

In this section, we demonstrate two separation results and discuss implications.

5.1 Separating OWSG from $\perp$-PRG

First, we show that there does not exist a black-box construction of a OWSGs from a $\perp$-PRG. The proof idea is as follows. We consider two oracles: a oracle for a PSPACE-complete language and a $\perp$-pseudodeterministic quantum oracle. Through standard arguments, we show that the second 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 second oracle prevents its use in the construction of deterministic primitives such as a OWSG. Specifically, through a careful argument, we show that any OWSG must exist independently of the second oracle and, thus, cannot exist in the presence of a PSPACE oracle. Given this distinction, there cannot exist a black-box construction of a OWSG from a $\perp$-PRG. The proof is given in Appendix A.1, but the result is stated below.

Theorem 8. *Let $\lambda \in \mathbb{N}$ be the security parameter. For any function $w(\lambda)$, polynomial $m(\lambda) > \lambda$, and pseudodeterminism error $\mu(\lambda) = O(\lambda^{-c})$ for some $c > 0$, there does not exist a black-box construction of a w-OWSG from 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 exist black-box constructions of the following primitives from* $\bot$-PRG*s:*

1. $\bot$-PRF*s.*
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 8 and Lemma 11, we obtain a separation between OWSGs and some cryptographic applications.

Corollary 6. OWSG*s are black-box separated from:*

1. $\bot$-PRF*s.*
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 Separating $\bot$-OWSG from PRF$^{\text{qs}}$

We show that there does not exist a black-box construction of a $\bot$-OWSG from a PRF$^{\text{qs}}$. The idea is to consider three oracles. The first is a restricted-access oracle $\mathcal{O}$, that can only be accessed with a key as input. This oracle will acts as a PRF$^{\text{qs}}$. Next, we use another oracle σ that 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 PRF$^{\text{qs}}$, 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 $\bot$-OWSG is incapable of obtaining deterministic evaluations from $\mathcal{O}$. This informally means that any $\bot$-OWSG cannot depend on $(\sigma, \mathcal{O})$ and, thus, cannot exist in the presence of a PSPACE oracle. Therefore, there does not exist a black-box construction of a $\bot$-OWSG from a PRF$^{\text{qs}}$. The proof is given in Appendix A.1, but the result is stated below.

Theorem 9. *Let* $\lambda, n \in \mathbb{N}$ *be security parameters,* $a = a(n) \geq n$ *and* $b = b(n)$ *be polynomials,* $s = s(\lambda)$ *be a function, and* $\mu = \mu(\lambda) \leq \lambda^{-c}$ *be a function. There does not exist a black-box construction of a* (μ, s)-$\bot$-OWSG *from a* (a, b)-PRF^{qs} *for large enough constant* $c > 0$.

The following lemma generalizes the applications of PRFs in cryptography [16, 2, 4] to PRF$^{\text{qs}}$s. These follow in the same way as using quantum key generation rather than uniform key generation does not affect the proofs.

Lemma 12. *There exists 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 Theorem 9 and Lemma 12, we obtain a separation between $\perp$-OWSGs and many other MicroCrypt primitives. We note that this separation extends to $\perp$-PRGs and SPRSs since they imply $\perp$-OWSGs [4, 23].

Corollary 7. *$\perp$-OWSGs, $\perp$-PRGs, and SPRSs are black-box separated from:*

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.*

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

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

[7] 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).

[8] 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.

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

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] William Kretschmer, Luowen Qian, and Avishay Tal. "Quantum-Computable One-Way Functions without One-Way Functions". In: *arXiv preprint arXiv:2411.02554* (2024).
[20] Fermi Ma and Hsin-Yuan Huang. "How to construct random unitaries". In: *arXiv preprint arXiv:2410.10116* (2024).
[21] Tomoyuki Morimae, Shogo Yamada, and Takashi Yamakawa. *Quantum Unpredictability*. Cryptology ePrint Archive, Paper 2024/701. 2024. URL: https://eprint.iacr.org/2024/701.
[22] Tomoyuki Morimae and Takashi Yamakawa. "One-wayness in quantum cryptography". In: *arXiv preprint arXiv:2210.03394* (2022).
[23] 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.
[24] 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.
[25] 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.
[26] Mark Zhandry. "A note on quantum-secure PRPs". In: *arXiv preprint arXiv:1611.05564* (2016).
[27] Mark Zhandry. "How to construct quantum random functions". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE. 2012, pp. 679–687.

A Proof of Separation Results

A.1 Separating OWSG from ⊥-PRG

To demonstrate the separation between OWSGs and ⊥-PRGs, we will use two independent oracles. The first oracle $\mathcal{C}$ answers membership queries to a PSPACE-complete language, while the second oracle $\mathcal{O}$ is a ⊥-pseudodeterministic quantum random oracle.

