

Quantum Cryptography and Hardness of Non-Collapsing Measurements

Tomoyuki Morimae¹, Yuki Shirakawa¹, and Takashi Yamakawa^{2,3,1}

¹Yukawa Institute for Theoretical Physics, Kyoto University, Kyoto, Japan

tomoyuki.morimae@yukawa.kyoto-u.ac.jp

yuki.shirakawa@yukawa.kyoto-u.ac.jp

²NTT Social Informatics Laboratories, Tokyo, Japan

takashi.yamakawa@ntt.com

³NTT Research Center for Theoretical Quantum Information, Atsugi, Japan

Abstract

One-way puzzles (OWPuzzs) introduced by Khurana and Tomer [STOC 2024] are a natural quantum analogue of one-way functions (OWFs), and one of the most fundamental primitives in “Microcrypt” where OWFs do not exist but quantum cryptography is possible. OWPuzzs are implied by almost all quantum cryptographic primitives, and imply several important applications such as non-interactive commitments and multi-party computations. A significant goal in the field of quantum cryptography is to base OWPuzzs on plausible assumptions that will not imply OWFs. In this paper, we base OWPuzzs on hardness of non-collapsing measurements. To that end, we introduce a new complexity class, **SampPDQP**, which is a sampling version of the decision class **PDQP** introduced in [Aaronson, Bouland, Fitzsimons, and Lee, ITCS 2016]. We show that if **SampPDQP** is hard on average for quantum polynomial time, then OWPuzzs exist. We also show that if **SampPDQP** $\not\subseteq$ **SampBQP**, then auxiliary-input OWPuzzs exist. **SampPDQP** is the class of sampling problems that can be solved by a classical polynomial-time deterministic algorithm that can make a single query to a non-collapsing measurement oracle, which is a “magical” oracle that can sample measurement results on quantum states without collapsing the states. Such non-collapsing measurements are highly unphysical operations that should be hard to realize in quantum polynomial-time, and therefore our assumptions on which OWPuzzs are based seem extremely plausible. Moreover, our assumptions do not seem to imply OWFs, because the possibility of inverting classical functions would not be helpful to realize quantum non-collapsing measurements. We also study upperbounds of the hardness of **SampPDQP**. We introduce a new primitive, *distributional collision-resistant puzzles* (*dCRPuzzs*), which are a natural quantum analogue of distributional collision-resistant hashing [Dubrov and Ishai, STOC 2006]. We show that dCRPuzzs imply average-case hardness of **SampPDQP** (and therefore OWPuzzs as well). We also show that two-message honest-statistically-hiding commitments with classical communication and one-shot message authentication codes (MACs), which are a privately-verifiable version of one-shot signatures [Amos, Georgiou, Kiayias, Zhandry, STOC 2020], imply dCRPuzzs.

Contents

1	Introduction	3
1.1	Our Results	3
1.2	Related Works	6
1.3	Technical Overview	7
2	Preliminaries	10
2.1	Basic Notations	10
2.2	One-Way Puzzles and Distributional One-Way Puzzles	10
3	SampPDQP	12
3.1	PDQP	12
3.2	SampPDQP	13
4	One-Way Puzzles from Average-Case Hardness of SampPDQP	14
5	Adaptive PDQP and Auxiliary-Input One-Way Puzzles	19
6	Distributional Collision-Resistant Puzzles	24
6.1	Definition of classical dCRH	24
6.2	Definition of dCRPuzzs	25
6.3	dCRPuzzs imply average-case hardness of SampPDQP	25
7	One-Shot Signatures and MACs	26
7.1	Definitions	26
7.2	One-shot MACs imply dCRPuzzs	28
8	Commitments	30
8.1	Definitions	30
8.2	Commitments imply dCRPuzzs	31

1 Introduction

It is widely accepted that the existence of one-way functions (OWFs) is the minimal assumption in classical cryptography [IL89]. On the other hand, in quantum cryptography, OWFs would not necessarily be the minimal assumption: there could exist a new world, “Microcrypt”, where quantum cryptography is possible without OWFs [Kre21, MY22, AQY22]. Many fundamental quantum cryptographic primitives and applications have been found in Microcrypt [JLS18, AGQY22, BCQ23, MY24, KT24, BGH⁺23, ALY24, MYY24, BBSS23, MX24, MH25, CGG24, BJ24] and separated from OWFs [Kre21, KQST23, LMW24, KQT25].

Among these many fundamental primitives, one-way puzzles (OWPuzzs) introduced by Khurana and Tomer [KT24] are a natural quantum analogue of OWFs. A OWPuzz is a pair (Samp, Ver) of a quantum polynomial-time (QPT) sampling algorithm Samp and a (not-necessarily-efficient) verification algorithm Ver. Samp outputs two bit strings, ans and puzz, that pass the verification $\text{Ver}(\text{ans}, \text{puzz})$ with high probability. The security of OWPuzzs requires that no QPT adversary $\mathcal{A}$ given puzz can output ans' that passes the verification $\text{Ver}(\text{puzz}, \text{ans}')$ with high probability. Almost all primitives in Microcrypt imply OWPuzzs and several important primitives and applications are implied by OWPuzzs including EFI pairs [BCQ23, KT24], non-interactive commitments [Yan22, MY22], and multi-party computations [MY22, AQY22, BCKM21, GLSV21].

OWPuzzs can be trivially constructed from (quantumly-secure) OWFs, but their construction based on other plausible assumptions, particularly those that do not imply OWFs, has not been extensively studied. (There are only three results. See Section 1.2.) One of the most significant goals in the field of quantum cryptography is to base OWPuzz on some plausible assumptions that will not imply OWFs.

1.1 Our Results

According to the laws of quantum physics, quantum states are generally collapsed by measurements. However, we could imagine some magical *non-collapsing measurements* that can sample measurement results without collapsing quantum states. The assumption that such highly-unphysical measurements are impossible in QPT seems extremely plausible.¹ The contribution of this paper is to base quantum cryptography on such a “physically reasonable” assumption, directly motivated by a fundamental law of quantum physics. Specifically, we show that OWPuzzs can be based on the computational hardness of simulating non-collapsing measurements.

Construction of OWPuzzs. To that end, we first introduce a new complexity class, **SampPDQP**, that characterizes a computational power of non-collapsing measurements. We then show the following result.

Theorem 1.1. *If **SampPDQP** is hard on average,² then OWPuzzs exist.*

SampPDQP is the sampling version of **PDQP**. Let us first explain **PDQP**. **PDQP** was introduced by Aaronson, Bouland, Fitzsimons, and Lee [ABFL16]. **PDQP** is the class of decision problems that can be solved with a classical deterministic polynomial-time algorithm that can make a single query to a magical

¹If a single copy of unknown quantum state is given, non-collapsing measurements on the state are statistically impossible (as long as we believe the standard quantum physics). On the other hand, as we will explain later, in our setup, a classical description of a quantum circuit that generates states is given, and therefore non-collapsing measurements are possible in unbounded time. Our assumptions are therefore computational ones.

²We say that a complexity class $\mathcal{C}$ of sampling problems is hard on average if there exist a sampling problem $\{\mathcal{D}_x\}_x \in \mathcal{C}$, a polynomial p , and a QPT samplable distribution $\mathcal{E}(1^\lambda) \rightarrow x \in \{0, 1\}^\lambda$ such that for any QPT algorithm $\mathcal{F}$ and for all sufficiently large $\lambda \in \mathbb{N}$, $\text{SD}(\{x, \mathcal{F}(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{D}_x\}_{x \leftarrow \mathcal{E}(1^\lambda)}) > \frac{1}{p(\lambda)}$. Here, SD is the statistical distance.

oracle $\mathcal{Q}$ that can perform non-collapsing measurements. More precisely, $\mathcal{Q}$ takes a classical description of a quantum circuit $(U_1, M_1, \dots, U_T, M_T)$ as input. Here, for each $i \in [T]$, U_i is an ℓ -qubit unitary and $M_i := \{P_j^i\}_j$ is a *collapsing* projection measurement on m_i qubits, where $0 \leq m_i \leq \ell$. (When $m_i = 0$, this means that M_i does not do any measurement.) The oracle $\mathcal{Q}$ first generates $U_1|0^\ell\rangle$ and performs the *collapsing* projection measurement M_1 on the state. Assume that the result j_1 is obtained. Then the post-measurement state is $|\psi_1\rangle := P_{j_1}^1 U_1|0^\ell\rangle / \sqrt{\|P_{j_1}^1 U_1|0^\ell\rangle\|^2}$. Then the oracle $\mathcal{Q}$ does a *non-collapsing* measurement on all qubits of $|\psi_1\rangle$ in the computational basis, and gets the measurement result $v_1 \in \{0, 1\}^\ell$. Because this is a non-collapsing measurement, this measurement does not collapse $|\psi_1\rangle$ to $|v_1\rangle$: even after obtaining v_1 , the state is still $|\psi_1\rangle$. The oracle $\mathcal{Q}$ then applies U_2 on $|\psi_1\rangle$, and performs the *collapsing* projection measurement M_2 on $U_2|\psi_1\rangle$. Assume that the result j_2 is obtained. Then the post-measurement state is $|\psi_2\rangle := P_{j_2}^2 U_2|\psi_1\rangle / \sqrt{\|P_{j_2}^2 U_2|\psi_1\rangle\|^2}$. The oracle $\mathcal{Q}$ again performs the non-collapsing computational-basis measurement on the all qubits of $|\psi_2\rangle$ to get the result $v_2 \in \{0, 1\}^\ell$. The oracle $\mathcal{Q}$ then applies U_3 on $|\psi_2\rangle$, measures it with M_3 , and performs the non-collapsing measurement on the post-measurement state of M_3 to get $v_3 \in \{0, 1\}^\ell$, and so on. In this way, the oracle obtains $(v_1, \dots, v_T)$. The oracle finally outputs $(v_1, \dots, v_T)$.

SampPDQP is the sampling version of **PDQP**. In other words, **SampPDQP** is the class of sampling problems³ that are solved by a classical deterministic polynomial-time algorithm that can make a single query to the non-collapsing measurement oracle $\mathcal{Q}$. We can show the following lemma:

Lemma 1.2. *If **PDQP** is hard on average,⁴ then **SampPDQP** is hard on average.*

We therefore obtain the following result as a corollary of Theorem 1.1.

Corollary 1.3. *If **PDQP** is hard on average, then **OWPuzzs** exist.*

Construction of auxiliary-input **OWPuzzs.** We have seen that average-case hardness of **SampPDQP** implies **OWPuzzs**. What happens for the worst-case hardness, **SampPDQP** $\not\subseteq$ **SampBQP**? We can actually show that such a worst-case hardness implies auxiliary-input **OWPuzzs**.⁵

Theorem 1.4. *If **SampPDQP** $\not\subseteq$ **SampBQP**, then auxiliary-input **OWPuzzs** exist.*

We can consider a generalization of **SampPDQP**, which we call **SampAdPDQP**. We show that the worst-case hardness of **SampAdPDQP** is equivalent to that of **SampPDQP**.

Theorem 1.5. ***SampAdPDQP** $\not\subseteq$ **SampBQP** if and only if **SampPDQP** $\not\subseteq$ **SampBQP**.*

SampAdPDQP is a generalization of **SampPDQP**. In **SampPDQP**, the base algorithm can query the non-collapsing measurement oracle $\mathcal{Q}$ only once, but in **SampAdPDQP**, it can query many times adaptively.⁶ By combining Theorems 1.4 and 1.5, we obtain the following corollary.

Corollary 1.6. *If **SampAdPDQP** $\not\subseteq$ **SampBQP**, then auxiliary-input **OWPuzzs** exist.*

³A sampling problem is a family $\{\mathcal{D}_x\}_x$ of distributions over bit strings. We say that a sampling problem $\{\mathcal{D}_x\}_x$ is solved if all $\mathcal{D}_x$ can be sampled.

⁴We say that a complexity class $\mathbf{C}$ of decision problems is hard on average if there exist a language $L \in \mathbf{C}$, a polynomial p , and a QPT samplable distribution $\mathcal{E}(1^\lambda) \rightarrow x \in \{0, 1\}^\lambda$ such that for any QPT algorithm $\mathcal{F}$ and for all sufficiently large $\lambda \in \mathbb{N}$, $\Pr_{x \leftarrow \mathcal{E}(1^\lambda)}[\mathcal{F}(x) \neq L(x)] > 1/p(\lambda)$.

⁵Roughly, an auxiliary-input **OWPuzz** is a pair (Samp, Ver) of algorithms such that for any QPT algorithm $\mathcal{A}$, there exists x such that under the sampling (puzz, ans) $\leftarrow$ Samp(x), $\mathcal{A}(x, \text{puzz})$ fails to find ans' that is accepted by Ver($x, \text{puzz}, \text{ans}'$).

⁶The non-collapsing measurement oracle is stateless, which means that it does not keep its internal quantum state between queries.

Relations to OWFs. Although there is no formal proof, our assumptions do not seem to imply OWFs, because the ability of inverting classical functions does not seem to be useful to realize quantum non-collapsing measurements. Moreover, the following argument also suggests that the average-case hardness of **SampPDQP** will not imply auxiliary-input OWFs (and therefore OWFs as well): [KQT25] left an open problem to separate quantum-evaluation collision-resistant hashing (CRH) from $\mathbf{P} = \mathbf{NP}$ and gave a concrete candidate construction for it relative to an oracle. If this open problem is resolved, average-case hardness of **SampPDQP** does not imply auxiliary-input OWFs, because (as we will see later) quantum-evaluation CRH implies average-case hardness of **SampPDQP**, and auxiliary-input OWFs imply $\mathbf{P} \neq \mathbf{NP}$.

Distributional collision-resistant puzzles. We also study upperbounds of the hardness of **SampPDQP**. We introduce a new primitive, *distributional collision-resistant puzzles* (dCRPuzzs). They are a natural quantum analogue of distributional collision-resistant hashing (dCRH) [DI06].⁷

A dCRPuzz is a set (Setup, Samp) of algorithms. Setup is a QPT algorithm that, on input the security parameter λ , outputs a public parameter pp . Samp is a QPT algorithm that, on input pp , outputs two classical bit strings, $puzz$ and ans . The security requirement is that for any QPT adversary $\mathcal{A}$, the statistical distance between two distributions, $(pp, \mathcal{A}(pp))_{pp \leftarrow \text{Setup}(1^\lambda)}$ and $(pp, \text{Col}(pp))_{pp \leftarrow \text{Setup}(1^\lambda)}$, is large. Here, $\text{Col}(pp) \rightarrow (puzz, ans, ans')$ is the following distribution: It first samples $(puzz, ans) \leftarrow \text{Samp}(pp)$, and then samples ans' with the conditional probability $\Pr[ans'|puzz] = \Pr[(ans', puzz) \leftarrow \text{Samp}(pp)] / \Pr[puzz \leftarrow \text{Samp}(pp)]$.

It is trivial that (quantumly-secure) dCRH (and therefore collision-resistant hashing (CRH) as well) imply dCRPuzzs. Moreover, because an average-case hardness of **SZK** for QPT implies quantumly-secure dCRH [KY18], the average-case hardness of **SZK** for QPT also implies dCRPuzzs.

We show that dCRPuzzs are an upperbound of the hardness of **SampPDQP**:

Theorem 1.7. *If dCRPuzzs exist, then SampPDQP is hard on average.*

Applications that imply dCRPuzzs. We show that several applications imply dCRPuzzs. We first show that one-shot message authentication codes (MACs) imply dCRPuzzs.

Theorem 1.8. *If one-shot MACs exist, then dCRPuzzs exist.*

A one-shot signature [AGKZ20] is a digital signature scheme with a quantum signing key. Signing a message with the key can be done only once. One-shot MACs [CKNY24] are a privately-verifiable version of one-shot signatures. One-shot MACs are also a relaxation of two-tier one-shot signatures [MPY23], where the verification is partially public. Because two-tier one-shot signatures can be constructed from the LWE assumption [MPY23], one-shot MACs can also be constructed from the LWE assumption [CKNY24].

We also show that two-message honest-statistically-hiding commitments with classical communication imply dCRPuzzs.⁸

Theorem 1.9. *If two-message honest-statistically-hiding commitments with classical communication exist, then dCRPuzzs exist.*

⁷We could also explore a quantum analogue of CRH. However, its definition is not so straightforward. For example, we could define a “collision-resistant puzzle” (Setup, Samp, Ver) as follows: Setup is a QPT algorithm that, on input the security parameter λ , outputs a public parameter pp . Samp is a QPT algorithm that, on input pp , outputs two classical bit strings, $puzz$ and ans . Ver is a (not-necessarily-efficient) algorithm that, on input pp , $puzz$, and ans , outputs $\top/\perp$. The “collision-resistance” requires that no QPT adversary $\mathcal{A}$ that receives pp as input can output $(puzz, ans, ans')$ such that $ans \neq ans'$ and both $(puzz, ans)$ and $(puzz, ans')$ are accepted by Ver with high probability. However, even if we require that the length of $puzz$ is shorter than that of ans , there is a trivial statistically-secure construction: Samp always outputs $puzz = 0$ and $ans = 00$. Ver accepts only $(puzz = 0, ans = 00)$.

⁸Here, honest statistical-hiding means that the adversary behaves honestly in the commit phase.

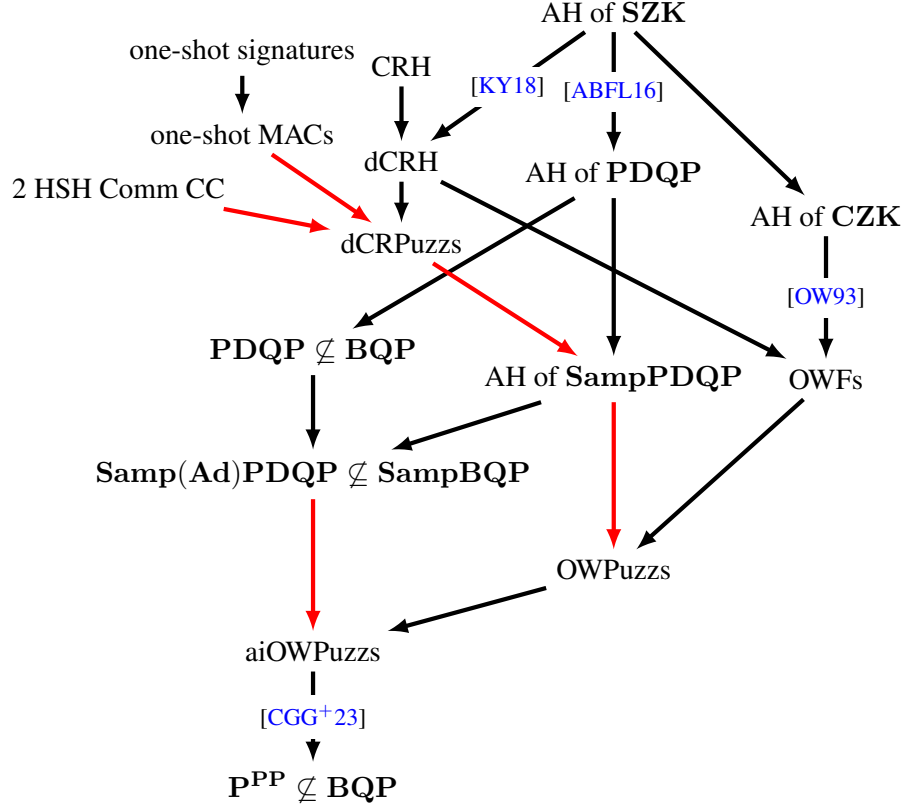


Figure 1: A summary of results. Black lines are known results or trivial implications. Red lines are new in our work. “AH” stands for the average-case hardness. “ai” stands for auxiliary-input. “2 HSH Comm CC” is two-message honest-statistically-hiding commitments with classical communication.