Let $\lambda \in \mathbb{N}$ be the security parameter. 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 5. *Fix a pseudodeterminism error $\mu(n) = O(n^{-c})$. 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 $p_x \coloneqq q/2^n$, where q is interpreted as an integer in $[1 : 2^n]$.*
– *Compute $z = P_n(x)$. If the first w-bits of z are 0^w, then let $|\phi_x\rangle \coloneqq \sqrt{p_x}|\bot\rangle + \sqrt{1-p_x}|y\rangle$.*
– *Otherwise, let $|\phi_x\rangle \coloneqq |y\rangle$.*
– *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:

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

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

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

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

– *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. $\Pr[y \leftarrow \mathcal{O}_n(x)] = 1 - p_x$.
  2. $\Pr[\bot \leftarrow \mathcal{O}_n(x)] = p_x$.

We will now show that there does not exist a black-box construction of a OWSG from a $\bot$-PRG.

Theorem 10. *Let $\lambda \in \mathbb{N}$ be the security parameter. For any function $w(\lambda)$, polynomial $m(\lambda) > \lambda$, and pseudodeterminism error $\mu(\lambda) = O(\lambda^{-c})$ for $c > 0$, there does not exist a black-box construction of a w-OWSG from a (μ, m)-$\bot$-PRG.*

Proof. Assume for contradiction that there exists a black-box construction of a w-OWSG G^F from any (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 5. *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 (μ, m)-$\bot$-PRG for every possible sequences P and Q and with probability 1 over the distribution of O. Furthermore, correctness is satisfied for any oracle.*

Proof. By Lemma 13, $\mathcal{O}$ 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 $\mathcal{A}$:

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

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 [25] states that a random oracle acts as a PRG i.e.:

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

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:

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

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,

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

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

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

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

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

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 that $G^{\mathcal{O}[P,Q,O]}$ is a OWSG relative to $(\mathcal{O}, \mathcal{C})$ for any P, Q and uniformly sampled O, with probability 1.

Claim 6. *For any QPT adversary $\mathcal{A}$ and polynomial $t = t(\lambda)$, there exists a negligible function δ such that:*

$$\Pr_{\mathcal{O}}[\mathit{Adtg}^{\mathit{OWSG}}_{\mathcal{A}^{(\mathcal{O},\mathcal{C})},G^{\mathcal{O}}}(1^\lambda, 1^t) \leq \delta(\lambda)] \geq \frac{1}{\lambda^2}.$$

where the probability is taken over the oracle distribution.

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

$$\Pr_{\mathcal{O}}[\mathsf{Adtg}^{\mathsf{OWSG}}_{\mathcal{A}^{(\mathcal{O},\mathcal{C})},G^{\mathcal{O}}}(1^\lambda, 1^t) > \frac{1}{p(\lambda)}] > 1 - \frac{1}{\lambda^2}$$

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

By Borel-Cantelli Lemma, since $\sum_\lambda \frac{1}{\lambda^2}$ converges, with probability 1 over the distribution of $\mathcal{O}$, it holds that $\mathsf{Adtg}^{\mathsf{OWSG}}_{\mathcal{A}^{(\mathcal{O},\mathcal{C})},G^{\mathcal{O}}}(1^\lambda, 1^t) > \frac{1}{p(\lambda)}$ for infinitely many $\lambda \in \mathbb{N}$.

In other words, $\mathcal{A}$ is successful in breaking the security of $G^{\mathcal{O}}$ with probability 1 over the oracle distribution. Therefore, $G^{\mathcal{O}}$ is not a OWSG with probability 1 over the oracle distribution, giving a contradiction. □

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 7. *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. $P^i_n \in \mathit{\Pi}_n$, $Q^i_n \in \mathit{\Pi}_{n,n}$, and $O^i_n \in \mathit{\Pi}_{n,m}$ 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 $i \in [\ell]$,

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

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 modify Q^1_n by setting $Q^2_n(x)$ to $Q^1_n(x)+1$ or $Q^1_n(x)-1$, 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^j_n, that sends all values to 1^n, while keeping all other functions fixed. Then, we set $O^{j+1}_n(x) = O''_n(x)$ for all x such that $P_n(x)_{[1:w]} = 0^w$ and keep $O^{j+1}_n(x) = O^j_n(x)$ otherwise. This step does not change the oracle, i.e. $\mathcal{O}^j = \mathcal{O}^{j+1}$, because Q^j_n and Q^{j+1}_n return 1^n on these inputs so both oracles return $\perp$.

Next, we perform small changes to Q^{j+1}_n until we reach a function, say Q^t_n for some $t > j$, that sends all values to 0^n. Then, we set P^{t+1}_n 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^t_n and Q^{t+1}_n return 0^n on any input. The new function P^{t+1}_n allows us to perform the first step on a new set of inputs i.e. we can modify O^{t+1}_n on all inputs such that $P^{t+1}_n(x)_{[1:w]} = 0^w$. 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 $P^i_n \in \Pi_n$, $Q^i_n \in \Pi_{n,n}$, and $O^i_n \in \Pi_{n,m}$ for any $n \in \mathbb{N}$, by Claim 5, $G^{\mathcal{O}^i}$ is deterministic for any $i \in [\ell]$.