Summary of our results. Finally, all our results obtained in this paper and known results are summarized in Figure 1.

1.2 Related Works

Basing OWPuzzs on plausible assumptions. Recently, several results have been obtained that base OWPuzzs on some plausible assumptions that do not seem to imply OWFs. Khurana and Tomer [KT25] showed that OWPuzzs can be constructed from the assumption of $P^{PP} \not\subseteq BQP$ plus certain hardness assumptions that imply sampling-based quantum advantage. Hiroka and Morimae [HM25], and Cavalar, Goldin, Gray and Hall [CGGH25] independently showed that the existence of OWPuzzs is equivalent to certain average-case hardness of estimating Kolmogorov complexity. We do not know if their assumptions are related to our generic assumption of average-case hardness of $(\text{Samp})PDQP$. It would be interesting if there are some relations.

Relations to SZK and CZK. It is well known in classical cryptography that (classically-secure) OWFs exist if SZK is hard on average for probabilistic polynomial-time (PPT), and that (classically-secure) auxiliary-input OWFs exist if $SZK \not\subseteq BPP$ [Ost91]. Their proofs can be easily extended to show that

quantumly-secure OWFs exist if **SZK** is hard on average for QPT and that quantumly-secure auxiliary-input OWFs exist if $\mathbf{SZK} \not\subseteq \mathbf{BQP}$. Because quantumly-secure (auxiliary-input) OWFs imply (auxiliary-input) OWPuzzs, this means that

Corollary 1.10 ([Ost91]). *If **SZK** is hard on average for QPT, then OWPuzzs exist.*

Corollary 1.11 ([Ost91]). *If $\mathbf{SZK} \not\subseteq \mathbf{BQP}$, then auxiliary-input OWPuzzs exist.*

Because $\mathbf{SZK} \subseteq \mathbf{PDQP}$ [ABFL16], our results, Theorem 1.1 and Corollary 1.6, are improvements of Corollaries 1.10 and 1.11.

[OW93] improved the results of [Ost91] to **CZK**, and their proofs can be easily extended to the quantum case as well. We therefore obtain

Corollary 1.12 ([OW93]). *If **CZK** is hard on average for QPT, then OWPuzzs exist.*

Corollary 1.13 ([OW93]). *If $\mathbf{CZK} \not\subseteq \mathbf{BQP}$, then auxiliary-input OWPuzzs exist.*

To our knowledge, there is no known relation between **PDQP** and **CZK**, and therefore our results, Theorem 1.1 and Corollary 1.6, are incomparable to Corollaries 1.12 and 1.13.

Relations to PP. Khurana and Tomer [KT25] constructed OWPuzzs from the assumption of $\mathbf{P}^{\mathbf{PP}} \not\subseteq \mathbf{BQP}$ plus some assumptions on which sampling-based quantum advantage are based. They left the question of whether OWPuzzs can be based solely on $\mathbf{P}^{\mathbf{PP}} \not\subseteq \mathbf{BQP}$ or its average-case version. Because $\mathbf{PDQP} \subseteq \mathbf{P}^{\mathbf{PP}}$ [ABFL16], and $\mathbf{SampAdPDQP} \not\subseteq \mathbf{SampBQP}$ implies $\mathbf{P}^{\mathbf{PP}} \not\subseteq \mathbf{BQP}$, our results, Theorem 1.1 and Corollary 1.6, solve weaker versions of their open problem.

1.3 Technical Overview

In this subsection, we provide high-level overviews of our proofs.

OWPuzzs from the average-case hardness of $\mathbf{SampPDQP}$. We first explain our construction of OWPuzzs from average-case hardness of $\mathbf{SampPDQP}$. Our proof technique is inspired by the notion of universal extrapolation [IL90, Ost91] and its application in the quantum setting [KT25, HM25, CGGH25]. Because of the equivalence between OWPuzzs and distributional OWPuzzs (DistOWPuzzs) [CGG24], it suffices to construct DistOWPuzzs. Here a DistOWPuzz is a QPT sampling algorithm $\mathbf{Samp}(1^\lambda) \rightarrow (\text{puzz}, \text{ans})$ that satisfies the following: There exists a polynomial q such that for any QPT algorithm $\mathcal{A}$,

$$\text{SD}(\{\text{puzz}, \text{ans}\}_{(\text{puzz}, \text{ans}) \leftarrow \mathbf{Samp}(1^\lambda)}, \{\text{puzz}, \mathcal{A}(\text{ans})\}_{(\text{puzz}, \text{ans}) \leftarrow \mathbf{Samp}(1^\lambda)}) > \frac{1}{q(\lambda)} \quad (1)$$

for all sufficiently large $\lambda \in \mathbb{N}$.

Assume that $\mathbf{SampPDQP}$ is hard on average. This means that there exist a sampling problem $\{\mathcal{D}_x\}_x \in \mathbf{SampPDQP}$, a polynomial p and a QPT algorithm $\mathcal{E}(1^\lambda) \rightarrow x \in \{0, 1\}^\lambda$ such that for any QPT algorithm $\mathcal{F}$,

$$\text{SD}(\{x, \mathcal{F}(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{D}_x\}_{x \leftarrow \mathcal{E}(1^\lambda)}) > \frac{1}{p(\lambda)} \quad (2)$$

for all sufficiently large $\lambda \in \mathbb{N}$. From this $\mathcal{E}$, we construct a DistOWPuzzs $\mathbf{Samp}$ as follows.

1. Sample $x \leftarrow \mathcal{E}(1^\lambda)$.
2. Let $C_x := (U_1, M_1, \dots, U_T, M_T)$ be (a classical description of) a quantum circuit that is queried to $\mathcal{Q}$ corresponding to the instance x .
3. Choose $i \leftarrow [T]$.
4. Run $(U_1, M_1, \dots, U_i, M_i)$ to get the measurement results $(u_1, \dots, u_i)$, where u_i is the measurement result of the measurement M_i .
5. Measure all qubits of the resulting state to get the result $v_i = u_i \| w_i$.
6. Output $\text{puzz} := (x, i, u_1, \dots, u_i)$ and $\text{ans} := w_i$.

For the sake of contradiction, assume that Samp is not a DistOWPuzz . Then, for any polynomial q , there exists a QPT $\mathcal{A}$ such that

$$\text{SD}(\{x, i, u_1, \dots, u_i, w_i\}, \{x, i, u_1, \dots, u_i, \mathcal{A}(x, i, u_1, \dots, u_i)\}) \leq \frac{1}{q(\lambda)} \quad (3)$$

for infinitely many $\lambda \in \mathbb{N}$, where $(x, i, u_1, \dots, u_i, w_i) \leftarrow \text{Samp}(1^\lambda)$. Our goal is to construct a QPT algorithm that contradicts Equation (2). Define the QPT algorithm $\mathcal{B}$ that simulates the output distribution of $\mathcal{Q}$ as follows:

1. Take x and $C_x = (U_1, M_1, \dots, U_T, M_T)$ as input.
2. Run C_x and obtain $(u_1, \dots, u_T)$, where u_i is the outcome of the measurement M_i .
3. For all $i \in [T]$, run $w_i \leftarrow \mathcal{A}(x, i, u_1, \dots, u_i)$.
4. Output $(u_1 \| w_1, \dots, u_T \| w_T)$.

Roughly speaking, because of Equation (3), the distribution $w_i \leftarrow \mathcal{A}(x, i, u_1, \dots, u_i)$ in the third step of $\mathcal{B}$ is close to the distribution $\Pr[w_i \leftarrow \mathcal{Q}(C_x) | x \leftarrow \mathcal{E}(1^\lambda), i \leftarrow [T], (u_1, \dots, u_i) \leftarrow \mathcal{Q}(C_x)]$. Therefore, the output distribution of $\mathcal{B}$ is close to that of $\mathcal{Q}$. Hence a QPT algorithm that runs the base machine of **SampPDQP**, which is a polynomial-time deterministic machine, and runs the QPT algorithm $\mathcal{B}$ instead of the query to $\mathcal{Q}$ breaks Equation (2).

Auxiliary-input OWPuzzs from the worst-case hardness of SampAdPDQP. Our second result is the construction of auxiliary-input OWPuzzs from the worst-case assumption.⁹ The basic idea of the proof is similar to that of the first result, but there are two crucial issues, and we need more careful investigations. The first issue is that the assumption is now the worst-case hardness. The second issue is that now adaptive queries are allowed. Primitives constructed from worst-case assumptions often have to be auxiliary-input ones, and in fact, this is also the case here: what we construct is an auxiliary-input OWPuzzs. A slightly non-trivial and an interesting point is that the second issue is also resolved by considering only the auxiliary-input situation!

The first issue is easily resolved by giving the instance x to the OWPuzz as input. Let us explain more details about the second issue. When adaptive queries are allowed, the second query to $\mathcal{Q}$ can be a bit string that is not necessarily QPT generatable, because the second query can depend on the non-collapsing measurement results done in the first query to $\mathcal{Q}$. We resolve the issue by providing the answers of previous queries as auxiliary-input. Our construction of auxiliary-input DistOWPuzzs Samp is as follows:

⁹Here, we introduce an idea that directly proves Corollary 1.6. The proof of Theorem 1.4 follows as a special case of the same technique.

1. Take $x \in \{0, 1\}^*$, $k \in \mathbb{N}$, and a collection $(V_1, \dots, V_k)$ of k bit strings as input.
2. Let $C_{x,k+1} := (U_1, M_1, \dots, U_T, M_T)$ be (a classical description of) a quantum circuit that is the $(k+1)$ -th query to $\mathcal{Q}$ corresponding to the instance x . This is generated in polynomial-time by running the base machine of **SampAdPDQP** and using $(V_1, \dots, V_k)$ as answers of the previous k queries.
3. Choose $i \leftarrow [T]$.
4. Run $(U_1, M_1, \dots, U_i, M_i)$ to get the measurement results $(u_1, \dots, u_i)$, where u_i is the measurement result of the measurement M_i .
5. Measure all qubits of the resulting state to get the result $v_i = u_i || w_i$.
6. Output $\text{puzz} := (i, u_1, \dots, u_i)$ and $\text{ans} := w_i$.

Then, we can show the result in a similar way as the average case.

dCRPuzzs imply average-case hardness of SampPDQP. Remember that a dCRPuzz is a pair $(\text{Setup}, \text{Samp})$ of QPT algorithms such that for any QPT adversary $\mathcal{A}$, the statistical distance between two distributions, $(\text{pp}, \mathcal{A}(\text{pp}))_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}$ and $(\text{pp}, \text{Col}(\text{pp}))_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}$, is large. Here, $\text{Col}(\text{pp}) \rightarrow (\text{puzz}, \text{ans}, \text{ans}')$ is the following distribution: It first samples $(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(\text{pp})$, and then samples ans' with the conditional probability $\Pr[\text{ans}' | \text{puzz}] = \Pr[(\text{ans}', \text{puzz}) \leftarrow \text{Samp}(\text{pp})] / \Pr[\text{puzz} \leftarrow \text{Samp}(\text{pp})]$. Without loss of generality, we can assume that **Samp** runs as follows:

1. Apply a unitary V_{pp} on $|0\dots 0\rangle$ to generate $V_{\text{pp}}|0\dots 0\rangle = \sum_{\text{puzz}, \text{ans}} c_{\text{puzz}, \text{ans}} |\text{puzz}\rangle |\text{ans}\rangle$.
2. Measure the first register to get **puzz**.
3. Measure the second register to get **ans**.
4. Output $(\text{puzz}, \text{ans})$.

It is easy to see that the dCRPuzz is broken by querying the following $C = (U_1, M_1, U_2, M_2)$ to $\mathcal{Q}$:

1. $U_1 = V_{\text{pp}}$
2. M_1 is the measurement on the first register.
3. $U_2 = I$.
4. M_2 does not do any measurement.

One-shot MACs imply dCRPuzzs. Let m_0 and m_1 be any two different classical messages. Let vk be a verification key and $|\text{sigk}\rangle$ be a quantum signing key. If we apply the signing algorithm **Sign** coherently on $(|m_0\rangle + |m_1\rangle)|\text{sigk}\rangle|0\dots 0\rangle$ and measure the last register, we get (m_0, σ_0) or (m_1, σ_1) , where, for each $b \in \{0, 1\}$, σ_b is a valid signature for m_b . If we consider vk as **puzz** and (m_b, σ_b) as **ans**, non-existence of dCRPuzzs means that we can sample (m_b, σ_b) twice with the same vk . Then, with probability at least $1/2$, we get both (m_0, σ_0) and (m_1, σ_1) . Hence the one-shot MAC is broken.

Commitments imply dCRPuzzs. We can also show that two-message honest-statistically-hiding bit commitments with classical communication imply dCRPuzzs. In the two-message commitment, the sender receives a message r from the receiver, and then returns com , which is the commitment. Assume that sender commits both 0 and 1 in superposition. This means that the sender applies the commitment algorithm coherently on $(|0\rangle + |1\rangle)|0\dots 0\rangle$ and measures the second register to get the commitment com . Because of the statistical hiding, the state after the measurement is close to $|0\rangle|\text{decom}_0\rangle|\text{junk}_0\rangle + |1\rangle|\text{decom}_1\rangle|\text{junk}_1\rangle$, where decom_b is the decommitment for bit b . Consider com as puzz and (b, decom_b) as ans . Then if dCRPuzzs do not exist, we can sample both $(0, \text{decom}_0)$ and $(1, \text{decom}_1)$ with the same com , which breaks the binding.

2 Preliminaries

2.1 Basic Notations

We use standard notations of quantum computing and cryptography. For a bit string x , $|x|$ is its length. For bit strings x and y , $x||y$ is their concatenation. $\mathbb{N}$ is the set of natural numbers. We use λ as the security parameter. $[n]$ means the set $\{1, 2, \dots, n\}$. For a finite set S , $x \leftarrow S$ means that an element x is sampled uniformly at random from the set S . negl is a negligible function, and poly is a polynomial. All polynomials appear in this paper are positive, but for simplicity we do not explicitly mention it. PPT stands for (classical) probabilistic polynomial-time and QPT stands for quantum polynomial-time. For an algorithm $\mathcal{A}$, $y \leftarrow \mathcal{A}(x)$ means that the algorithm $\mathcal{A}$ outputs y on input x . If $\mathcal{A}$ is a classical probabilistic or quantum algorithm that takes x as input and outputs bit strings, we often mean $\mathcal{A}(x)$ by the output probability distribution of $\mathcal{A}$ on input x . For two probability distributions $P := \{p_i\}_i$ and $Q := \{q_i\}_i$, $\text{SD}(Q, P) := \frac{1}{2} \sum_i |p_i - q_i|$ is their statistical distance.

2.2 One-Way Puzzles and Distributional One-Way Puzzles

We first review the definition of one-way puzzles (OWPuzzs).

Definition 2.1 (OWPuzzs [KT24]). A one-way puzzle (OWPuzz) is a pair $(\text{Samp}, \text{Ver})$ of algorithms such that

- $\text{Samp}(1^\lambda) \rightarrow (\text{puzz}, \text{ans})$: It is a QPT algorithm that, on input the security parameter λ , outputs a pair $(\text{puzz}, \text{ans})$ of classical strings.
- $\text{Ver}(\text{puzz}, \text{ans}') \rightarrow \top/\perp$: It is an unbounded algorithm that, on input $(\text{puzz}, \text{ans}')$, outputs either $\top/\perp$.

They satisfy the following properties.

- **Correctness:**

$$\Pr[\top \leftarrow \text{Ver}(\text{puzz}, \text{ans}) : (\text{puzz}, \text{ans}) \leftarrow \text{Samp}(1^\lambda)] \geq 1 - \text{negl}(\lambda). \quad (4)$$

- **Security:** For any QPT adversary $\mathcal{A}$,

$$\Pr[\top \leftarrow \text{Ver}(\text{puzz}, \mathcal{A}(1^\lambda, \text{puzz})) : (\text{puzz}, \text{ans}) \leftarrow \text{Samp}(1^\lambda)] \leq \text{negl}(\lambda). \quad (5)$$

We also review the definition of distributional one-way puzzles (DistOWPuzzs).

Definition 2.2 (DistOWPuzzs [CGG24]). A uniform QPT algorithm Samp that takes the security parameter 1^λ as input and outputs a pair $(\text{puzz}, \text{ans})$ of bit strings is called an α -distributional one-way puzzle (α -DistOWPuzz) if there exists a function $\alpha : \mathbb{N} \rightarrow [0, 1]$ such that for any QPT adversary $\mathcal{A}$, and for all sufficiently large $\lambda \in \mathbb{N}$,

$$\text{SD} \left(\{ \text{puzz}, \text{ans} \}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(1^\lambda)}, \{ \text{puzz}, \mathcal{A}(1^\lambda, \text{puzz}) \}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(1^\lambda)} \right) \geq \alpha(\lambda). \quad (6)$$

If Samp is a λ^{-c} -DistOWPuzz for some constant $c > 0$, we simply say Samp is a DistOWPuzz.

Clearly, if $(\text{Samp}, \text{Ver})$ is a OWPuzz, then Samp is a DistOWPuzz. Chung, Goldin, and Gray [CGG24] showed that DistOWPuzzs imply OWPuzzs. Combining them, the following equivalence is known.