Note that for any $i \in [\ell]$, $\mathcal{O}_i$ is an oracle with classical output and $d_{\mathrm{TD}}(\mathcal{O}^i, \mathcal{O}^{i+1}) \leq \frac{1}{m}$. Therefore, with probability at least $1 - \frac{r}{m}$, 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, 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 $(\tilde{P}_n, \tilde{Q}_n, \tilde{O}_n)_{n\leq\log(2m)} = (P_n, Q_n, O_n)_{n\leq\log(2m)}$,

$$\Pr[G^{\mathcal{O}[\tilde{P},\tilde{Q},\tilde{O}]}(k) = \left|\psi_k^{\mathcal{O}_{\leq\log(2m)}}\right\rangle] \geq 1 - \mathsf{negl}(\lambda). \tag{5}$$

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 5 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.
- Output the result of $G(k_1)$.

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 5.
  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 5.
  2. $b \leftarrow \mathsf{Exp}^{\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{C}},G^{\mathcal{O}}}(1^\lambda, 1^t)$. *Notice that $\mathcal{A}$ no longer has access to $\mathcal{O}$ in this experiment or it is simply be replaced with a oracle that* 0.
  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 6, for any QPT adversary $\mathcal{A}$,

$$\Pr_{\mathcal{O}}[\mathsf{Adtg}^{\mathsf{OWSG}}_{\mathcal{A}^{(\mathcal{O},\mathcal{C})},G^{\mathcal{O}}}(1^\lambda, 1^t) \leq \delta(\lambda)] \geq \frac{1}{\lambda^2}.$$

Therefore,

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

Next, the only difference between $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$ and $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$ is that $\mathcal{A}$ is no longer given access to $\mathcal{O}$. Hence, for any QPT adversary $\mathcal{A}$, there exists a QPT $\mathcal{A}'$ such that

$$\Pr[\mathsf{Exp}_2^{\mathcal{A}}(\lambda) = 1] \leq \Pr[\mathsf{Exp}_1^{\mathcal{A}'}(\lambda) = 1] \leq 1 - \frac{1}{\lambda^2} + \delta \cdot \frac{1}{\lambda^2}$$

Note that the functions $(\mathcal{O}_n^{k_2})_{n\leq\log(2m)}$ used in $\mathsf{Exp}_3^{\mathcal{A}}$ have the same distribution as the functions $(\mathcal{O}_n)_{n\leq\log(2m)}$ sampled in $\mathsf{Exp}_2^{\mathcal{A}}$. Furthermore, if $(\mathcal{O}_n^{k_2})_{n\leq\log(2m)} = (\mathcal{O}_n)_{n\leq\log(2m)}$, then for any key $k_1 \in \{0,1\}^\lambda$, by Eq. (5), $\overline{G}(k_1,k_2)$ produces the same output as $G^{\mathcal{O}}(k_1)$ except with negligible probability. Therefore, any QPT adversary $\mathcal{A}$ cannot distinguish between $\mathsf{Exp}_2^{\mathcal{A}}$ and $\mathsf{Exp}_3^{\mathcal{A}}$ except with negligible probability, meaning that for large enough λ,

$$\Pr[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1] \leq \Pr[\mathsf{Exp}_2^{\mathcal{A}}(\lambda)=1] + \mathsf{negl}(\lambda) \tag{6}$$

$$\leq 1 - \frac{1}{\lambda^2} + \delta\cdot\frac{1}{\lambda^2} + \mathsf{negl}(\lambda) \leq 1 - \frac{1}{\lambda^3} \tag{7}$$

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, [7] presents an attack against a OWSG using a PSPACE oracle. In particular, they show that there exists an adversary $\overline{\mathcal{A}}$ such that

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

contradicting Eq. (7) above.

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

A.2 Separating $\bot$-OWSG from $\mathsf{PRF}^{\mathsf{qs}}$

In this section, we show that there does not exist a black-box construction of a $\bot$-OWSG from a $\mathsf{PRF}^{\mathsf{qs}}$. The result can be seen more generally as a method to separate deterministic MicroCrypt primitives with quantum input sampling from those utilizing classical input sampling. We choose to apply it to $\mathsf{PRF}^{\mathsf{qs}}$ and $\bot$-OWSG as this seems to give the strongest separation: (as far as we know) all $\bot$-pseudodeterministic MicroCrypt primitives with uniform input sampling imply $\bot$-OWSG, while $\mathsf{PRF}^{\mathsf{qs}}$ implies MicroCrypt primitives with quantum input sampling.

We first describe the oracles used in the separation.

Construction 6. *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 $\mathcal{C} := \{\mathcal{C}_n\}_{n\in\mathbb{N}}$. For $n\in\mathbb{N}$, let $O_n \leftarrow \Pi_{n,n}$ and $P_n \leftarrow \Pi_{3n,n}$ be random functions. The oracles are defined as follows:*

1. *$\mathcal{C}$ is for membership in a PSPACE-complete language and is independent of the other oracles.*
2. *$\sigma_n(1^n)$:*
   (a) *Sample $x \leftarrow \{0,1\}^n$.*
   (b) *Output $|x, O_n(x)\rangle$.*
3. *$\mathcal{O}_n$ takes as input a state ρ_{XYAB} where each register consists of n-qubits and does as follows:*
   (a) *Measure registers X and Y in the computational basis and let (x,y) denote the result and let ρ_{XAB}^{y} denote the resulting state in registers XAB.*

(b) If $O_n(x) = y$, then compute $\mathcal{P}_n(\rho^y_{XAB} \otimes |0^n\rangle\langle 0^n|)$, where $\mathcal{P}_n$ is the quantum channel mapping $|z\rangle$ to $|z\rangle|P(z)\rangle$ for any $z \in \{0,1\}^{3n}$. Measure the last n-qubits in the computational basis and output the result.
(c) Otherwise, output $\perp$.

We first 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 given in Construction 6. 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. *Let $\lambda, n \in \mathbb{N}$ be security parameters, $a = a(n) \geq n$ and $b = b(n)$ be polynomials, $s = s(\lambda)$ be a function, and $\mu = \mu(\lambda) \leq \lambda^{-c}$ be a function. There does not exist a black-box construction of a (μ, s)-$\perp$-OWSG from a (a, b)-PRF^{qs} for large enough constant c.*

Proof. For simplicity, we only prove the case $a = 3n$ and $b = n$, but the proof easily generalizes to other polynomials by modifying the parameters of the oracles.

Assume, for the purpose of obtaining a contradiction, that G^F is a black-box construction of a t-query (μ, s)-$\perp$-OWSG, from any $(3n, n)$-$\mathsf{PRF}^{\mathrm{qs}}$ F, where $t = c'\lambda$, and c' is some sufficiently large constant. We first show that there exists a $\mathsf{PRF}^{\mathrm{qs}}$ relative to $\mathcal{T}$.

Claim 8. *There exists a (quantum-query-secure) $(3n, n)$-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 7.** *The algorithms of a PRF^{qs} with oracle access to $\mathcal{T}$ is as follows:*

- $\mathsf{QSamp}^{\mathcal{T}}(1^n)$:
  1. *Query $\sigma_n(1^n)$ and let $|x, y\rangle$ denote the result.*
  2. *Sample a random string $a \leftarrow \{0,1\}^n$.*
  3. *Output $k = (x, y, a)$.*
- $F_k^{\mathcal{T}}(b)$: *Interpret k as (x, y, a). Output $\mathcal{O}_n(x, y, a, b)$.*

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 $(x, y) \leftarrow \sigma_n(1^n)$, $\mathcal{O}_n(x, y, a, \cdot) = P_n(x, a, \cdot)$, where $P_n(x, a, \cdot)$ is a random function independent of σ. Lemma 2.2 from [25] states that a random oracle acts as a PRF i.e. for any QPT $\mathcal{A}$:

$$\mathop{\mathbb{E}}_{P_n \leftarrow \Pi_{3n,n}, \tilde{P}_n \leftarrow \Pi_{n,n}} \left[\left| \Pr_{(x,a) \leftarrow \{0,1\}^n \times \{0,1\}^n} [\mathcal{A}^{P_n, P_n(x,a,\cdot)}(1^n) = 1] - \Pr[\mathcal{A}^{P_n, \tilde{P}_n}(1^n) = 1] \right| \right] \leq \frac{1}{2^{n/4}}.$$

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:

$$\mathop{\mathbb{E}}_{P_n\leftarrow\Pi_{3n,n},\tilde{P}_n\leftarrow\Pi_{n,n}}\left[\left|\Pr_{(x,a)\leftarrow\{0,1\}^n\times\{0,1\}^n}[\mathcal{A}^{P_n,P_n(x,a,\cdot),\mathcal{C}}(1^n)=1]-\Pr[\mathcal{A}^{P_n,\tilde{P}_n,\mathcal{C}}(1^n)=1]\right|\right]\leq\frac{1}{2^{n/4}}.$$

By Markov inequality, we get that

$$\Pr_{P_n\leftarrow\Pi_{3n,n},\tilde{P}_n\leftarrow\Pi_{n,n}}\left[\left|\Pr_{(x,a)\leftarrow\{0,1\}^n\times\{0,1\}^n}[\mathcal{A}^{P_n,P_n(x,a,\cdot),\mathcal{C}}(1^n)=1]-\Pr[\mathcal{A}^{P_n,\tilde{P}_n,\mathcal{C}}(1^n)=1]\right|\geq 2^{-n/8}\right]\leq 2^{-n/8}$$

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

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

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. □

Construction 7 is a $\mathsf{PRF}^{\mathrm{qs}}$ relative to $\mathcal{T}$ with probability 1 over the oracle distribution. Furthermore, correctness is satisfied for all oracles $\mathcal{T}$. Therefore, $G^{\mathcal{T}}$ is a $\perp$-OWSG with probability 1 over the oracles $\mathcal{T}$ and satisfies correctness for any oracle $\mathcal{T}\in\mathbb{T}$.