Lemma 2.3 ([CGG24]). OWPuzzs exist if and only if DistOWPuzzs exist.

We define auxiliary-input variants of OWPuzzs and DistOWPuzzs.

Definition 2.4 (Auxiliary-Input OWPuzzs). An auxiliary-input one-way puzzle (auxiliary-input OWPuzz) is a pair $(\text{Samp}, \text{Ver})$ of algorithms such that

- $\text{Samp}(x) \rightarrow (\text{puzz}, \text{ans})$: It is a QPT algorithm that, on input a bit string x , outputs a pair $(\text{puzz}, \text{ans})$ of classical bit strings.
- $\text{Ver}(x, \text{puzz}, \text{ans}') \rightarrow \top / \perp$: It is an unbounded algorithm that, on input $(x, \text{puzz}, \text{ans}')$, outputs either $\top / \perp$.

They satisfy the following properties.

- **Correctness:**

$$\Pr[\top \leftarrow \text{Ver}(x, \text{puzz}, \text{ans}) : (\text{puzz}, \text{ans}) \leftarrow \text{Samp}(x)] \geq 1 - \text{negl}(|x|). \quad (7)$$

- **Security:** For any QPT adversary $\mathcal{A}$, there exists an infinite subset $I \subseteq \{0, 1\}^*$ such that for all $x \in I$,

$$\Pr[\top \leftarrow \text{Ver}(x, \text{puzz}, \mathcal{A}(x, \text{puzz})) : (\text{puzz}, \text{ans}) \leftarrow \text{Samp}(x)] \leq \text{negl}(|x|). \quad (8)$$

Definition 2.5 (Auxiliary-Input DistOWPuzzs). A uniform QPT algorithm Samp that takes an advice bit string $x \in \{0, 1\}^*$ as input and outputs a pair $(\text{puzz}, \text{ans})$ of bit strings is called an auxiliary-input α -distributional one-way puzzle (auxiliary-input α -DistOWPuzz) if there exists a function $\alpha : \mathbb{N} \rightarrow [0, 1]$ such that for any QPT adversary $\mathcal{A}$, there exists an infinite subset $I \subseteq \{0, 1\}^*$ such that for all $x \in I$,

$$\text{SD} \left(\{ \text{puzz}, \text{ans} \}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(x)}, \{ \text{puzz}, \mathcal{A}(x, \text{puzz}) \}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(x)} \right) \geq \alpha(|x|). \quad (9)$$

If Samp is an auxiliary-input λ^{-c} -DistOWPuzzs for some constant $c > 0$, we simply say that Samp is an auxiliary-input DistOWPuzzs.

The auxiliary-input version of Lemma 2.3 can be obtained by slightly modifying the proof of Lemma 2.3.

Lemma 2.6 ([CGG24]). Auxiliary-input OWPuzzs exist if and only if auxiliary-input DistOWPuzzs exist.

3 SampPDQP

In this section, we introduce a sampling version of **PDQP**, which we call **SampPDQP**.

3.1 PDQP

Before introducing **SampPDQP**, we review the definition of the decision class **PDQP** introduced in [ABFL16]. The complexity class **PDQP** is a class of decision problems that can be solved with a polynomial-time classical deterministic algorithm that has a single query access to the non-collapsing measurement oracle. We first define the non-collapsing measurement oracle.

Definition 3.1 (Non-Collapsing Measurement Oracle [ABFL16]). *A non-collapsing measurement oracle $\mathcal{Q}$ is an oracle that behaves as follows:*

1. *Take (a classical description of) a quantum circuit $C = (U_1, M_1, \dots, U_T, M_T)$ and an integer $\ell > 0$ as input. Here each U_i is a unitary operator on ℓ qubits and each M_i is a computational-basis projective measurement on m_i qubits such that $0 \leq m_i \leq \ell$. (When $m_i = 0$, this means that no measurement is done.)*
2. *Let $|\psi_0\rangle := |0^\ell\rangle$. Run C on input $|\psi_0\rangle$, and obtain $(u_1, \dots, u_T)$, where $u_t \in \{0, 1\}^{m_t}$ is the outcome of the measurement M_t for each $t \in [T]$. Let $\tau_t := (u_1, \dots, u_t)$. For each $t \in [T]$, let $|\psi_t^{\tau_t}\rangle$ be the (normalized) post-measurement state immediately after the measurement M_t , i.e.,*

$$|\psi_t^{\tau_t}\rangle := \frac{(|u_t\rangle\langle u_t| \otimes I)U_t|\psi_{t-1}^{\tau_{t-1}}\rangle}{\sqrt{\langle\psi_{t-1}^{\tau_{t-1}}|U_t^\dagger(|u_t\rangle\langle u_t| \otimes I)U_t|\psi_{t-1}^{\tau_{t-1}}\rangle}}. \quad (10)$$

3. *For each $t \in [T]$, sample $v_t \in \{0, 1\}^\ell$ with probability $|\langle v_t | \psi_t^{\tau_t} \rangle|^2$.*
4. *Output $(v_1, \dots, v_T)$.*

Remark 3.2. Note that each non-collapsing measurement is done on *all* qubits including those that have been measured by the previous collapsing measurement. Therefore, the measurement result v_i of the i th non-collapsing measurement is written as $v_i = u_i \| w_i$ with a bit string $w_i \in \{0, 1\}^{\ell - m_i}$, where u_i is the measurement result of the collapsing measurement M_i . For example, after the measurement of M_i , the entire state becomes $|u_i\rangle \otimes |\phi_i\rangle$ with a certain $(\ell - m_i)$ -qubit state $|\phi_i\rangle$, where u_i is the measurement result of M_i . Then the non-collapsing measurement measures all qubits. The measurement result on the first register is always u_i , and therefore the measurement result v_i of the non-collapsing measurement is always written as $v_i = u_i \| w_i$, where $w_i \in \{0, 1\}^{\ell - m_i}$ is the measurement result of the non-collapsing measurement on $|\phi_i\rangle$.

With the non-collapsing measurement oracle, the class **PDQP** is defined as follows.

Definition 3.3 (PDQP [ABFL16]). *A language L is in **PDQP** if there exists a polynomial-time classical deterministic Turing machine R with a single query to a non-collapsing measurement oracle $\mathcal{Q}$ such that*

- *For all $x \in L$, $\Pr[1 \leftarrow R^{\mathcal{Q}}(x)] \geq \alpha(|x|)$,*
- *For all $x \notin L$, $\Pr[1 \leftarrow R^{\mathcal{Q}}(x)] \leq \beta(|x|)$,*

where α, β are functions such that $\alpha(|x|) - \beta(|x|) \geq 1/\text{poly}(|x|)$.

Remark 3.4. Note that the error bound (α, β) can be amplified to $(1 - \text{negl}, \text{negl})$ by the repetition [ABFL16].

We also use the notion of average-case hardness.

Definition 3.5 (Average-Case Hardness of PDQP). We say that **PDQP** is hard on average if the following is satisfied: there exist a language $L \in \mathbf{PDQP}$, a polynomial p , and a QPT algorithm $\mathcal{E}(1^\lambda) \rightarrow \{0, 1\}^\lambda$ such that for any QPT algorithm $\mathcal{F}$ and for all sufficiently large $\lambda \in \mathbb{N}$,

$$\Pr_{x \leftarrow \mathcal{E}(1^\lambda)} [\mathcal{F}(x) \neq L(x)] \geq \frac{1}{p(\lambda)}, \quad (11)$$

where

$$L(x) := \begin{cases} 1 & x \in L \\ 0 & x \notin L. \end{cases} \quad (12)$$

3.2 SampPDQP

Next we define **SampPDQP**. **SampPDQP** is the class of sampling problems that are solved with a polynomial-time classical deterministic algorithm that can make a single query to the non-collapsing measurement oracle. Sampling problems are defined as follows.

Definition 3.6 (Sampling Problems [Aar14, ABK24]). A (polynomially-bounded) sampling problem is a collection $\{D_x\}_{x \in \{0,1\}^*}$ of probability distributions, where D_x is a distribution over $\{0, 1\}^{p(|x|)}$, for some fixed polynomial p .

The sampling complexity class, **SampBQP**, is defined as follows.

Definition 3.7 (SampBQP [Aar14, ABK24]). **SampBQP** is the class of (polynomially-bounded) sampling problems $\{D_x\}_{x \in \{0,1\}^*}$ for which there exists a QPT algorithm $\mathcal{B}$ such that for all x and all $\epsilon > 0$, $\text{SD}(\mathcal{B}(x, 1^{\lceil 1/\epsilon \rceil}), D_x) \leq \epsilon$, where $\mathcal{B}(x, 1^{\lceil 1/\epsilon \rceil})$ is the output probability distribution of $\mathcal{B}$ on input $(x, 1^{\lceil 1/\epsilon \rceil})$.

We define **SampPDQP** as follows.

Definition 3.8 (SampPDQP). **SampPDQP** is the class of (polynomially-bounded) sampling problems $\{D_x\}_{x \in \{0,1\}^*}$ for which there exists a classical deterministic polynomial-time algorithm $\mathcal{B}$ that makes a single query to the non-collapsing measurement oracle $\mathcal{Q}$ such that for all x and all $\epsilon > 0$, $\text{SD}(\mathcal{B}(x, 1^{\lceil 1/\epsilon \rceil}), D_x) \leq \epsilon$, where $\mathcal{B}(x, 1^{\lceil 1/\epsilon \rceil})$ is the output probability distribution of $\mathcal{B}$ on input $(x, 1^{\lceil 1/\epsilon \rceil})$.