Claim 9. *For any QPT adversary $\mathcal{A}$ and polynomial $t=t(\lambda)$, there exists a negligible function δ such that:*

$$\Pr_{\mathcal{T}\leftarrow\mathbb{T}}[\mathsf{Adtg}^{\perp\text{-}\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{T}},G^{\mathcal{T}}}(1^\lambda,1^t)\leq\delta(\lambda)]\geq\frac{1}{\lambda^2}.$$

Proof. Same as the proof of Claim 6. □

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.

Claim 10. *There exists a set $\mathbb{G}_\lambda^{\mathcal{T}_{\leq\log(2m)}}\subseteq\{0,1\}^\lambda$ such that:*

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

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

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 $\tilde{\mathcal{T}} \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]$.

We first show that at least $1-\sqrt{\mu}$ fraction of inputs are in this set i.e. $\Pr_{k\leftarrow\{0,1\}^\lambda}[k \in \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}] \geq 1-\sqrt{\mu}$. Otherwise, we would have that at least $\sqrt{\mu}$ fraction of inputs that 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

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

This gives a contradiction since $\mathop{\mathbb{E}}_{\mathcal{T}\leftarrow\mathbb{T}}[|\mathcal{G}_\lambda^{\mathcal{T}}|] \geq 1-\mu$. Hence, $\Pr_{k\leftarrow\{0,1\}^\lambda}[k \in \mathbb{G}_\lambda^{\mathcal{T}_{\leq \log(2m)}}] \geq 1-\sqrt{\mu}$.

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

$$\Pr[G_\lambda^{\tilde{\mathcal{T}}}(k) \neq \perp : \tilde{\mathcal{T}} \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]] \geq 1-2\sqrt{\mu}. \tag{8}$$

It is sufficient to show that

$$\Pr\left[\perp \neq \rho_1 \neq \rho_2 \neq \perp \;\middle|\; \begin{array}{c} \mathcal{T}', \mathcal{T}'' \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}] \\ \rho_1 \leftarrow G^{\mathcal{T}'}(k) \\ \rho_2 \leftarrow G^{\mathcal{T}''}(k) \end{array}\right] \leq \sqrt{\mu}. \tag{9}$$

If this holds, then combining this with Eq. (8), there exists a unique pure-state $|\psi_k^{\mathcal{T}_{\leq \log(2m)}}\rangle$ such that

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

which is the result we need to show. Assume that Eq. (9) does not hold. To show a contradiction, we commence with a hybrid argument.

- Hybrid H_0:
  1. Sample an oracle $\tilde{\mathcal{T}} \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]$ and let O' and P' denote the functions encoded in $\tilde{\mathcal{T}}$.
  2. Sample $k \leftarrow \{0,1\}^\lambda$.
  3. Compute $\rho_1 \leftarrow G^{\mathcal{T}'}(k)$.
  4. Compute $\rho_2 \leftarrow G^{\mathcal{T}'}(k)$.
  5. Output (ρ_1, ρ_2).
- Hybrid H_1:
  1. Sample oracle $\tilde{\mathcal{T}} \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]$.
  2. Sample $k \leftarrow \{0,1\}^\lambda$.
  3. Initiate empty memory $\mathbf{M}$.
  4. Compute $\rho_1 \leftarrow G^{\mathcal{T}'_{\mathbf{M}}}(k)$.
  5. Reset $\mathbf{M}$ to empty.
  6. Compute $\rho_2 \leftarrow G^{\mathcal{T}'_{\mathbf{M}}}(k)$.

7. Output (ρ_1, ρ_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 $(x, O'_n(x))$.
- $\mathcal{O}'_{\mathbf{M}}(\rho_{XYAB})$:
  1. If $|\rho_{XYAB}| = 4n$ and $n \leq \log(2m)$, then output $\mathcal{O}_n(\rho_{XYAB})$.
  2. Otherwise, measure registers X and Y in the computational basis and let (x, y) denote the result.
  3. If $x \in \mathbf{M}$ and $y = O'_n(x)$, then output $P'_n(\rho^y_{XAB})$.
  4. Otherwise, output $\perp$.

– Hybrid H_2:
  1. Sample oracles $\mathcal{T}', \mathcal{T}'' \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]$.
  2. Sample $k \leftarrow \{0,1\}^\lambda$.
  3. Initiate empty memory $\mathbf{M}$.
  4. Compute $\rho_1 \leftarrow G^{\mathcal{T}'^{\mathbf{M}}}(k)$.
  5. Reset $\mathbf{M}$ to empty.
  6. Compute $\rho_2 \leftarrow G^{\mathcal{T}''^{\mathbf{M}}}(k)$.
  7. Output (ρ_1, ρ_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 $\mathcal{T}', \mathcal{T}'' \leftarrow \mathbb{T}[\mathcal{T}_{\leq \log(2m)}]$.
  2. Sample $k \leftarrow \{0,1\}^\lambda$.
  3. Compute $\rho_1 \leftarrow G^{\mathcal{T}'}(k)$.
  4. Compute $\rho_2 \leftarrow G^{\mathcal{T}''}(k)$.
  5. Output (ρ_1, ρ_2).

Claim 11. *With probability at least $1 - \mu/8$, hybrids H_0 and H_1 are indistinguishable.*

Proof. The only way these hybrids may differ is if G submits a query ρ_{XYAB} of length $4n$ with $n > \log(2m)$ to $\mathcal{O}'_n$ such that the measurement result (x, y) satisfies $O'_n(x) = y$ and was not generated by the oracle σ'_n in an earlier query. Due to the randomness of the function O'_n and the run-time of G, this occurs with probability at most

$$1 - (1 - \frac{1}{2m})^{2r} \leq \frac{r}{m} \leq \frac{\mu}{8}.$$

□

Claim 12. *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 and O' are random functions. This scenario occurs with at least $1 - \frac{(2r)^2}{m} \geq 1 - \mu/8$ probability by the birthday problem. □

Claim 13. *With probability at least $1 - \mu/8$, hybrids H_2 and H_3 are indistinguishable.*

Proof. This follows in the same way as Claim 11. □

By the above three claims, 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 (ρ_1, ρ_2) are non-$\perp$ distinct states is at least $\sqrt{\mu}$. Therefore, the probability that the two states generated in hybrid H_0 also are non-$\perp$ distinct states 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 state or $\perp$ except with negligible probability. □

We are now ready to prove the main result (Theorem 9) using a hybrid argument.

Firstly, note that [27] shows that even computationally unbounded adversaries cannot distinguish a 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. Henceforth, for a string $w \in \{0,1\}^{6rn}$, we let $F_w : \mathbb{F}_{2^{3n}} \to \mathbb{F}_{2^{3n}}$ denote the polynomial of degree $2r-1$ on $3n$-bit inputs determined by w in the natural way. Fix a polynomial $t = t(\lambda)$.

- $\mathsf{Exp}_0^{\mathcal{A}}(\lambda)$: Sample oracle $\mathcal{T} \leftarrow \mathbb{T}$. Run the standard $\perp$-OWSG security experiment $\mathsf{Exp}^{\perp\text{-}\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{T}}, G^{\mathcal{T}}}(1^\lambda, 1^t)$ for G against an adversary $\mathcal{A}$ relative to $\mathcal{T}$. G and $\mathcal{A}$ are both given oracle access to $(\sigma, \mathcal{O}, \mathcal{C})$.
- $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$: The experiment is the same except G is given oracle access to stateful oracles $\mathcal{T}_{\mathbf{M}} := (\sigma_{\mathbf{M}}, \mathcal{O}_{\mathbf{M}})$ where the memory M is re-initialized to empty at the start of every evaluation of G. $\mathcal{A}$ is still given access to the same oracles $(\sigma, \mathcal{O}, \mathcal{C})$. The modified oracles for G are defined as follows:
  - $\sigma_{\mathbf{M}}(1^n)$:
    1. If $n \leq \log(2m)$, then output $\sigma_n(1^n)$.
    2. Otherwise, sample $z_1, z_2 \leftarrow \{0,1\}^n$. Return $z := (z_1, z_2)$.
    3. Store $(z_1, z_2, \perp)$ in memory $\mathbf{M}$.
  - $\mathcal{O}_{\mathbf{M}}(\rho_{XYAB})$:
    1. If $|\rho_{XYAB}| = 4n$ and $n \leq \log(2m)$, then output $\mathcal{O}_n(\rho_{XYAB})$.
    2. Otherwise, measure registers X and Y in the computational basis and let (x, y) denote the result.
    3. If $(x, y, w) \in \mathbf{M}$ for some $w \in \{0,1\}^{6rn}$, then output $|F_w(\rho^y_{XAB})\rangle$.

4. If $(x,\cdot,\cdot) \notin \mathbf{M}$, then sample $y' \leftarrow \{0,1\}^n$ and store $(x,y',\bot)$ in $\mathbf{M}$.
5. If $(x,y,\bot) \in \mathbf{M}$, then sample $w' \leftarrow \{0,1\}^{6rn}$. Switch the entry $(x,y,\bot)$ in $\mathbf{M}$ with (x,y,w') and output $|F_{w'}(\rho^y_{XAB})\rangle$.
6. Otherwise, output $\bot$.