We also use the notion of average-case hardness.

Definition 3.9 (Average-case Hardness of SampPDQP). We say that **SampPDQP** is hard on average if the following is satisfied: there exist a sampling problem $\{D_x\}_x \in \mathbf{SampPDQP}$, a polynomial p , and a QPT algorithm $\mathcal{E}(1^\lambda) \rightarrow \{0, 1\}^\lambda$ such that for any QPT algorithm $\mathcal{F}$ and for all sufficiently large λ ,

$$\text{SD}(\{x, \mathcal{F}(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, D_x\}_{x \leftarrow \mathcal{E}(1^\lambda)}) > \frac{1}{p(\lambda)}. \quad (13)$$

We show the following lemma.

Lemma 3.10. If **PDQP** is hard on average, then **SampPDQP** is hard on average.

Proof of Lemma 3.10. Assume that **PDQP** is hard on average. Then there exists a language $L \in \mathbf{PDQP}$, a polynomial p , and a QPT algorithm $\mathcal{E}(1^\lambda) \rightarrow \{0, 1\}^\lambda$ such that for any QPT algorithm $\mathcal{F}$ and for all sufficiently large $\lambda \in \mathbb{N}$,

$$\Pr_{x \leftarrow \mathcal{E}(1^\lambda)} [\mathcal{F}(x) \neq L(x)] \geq \frac{1}{p(\lambda)}. \quad (14)$$

Because $L \in \mathbf{PDQP}$, there exists a classical deterministic polynomial-time Turing machine $\mathcal{R}$ and a non-collapsing measurement oracle $\mathcal{Q}$ such that for all $x \in \{0, 1\}^*$,

$$\Pr[\mathcal{R}^{\mathcal{Q}}(x) \neq L(x)] \leq \text{negl}(|x|). \quad (15)$$

Consider a sampling problem $\{\mathcal{R}^{\mathcal{Q}}(x)\}_{x \in \{0, 1\}^*}$. Clearly, $\{\mathcal{R}^{\mathcal{Q}}(x)\}_{x \in \{0, 1\}^*} \in \mathbf{SampPDQP}$. For the sake of contradiction, assume that **SampPDQP** is not hard on average. Then, there exists a QPT algorithm $\mathcal{F}^*$ such that for infinitely many $\lambda \in \mathbb{N}$,

$$\text{SD}(\{x, \mathcal{F}^*(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{R}^{\mathcal{Q}}(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}) \leq \frac{1}{2p(\lambda)}. \quad (16)$$

Our goal is to show that $\mathcal{F}^*$ breaks Equation (14). By Equations (15) and (16),

$$\Pr_{x \leftarrow \mathcal{E}(1^\lambda)} [\mathcal{F}^*(x) \neq L(x)] \leq \Pr_{x \leftarrow \mathcal{E}(1^\lambda)} [\mathcal{R}^{\mathcal{Q}}(x) \neq L(x)] + \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} [\text{SD}(\mathcal{F}^*(x), \mathcal{R}^{\mathcal{Q}}(x))] \quad (17)$$

$$\leq \text{negl}(\lambda) + \text{SD}(\{x, \mathcal{F}^*(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{R}^{\mathcal{Q}}(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}) \quad (18)$$

$$\leq \text{negl}(\lambda) + \frac{1}{2p(\lambda)} \quad (19)$$

$$\leq \frac{1}{p(\lambda)}, \quad (20)$$

holds for infinitely many $\lambda \in \mathbb{N}$. This contradicts Equation (14). □

4 One-Way Puzzles from Average-Case Hardness of SampPDQP

In this section, we construct OWPuzzs from the average-case hardness of **SampPDQP**.

Theorem 4.1. *If **SampPDQP** is hard on average, then OWPuzzs exist.*

Proof of Theorem 4.1. Because of the equivalence of OWPuzzs and DistOWPuzzs (Lemma 2.3), it suffices to construct DistOWPuzzs. Assume that **SampPDQP** is hard on average. Then there exist a sampling problem $S = \{\mathcal{D}_x\}_{x \in \{0, 1\}^*} \in \mathbf{SampPDQP}$, a polynomial p , and a QPT algorithm $\mathcal{E}(1^\lambda) \rightarrow \{0, 1\}^\lambda$ such that for any QPT algorithm $\mathcal{F}$ and for all sufficiently large $\lambda \in \mathbb{N}$,

$$\text{SD}(\{x, \mathcal{F}(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{D}_x\}_{x \leftarrow \mathcal{E}(1^\lambda)}) > \frac{1}{p(\lambda)}. \quad (21)$$

By the definition of **SampPDQP**, there exist a classical deterministic polynomial-time Turing machine $\mathcal{R}$ and a non-collapsing measurement oracle $\mathcal{Q}$ such that for all $x \in \{0, 1\}^*$ and for all $\epsilon > 0$,

$$\text{SD}(\mathcal{R}^{\mathcal{Q}}(x, 1^{\lceil 1/\epsilon \rceil}), \mathcal{D}_x) \leq \epsilon. \quad (22)$$

In the following, we fix $\epsilon := 2p(|x|)$. For each $x \in \{0, 1\}^*$, let $C_x := (U_1, M_1, \dots, U_{T_x}, M_{T_x})$ be the quantum circuit that $\mathcal{R}(x, 1^{\lfloor 1/\epsilon \rfloor})$ queries to $\mathcal{Q}$. Let C_x act on ℓ qubits. Here, each U_t be a unitary operator and each M_t be a computational basis projective measurement on m_i qubits such that $0 \leq m_i \leq \ell$.¹⁰ Let

$$T'(\lambda) := \max_{x \in \{0,1\}^\lambda} \{T_x\}. \quad (23)$$

Note that T' is a polynomial of $|x|$ because $\mathcal{R}$ is a polynomial-time machine.

By using $\mathcal{R}$ and $\mathcal{E}$, we construct a $\frac{1}{2pT'}$ -DistOWPuzz Samp as follows:

1. Take 1^λ as input.
2. Sample $x \leftarrow \mathcal{E}(1^\lambda)$.
3. Let $C_x := (U_1, M_1, \dots, U_T, M_T)$ be (a classical description of) a quantum circuit that is queried to $\mathcal{Q}$ corresponding to the instance x .
4. Sample $t \leftarrow [T_x]$.
5. Run $C_x = (U_1, M_1, \dots, U_t, M_t)$ on $|0^\ell\rangle$. Obtain $\tau_t := (u_1, \dots, u_t)$ and the resulting state $|\psi_t^{\tau_t}\rangle$, where $u_i \in \{0, 1\}^{m_i}$ is the measurement result of M_i for each $i \in [t]$.
6. Measure all qubits of $|\psi_t^{\tau_t}\rangle$ in the computational basis to obtain $v_t \in \{0, 1\}^\ell$, where $v_t := u_t \| w_t$ for some $w_t \in \{0, 1\}^{\ell-m_t}$.
7. Let $\text{puzz} := (x, t, \tau_t)$ and $\text{ans} := w_t$. Output $(\text{puzz}, \text{ans})$.

For the sake of contradiction, we assume that Samp is not a $\frac{1}{2pT'}$ -DistOWPuzz. Then by the definition of DistOWPuzzs, there exist a QPT algorithm $\mathcal{A}$ such that for infinitely many $\lambda \in \mathbb{N}$,

$$\text{SD} \left(\{\text{puzz}, \text{ans}\}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(1^\lambda)}, \{\text{puzz}, \mathcal{A}(1^\lambda, \text{puzz})\}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(1^\lambda)} \right) < \frac{1}{2p(\lambda)T'(\lambda)}. \quad (24)$$

Let $\Lambda \subseteq \mathbb{N}$ be the set of such λ . Note that

$$\sum_{x \in \{0,1\}^\lambda} \Pr[(x, t, \tau_t, w_t) \leftarrow \text{Samp}(1^\lambda)] = \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} \left[\frac{1}{T_x} \Pr[(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)] \right], \quad (25)$$

where $\mathcal{Q}^t$ is the following algorithm:

1. Take (a classical description of) a quantum circuit $C_x := (U_1, M_1, \dots, U_{T_x}, M_{T_x})$ acting on ℓ qubits as input. For each $i \in [T_x]$, U_i is a unitary operator and M_i is a computational basis projective measurement on m_i qubits such that $0 \leq m_i \leq \ell$.
2. Sample $(v_1, \dots, v_{T_x}) \leftarrow \mathcal{Q}(C_x)$, where $v_i = u_i \| w_i$ and u_i is a measurement result of M_i for each $i \in [T_x]$.
3. Let $\tau_t = (u_1, \dots, u_t)$. Output (τ_t, w_t) .

¹⁰Note that all of U_t , M_t , m_t , and ℓ depend on x , but for the notational simplicity, we omit their dependence on x .

By Equations (23) and (24), for all $\lambda \in \Lambda$,

$$\frac{1}{2p(\lambda)T'(\lambda)} > \text{SD} \left(\{\text{puzz}, \text{ans}\}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(1^\lambda)}, \{\text{puzz}, \mathcal{A}(1^\lambda, \text{puzz})\}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(1^\lambda)} \right) \quad (26)$$

$$= \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} \left[\frac{1}{T_x} \sum_{t \in [T_x]} \text{SD} \left(\{\tau_t, w_t\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)}, \{\tau_t, \mathcal{A}(1^\lambda, x, t, \tau_t)\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)} \right) \right] \quad (27)$$

$$\geq \frac{1}{T'(\lambda)} \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} \left[\sum_{t \in [T_x]} \text{SD} \left(\{\tau_t, w_t\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)}, \{\tau_t, \mathcal{A}(1^\lambda, x, t, \tau_t)\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)} \right) \right]. \quad (28)$$

Thus for all $\lambda \in \Lambda$,

$$\mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} \left[\sum_{t \in [T_x]} \text{SD} \left(\{\tau_t, w_t\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)}, \{\tau_t, \mathcal{A}(1^\lambda, x, t, \tau_t)\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)} \right) \right] < \frac{1}{2p(\lambda)}. \quad (29)$$

Our goal is to construct a QPT algorithm $\mathcal{F}$ that breaks Equation (21). We define $\mathcal{F}$ as follows:

1. Take $x \in \{0, 1\}^\lambda$ as input.
2. Run $\mathcal{R}(x, 1^{\lceil 1/\epsilon \rceil})$. Here instead of querying to $\mathcal{Q}$, run $(v_1, \dots, v_{T_x}) \leftarrow \mathcal{Q}^*(x, C_x)$ and use $(v_1, \dots, v_{T_x})$ as the outcome of $\mathcal{Q}$, where $\mathcal{Q}^*$ is the following QPT algorithm:
 - Take x and (a classical description of) a quantum circuit $C_x = (U_1, M_1, \dots, U_{T_x}, M_{T_x})$ that acts on ℓ qubits as input. For each $t \in [T_x]$, U_i is a unitary operator and M_i is a computational basis projective measurement on m_i qubits such that $0 \leq m_i \leq \ell$.
 - Run $(U_1, M_1, \dots, U_{T_x}, M_{T_x})$ on $|0^\ell\rangle$. Obtain $(u_1, \dots, u_{T_x})$, where $u_i \in \{0, 1\}^{m_i}$ is the measurement result of M_i for each $i \in [T_x]$.
 - For each $i \in [T_x]$, run $w_i \leftarrow \mathcal{A}(1^\lambda, x, i, \tau_i)$, where $\tau_i := (u_1, \dots, u_i)$. Let $v_i := u_i \| w_i$.
 - Output $(v_1, \dots, v_{T_x})$.

Later we will show that for all $\lambda \in \Lambda$,

$$\mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} [\text{SD}(\mathcal{Q}^*(x, C_x), \mathcal{Q}(C_x))] \leq \frac{1}{2p(\lambda)}. \quad (30)$$

Then by Equations (22) and (30),

$$\text{SD}(\{x, \mathcal{F}(x)\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{D}_x\}_{x \leftarrow \mathcal{E}(1^\lambda)}) \quad (31)$$

$$\leq \text{SD}(\{x, \mathcal{R}^{\mathcal{Q}^*}(x, 1^{\lfloor 1/\epsilon \rfloor})\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor})\}_{x \leftarrow \mathcal{E}(1^\lambda)}) \quad (32)$$

$$+ \text{SD}(\{x, \mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor})\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{D}_x\}_{x \leftarrow \mathcal{E}(1^\lambda)}) \quad (33)$$

$$= \text{SD}(\{x, \mathcal{R}^{\mathcal{Q}^*}(x, 1^{\lfloor 1/\epsilon \rfloor})\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor})\}_{x \leftarrow \mathcal{E}(1^\lambda)}) \quad (34)$$

$$+ \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} [\text{SD}(\mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{D}_x)] \quad (35)$$

$$\leq \text{SD}(\{x, \mathcal{R}^{\mathcal{Q}^*}(x, 1^{\lfloor 1/\epsilon \rfloor})\}_{x \leftarrow \mathcal{E}(1^\lambda)}, \{x, \mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor})\}_{x \leftarrow \mathcal{E}(1^\lambda)}) + \epsilon \quad (36)$$

$$= \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} [\text{SD}(\mathcal{R}^{\mathcal{Q}^*}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor})] + \epsilon \quad (37)$$

$$\leq \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} [\text{SD}(\mathcal{Q}^*(x, C_x), \mathcal{Q}(C_x))] + \epsilon \quad (38)$$

$$\leq \frac{1}{2p(\lambda)} + \epsilon \quad (39)$$

$$= \frac{1}{p(\lambda)} \quad (40)$$

for all $\lambda \in \Lambda$. In the last equality, we use $\epsilon = \frac{1}{2p(\lambda)}$. This contradicts Equation (21).

In the remaining part, we show Equation (30). To accomplish this, we define an unbounded-time algorithm $\mathcal{B}$ as follows:

- $\mathcal{B}(k, x, C_x) \rightarrow (v_1, \dots, v_t, v'_{t+1}, \dots, v'_{T_x})$:
 1. Take an integer $k \in \{0, \dots, T_x\}$, a bit string $x \in \{0, 1\}^*$, and (a classical description of) a quantum circuit $C_x = (U_1, M_1, \dots, U_{T_x}, M_{T_x})$ that acts on ℓ qubits as input.
 2. Run C_x on input $|0^\ell\rangle$ and let $u_t \in \{0, 1\}^{m_t}$ be the outcome of the measurement M_t for each $t \in [T_x]$. For each $t \in [T_x]$, let $\tau_t := (u_1, \dots, u_t)$ and let $|\psi_t^{\tau_t}\rangle$ be the (normalized) post-measurement state after the measurement M_t , i.e.,

$$|\psi_t^{\tau_t}\rangle := \frac{(|u_t\rangle\langle u_t| \otimes I)U_t|\psi_{t-1}^{\tau_{t-1}}\rangle}{\sqrt{\langle \psi_{t-1}^{\tau_{t-1}} | U_t^\dagger (|u_t\rangle\langle u_t| \otimes I) U_t | \psi_{t-1}^{\tau_{t-1}} \rangle}}. \quad (41)$$
 3. For $1 \leq i \leq k$, sample $v_i \in \{0, 1\}^\ell$ with probability $|\langle v_i | \psi_i^{\tau_i} \rangle|^2$.
 4. For $k+1 \leq i \leq T_x$, run $w'_i \leftarrow \mathcal{A}(1^\lambda, x, i, \tau_i)$ and let $v'_i := u_i \| w'_i$.
 5. Output $(v_1, \dots, v_k, v'_{k+1}, \dots, v'_{T_x})$.

Then, the distribution $\mathcal{B}(T_x, x, C_x)$ is equivalent to the distribution $\mathcal{Q}(C_x)$ and the distribution $\mathcal{B}(0, x, C_x)$ is equivalent to the distribution $\mathcal{Q}^*(x, C_x)$. By the triangle inequality,

$$\mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} [\text{SD}(\mathcal{Q}^*(x, C_x), \mathcal{Q}(C_x))] = \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} [\text{SD}(\mathcal{B}(0, x, C_x), \mathcal{B}(T_x, x, C_x))] \quad (42)$$

$$\leq \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} \left[\sum_{t \in [T_x]} \text{SD}(\mathcal{B}(t-1, x, C_x), \mathcal{B}(t, x, C_x)) \right]. \quad (43)$$

Thus, it suffices to show that

$$\mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} \left[\sum_{t \in [T_x]} \text{SD}(\mathcal{B}(t-1, x, C_x), \mathcal{B}(t, x, C_x)) \right] \leq \frac{1}{2p(\lambda)} \quad (44)$$

for all $\lambda \in \Lambda$.

To show this, we define two (unbounded) algorithms as follows:

- $\mathcal{Q}_1(C_x, u_1, \dots, u_t) \rightarrow (u_{t+1}, \dots, u_{T_x})$:
 1. Take (a classical description of) a quantum circuit $C_x = (U_1, M_1, \dots, U_{T_x}, M_{T_x})$ and bit strings $(u_1, \dots, u_t)$ such that $t \in [T_x]$ and $u_i \in \{0, 1\}^{m_i}$ for each $i \in [T_x]$ as input. Here for each $i \in [T_x]$, U_i is a unitary operator and M_i is a computational basis projective measurement on m_i qubits.
 2. Sample $(v'_1, \dots, v'_{T_x}) \leftarrow \mathcal{Q}(C_x)$, where $v'_i = u'_i \| w'_i$ and $u'_i \in \{0, 1\}^{m_i}$.
 3. If $u'_i = u_i$ for all $i \in [t]$, then output $(u_{t+1}, \dots, u_{T_x}) := (u'_{t+1}, \dots, u'_{T_x})$. Otherwise, go back to step 2.
- $\mathcal{Q}_2(C_x, u_1, \dots, u_t) \rightarrow (w_1, \dots, w_t)$:
 1. Take (a classical description of) a quantum circuit $C_x = (U_1, M_1, \dots, U_{T_x}, M_{T_x})$ and bit strings $(u_1, \dots, u_t)$ such that $t \in [T_x]$ and $u_i \in \{0, 1\}^{m_i}$ for each $i \in [T_x]$ as input. Here for each $i \in [T_x]$, U_i is a unitary operator and M_i is a computational basis projective measurement on m_i qubits.
 2. Sample $(v'_1, \dots, v'_{T_x}) \leftarrow \mathcal{Q}(C_x)$, where $v'_i = u'_i \| w'_i$ and $u'_i \in \{0, 1\}^{m_i}$.
 3. If $u'_i = u_i$ for all $i \in [t]$, then output $(w_1, \dots, w_t) := (w'_1, \dots, w'_t)$. Otherwise, go back to step 2.