– $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$: Same as $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$, except $\mathcal{A}$ is only given oracle access to $\mathcal{C}$.
– $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$: Same as $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$ except we replace G with the generator $\overline{G}$ that does not use any 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 $(\sigma_n^{k_2},\mathcal{O}_n^{k_2})_{n\le\log(2m)}$ in the same way as Construction 6 but with the randomness determined by k_2.
- Set $\sigma_n^{k_2}(x)=\bot$ and $\mathcal{O}_n^{k_2}(x)=\bot$ for all $x\in\{0,1\}^n$ and $n>\log(2m)$.
- Define $\mathcal{T}_{\mathbf{M}}^{k_2} := (\sigma_{\mathbf{M}}^{k_2},\mathcal{O}_{\mathbf{M}}^{k_2})$ in the same way as $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$.
- Output $G^{\mathcal{T}_{\mathrm{M}}^{k_2}}(k_1)$.

Claim 14. *For any QPT adversary $\mathcal{A}$*

$$d_{TD}(\mathsf{Exp}_0^{\mathcal{A}}(\lambda),\mathsf{Exp}_1^{\mathcal{A}}(\lambda)) \le 1/\lambda^4.$$

Proof. By Claim 10, there exists a set $\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}}$ such that:

1. $\Pr_{k\leftarrow\{0,1\}^\lambda}[k\in\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}}] \ge 1-\sqrt{\mu}$.
2. If $k\in\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}}$, then there exists a state $|\psi_k^{\mathcal{T}_{\le\log(2m)}}\rangle$ such that:

$$\Pr[G_\lambda^{\tilde{\mathcal{T}}}(k)=\left|\psi_k^{\mathcal{T}_{\le\log(2m)}}\right\rangle : \tilde{\mathcal{T}}\leftarrow\mathbb{T}[\mathcal{T}_{\le\log(2m)}]] \ge 1-3\sqrt{\mu}. \tag{10}$$

Let B denote the event that the key k and oracle $\mathcal{T}$ sampled in $\mathsf{Exp}_0^{\mathcal{A}}(\lambda)$ satisfy the following conditions: $k\in\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}}\cap\mathcal{G}^{\mathcal{T}}$ and $|\psi_k^{\mathcal{T}_{\le\log(2m)}}\rangle = |\psi_k^{\mathcal{T}}\rangle$, where $|\psi_k^{\mathcal{T}_{\le\log(2m)}}\rangle$ is the state that satisfies Eq. (10).

We now show that event B occurs with probability at least $1-6\sqrt{\mu}$. Note that $\Pr[k\in\mathcal{G}_\lambda^{\mathcal{T}} : k\leftarrow\{0,1\}^\lambda,\mathcal{T}\leftarrow\mathbb{T}] \ge 1-\mu$ and $\Pr[k\in\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}} : k\leftarrow\{0,1\}^\lambda,\mathcal{T}\leftarrow\mathbb{T}] \ge 1-\sqrt{\mu}$ so $\Pr[k\in\mathcal{G}_\lambda^{\mathcal{T}}\cap\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}} : k\leftarrow\{0,1\}^\lambda,\mathcal{T}\leftarrow\mathbb{T}] \ge 1-2\sqrt{\mu}$. It is sufficient to show that $|\psi_k^{\mathcal{T}_{\le\log(2m)}}\rangle = |\psi_k^{\mathcal{T}}\rangle$ with at least $(1-4\sqrt{\mu})$ probability if $k\leftarrow\mathcal{G}_\lambda^{\mathcal{T}}\cap\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}}$ and $\mathcal{T}\leftarrow\mathbb{T}$. If this does not hold, then by an averaging argument, we obtain

$$\Pr\left[G_\lambda^{\tilde{\mathcal{T}}}(k)=\left|\psi_k^{\mathcal{T}_{\le\log(2m)}}\right\rangle : \begin{matrix}\mathcal{T}\leftarrow\mathbb{T}\\ \tilde{\mathcal{T}}\leftarrow\mathbb{T}[\mathcal{T}_{\le\log(2m)}]\\ k\leftarrow\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}}\end{matrix}\right] =$$

$$\Pr\left[G_\lambda^{\mathcal{T}}(k)=\left|\psi_k^{\mathcal{T}_{\le\log(2m)}}\right\rangle : \begin{matrix}\mathcal{T}\leftarrow\mathbb{T}\\ k\leftarrow\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}}\end{matrix}\right] \le$$

$$(1-2\sqrt{\mu})\Pr\left[G_\lambda^{\mathcal{T}}(k)=\left|\psi_k^{\mathcal{T}_{\le\log(2m)}}\right\rangle : \begin{matrix}\mathcal{T}\leftarrow\mathbb{T}\\ k\leftarrow\mathbb{G}_\lambda^{\mathcal{T}_{\le\log(2m)}}\cap\mathcal{G}_\lambda^{\mathcal{T}}\end{matrix}\right] + 2\sqrt{\mu} \le$$

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

which contradicts Eq. (10), 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 $\mathsf{Exp}_0^{\mathcal{A}}(\lambda)$, the generator $G_\lambda^{\mathcal{T}}(k)$ returns $|\psi_k^{\mathcal{T}_{\leq \log(2m)}}\rangle^{\otimes t}$ with probability $(1-\mathsf{negl}(\lambda))^t$. While, in $\mathsf{Exp}_1^{\mathcal{A}}(\lambda)$, $G_\lambda^{\mathcal{T}_{\mathbf{M}}}(k)$ returns $|\psi_k^{\mathcal{T}_{\leq \log(2m)}}\rangle^{\otimes t}$ with probability $(1-3\sqrt{\mu})^t \geq 1-\frac{1}{4\lambda^4}$ as long as c is large enough.

Overall, if event B occurs, then the two experiments can only be distinguished with at most $\frac{1}{2\lambda^4}$ probability. Given that event B occurs with probability at least $1-6\sqrt{\mu}$, these two experiments can be distinguished with at most $6\sqrt{\mu}+\frac{1}{2\lambda^4} < \frac{1}{\lambda^4}$ probability as long as $c > 8$. □

Claim 15. *For any QPT adversary $\mathcal{A}$, there exists a QPT adversary $\mathcal{A}'$ such that*

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

Proof. This is clear because the only difference between theses experiments is that the adversary is no longer given access to $\mathcal{T}$. □

Claim 16. *For any QPT adversary $\mathcal{A}$*

$$d_{TD}(\mathit{Exp}_3^{\mathcal{A}}(\lambda), \mathit{Exp}_2^{\mathcal{A}}(\lambda)) = 0.$$

Proof. Notice that for any oracle $\mathcal{T}$, the modified oracle $\mathcal{T}_{\mathbf{M}}$ actually only depends on $\mathcal{T}_{\leq \log(2m)}$. Furthermore, it is clear that sampling $\mathcal{T}_{\leq \log(2m)}$ in $\mathsf{Exp}_2^{\mathcal{A}}(\lambda)$ is equivalent to sampling $\mathcal{T}_{\leq \log(2m)}^{k_2}$ in $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$. Therefore, for any $k \in \{0,1\}^\lambda$, $G^{\mathcal{T}_{\mathbf{M}}}(k)$ has the same distribution as $G^{k_2}_{\mathbf{M}}(k)$. □

By the above three claims and the triangle inequality, for any QPT adversary $\mathcal{A}$, there exists a QPT adversary $\mathcal{A}'$ such that

$$\left|\Pr[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1] - \Pr[\mathsf{Exp}_0^{\mathcal{A}'}(\lambda)=1]\right| \leq \frac{1}{\lambda^4} + \mathsf{negl}(\lambda). \tag{11}$$

By Claim 9, for any QPT adversary $\mathcal{A}$ and polynomial $t = t(\lambda)$, there exists a negligible function δ such that:

$$\Pr_{\mathcal{T}\leftarrow\mathbb{T}}[\mathsf{Adtg}^{\perp\text{-}\mathsf{OWSG}}_{\mathcal{A}^{\mathcal{T}},G^{\mathcal{T}}}(1^\lambda,1^t) \leq \delta(\lambda)] \geq \frac{1}{\lambda^2}.$$

Therefore, for any QPT adversary $\mathcal{A}$

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

By Eq. (11), for any QPT adversary $\mathcal{A}$ and large enough λ,

$$\Pr[\mathsf{Exp}_3^{\mathcal{A}}(\lambda)=1] \leq 1-\frac{1}{\lambda^3}.$$

Notice that $\overline{G}$ does not use any oracle access and satisfies the correctness condition of a $\mathsf{PD\text{-}OWSG}$ by Claim 10. Therefore, we can view $\mathsf{Exp}_3^{\mathcal{A}}(\lambda)$ as the security experiment $\mathsf{Exp}^{\mathsf{PD\text{-}OWSG}}_{\mathcal{A}^{\mathcal{C}},\overline{G}}(1^\lambda,1^t)$ of $\overline{G}$ against adversary $\mathcal{A}^{\mathcal{C}}$.

Therefore, we have that for any QPT adversary $\mathcal{A}$,

$$\Pr[\mathsf{Exp}^{\mathsf{PD\text{-}OWSG}}_{\mathcal{A}^{\mathcal{C}},\overline{G}}(1^\lambda,1^t)=1]\leq 1-\frac{1}{\lambda^3}. \tag{12}$$

Recall that a weak $\mathsf{PD\text{-}OWSG}$ can be converted into a OWPuzz (Lemmas 5 and 6) as long as c' is chosen large enough. On the other hand, there exists an attack against any OWPuzz using a PSPACE oracle given in [7]. In particular, there exists a QPT adversary $\overline{\mathcal{A}}$ and polynomial t such that

$$\Pr[\mathsf{Exp}^{\mathsf{PD\text{-}OWSG}}_{\overline{\mathcal{A}}^{\mathcal{C}},\overline{G}}(1^\lambda,1^t)=1]\geq 1-\mathsf{negl}(\lambda)$$

which contradicts Eq. (12).

So there does not exist a black-box construction of a t-copy (μ,s)-$\bot$-OWSG from a $\mathsf{PRF}^{\mathrm{qs}}$. □