Then,

$$\text{SD}(\mathcal{B}(t-1, x, C_x), \mathcal{B}(t, x, C_x)) \quad (45)$$

$$= \text{SD}(\{v_1, \dots, v_{t-1}, v'_t, \dots, v'_{T_x}\}, \{v_1, \dots, v_t, v'_{t+1}, \dots, v'_{T_x}\}) \quad (46)$$

$$= \sum_{w'_{t+1}, \dots, w'_{T_x}} \prod_{i \in \{t+1, \dots, T_x\}} \Pr[w'_i \leftarrow \mathcal{A}(1^\lambda, x, i, u_1, \dots, u_i)] \quad (47)$$

$$\times \text{SD}(\{v_1, \dots, v_{t-1}, v'_t, u_{t+1}, \dots, u_{T_x}\}, \{v_1, \dots, v_t, u_{t+1}, \dots, u_{T_x}\}) \quad (48)$$

$$= \text{SD}(\{v_1, \dots, v_{t-1}, v'_t, u_{t+1}, \dots, u_{T_x}\}, \{v_1, \dots, v_t, u_{t+1}, \dots, u_{T_x}\}) \quad (49)$$

$$= \sum_{u_{t+1}, \dots, u_{T_x}} \Pr[(u_{t+1}, \dots, u_{T_x}) \leftarrow \mathcal{Q}_1(C_x, u_1, \dots, u_t)] \text{SD}(\{v_1, \dots, v_{t-1}, v'_t\}, \{v_1, \dots, v_{t-1}, v_t\}) \quad (50)$$

$$= \text{SD}(\{v_1, \dots, v_{t-1}, v'_t\}, \{v_1, \dots, v_{t-1}, v_t\}) \quad (51)$$

$$= \sum_{w_1, \dots, w_{t-1}} \Pr[(w_1, \dots, w_{t-1}) \leftarrow \mathcal{Q}_2(C_x, u_1, \dots, u_{t-1})] \text{SD}(\{u_1, \dots, u_t, w'_t\}, \{u_1, \dots, u_t, w_t\}) \quad (52)$$

$$= \text{SD}(\{u_1, \dots, u_t, w'_t\}, \{u_1, \dots, u_t, w_t\}), \quad (53)$$

where $v_i = u_i \| w_i$, $v'_i = u'_i \| w'_i$, $(u_1 \| w_1, \dots, u_{T_x} \| w_{T_x}) \leftarrow \mathcal{Q}(C_x)$, and $w'_i \leftarrow \mathcal{A}(1^\lambda, x, i, u_1, \dots, u_i)$. By

Equation (29),

$$\mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} \left[\sum_{t \in [T_x]} \text{SD}(\mathcal{B}(t-1, x, C_x), \mathcal{B}(t, x, C_x)) \right] \quad (54)$$

$$= \mathbb{E}_{x \leftarrow \mathcal{E}(1^\lambda)} \left[\sum_{t \in [T_x]} \text{SD} \left(\{\tau_t, w_t\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)}, \{\tau_t, \mathcal{A}(1^\lambda, x, t, \tau_t)\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_x)} \right) \right] \quad (55)$$

$$\leq \frac{1}{2p(\lambda)} \quad (56)$$

for all $\lambda \in \Lambda$.

□

5 Adaptive PDQP and Auxiliary-Input One-Way Puzzles

In this section we consider the adaptive queries to non-collapsing measurement oracle. First, we define the class of decision problems that are solved with a polynomial-time deterministic algorithm that can make adaptive queries to the non-collapsing measurement oracle.

Definition 5.1 (AdPDQP). *A language L is in AdPDQP if there exists a polynomial-time classical deterministic Turing machine R that makes adaptive queries to a non-collapsing measurement oracle $\mathcal{Q}$ such that*

- For all $x \in L$, $\Pr[1 \leftarrow R^{\mathcal{Q}}(x)] \geq \alpha(|x|)$,
- For all $x \notin L$, $\Pr[1 \leftarrow R^{\mathcal{Q}}(x)] \leq \beta(|x|)$,

where α, β are functions such that $\alpha(|x|) - \beta(|x|) \geq 1/\text{poly}(|x|)$.

Remark 5.2. As in the case of (non-adaptive) PDQP, the error bound (α, β) in Definition 5.1 can be amplified to $(1 - \text{negl}, \text{negl})$ by the repetition.

Moreover, we define the class of sampling problems that are solved with a polynomial-time deterministic algorithm that can make adaptive queries to the non-collapsing measurement oracle.

Definition 5.3 (SampAdPDQP). *SampAdPDQP is the class of (polynomially-bounded) sampling problems $S = \{\mathcal{D}_x\}_{x \in \{0,1\}^*}$ for which there exists a classical deterministic polynomial-time algorithm $\mathcal{B}$ that makes adaptive queries to a non-collapsing measurement oracle $\mathcal{Q}$ such that for all x and for all $\epsilon > 0$, $\text{SD}(\mathcal{B}^{\mathcal{Q}}(x, 1^{\lceil 1/\epsilon \rceil}), \mathcal{D}_x) \leq \epsilon$, where $\mathcal{B}^{\mathcal{Q}}(x, 1^{\lceil 1/\epsilon \rceil})$ is the output probability distribution of $\mathcal{B}^{\mathcal{Q}}$ on input $(x, 1^{\lceil 1/\epsilon \rceil})$.*

We obtain the following lemma.

Lemma 5.4. *If AdPDQP $\not\subseteq$ BQP, then SampAdPDQP $\not\subseteq$ SampBQP.*

Next, we show that the worst-case hardness of SampAdPDQP is equivalent to that of SampPDQP.

Lemma 5.5. *SampAdPDQP $\not\subseteq$ SampBQP is and only if SampPDQP $\not\subseteq$ SampBQP.*

Proof of Lemma 5.5. The “if” direction holds immediately because $\mathbf{SampPDQP} \subseteq \mathbf{SampAdPDQP}$.

We show the “only if” direction. Assume that $\mathbf{SampAdPDQP} \not\subseteq \mathbf{SampBQP}$. Then, there exists a sampling problem $\{\mathcal{D}_x\}_{x \in \{0,1\}^*}$ that is contained in $\mathbf{SampAdPDQP}$ but not in $\mathbf{SampBQP}$. By the definition of $\mathbf{SampAdPDQP}$, there exists a classical deterministic polynomial-time algorithm $\mathcal{B}$ that makes adaptive queries to $\mathcal{Q}$ such that for all x and all $\epsilon > 0$,

$$\text{SD}(\mathcal{B}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{D}_x) \leq \epsilon. \quad (57)$$

Let $N = N(x, \epsilon)$ be the number of queries that $\mathcal{B}(x, 1^{\lfloor 1/\epsilon \rfloor})$ makes.

For the sake of contradiction, assume that $\mathbf{SampPDQP} \subseteq \mathbf{SampBQP}$. Our goal is to show that $\{\mathcal{D}_x\}_{x \in \{0,1\}^*} \in \mathbf{SampBQP}$. Let us consider the sampling problem $\{\mathcal{Q}_C\}_{C \in \{0,1\}^*}$, where $\mathcal{Q}_C$ is the output distribution of the following procedure: If C is a classical description of some quantum circuit that has the form $(U_1, M_1, \dots, U_T, M_T)$, where U_i is ℓ -qubit unitary and M_i is the m_i -qubit computational basis measurement, then query the non-collapsing measurement oracle $\mathcal{Q}$ on C . Otherwise, output $\perp$. Hence, we have $\{\mathcal{Q}_C\}_{C \in \{0,1\}^*} \in \mathbf{SampPDQP}$ and therefore $\{\mathcal{Q}_C\}_{C \in \{0,1\}^*} \in \mathbf{SampBQP}$. This means that there exists a QPT algorithm $\mathcal{A}$ such that for all $C \in \{0,1\}^*$ and for all $\epsilon > 0$,

$$\text{SD}(\mathcal{A}(C, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{Q}_C) \leq \epsilon. \quad (58)$$

For each $i \in [N]$, we define the distribution $\mathcal{B}_i(x, 1^{\lfloor 1/\epsilon \rfloor})$ as the output distribution of the following procedure: Given $(x, 1^{\lfloor 1/\epsilon \rfloor})$ as input, run $\mathcal{B}(x, 1^{\lfloor 1/\epsilon \rfloor})$, where the first i queries are made to $\mathcal{A}(\cdot, 1^{\lfloor N/\epsilon \rfloor})$ and the remaining $(N - i)$ queries are made to $\mathcal{Q}$. Let C be the classical description of the quantum circuit that $\mathcal{B}(x, 1^{\lfloor 1/\epsilon \rfloor})$ queries on its first query. Then, for all x and for all $\epsilon > 0$,

$$\text{SD}(\mathcal{B}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{B}_1(x, 1^{\lfloor 1/\epsilon \rfloor})) \leq \text{SD}(\mathcal{Q}_C, \mathcal{A}(C, 1^{\lfloor N/\epsilon \rfloor})) \leq \frac{\epsilon}{N}, \quad (59)$$

where the first inequality follows from the data processing inequality, and the second from Equation (58). Similarly, for all $i \in [N - 1]$, we have

$$\text{SD}(\mathcal{B}_i(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{B}_{i+1}(x, 1^{\lfloor 1/\epsilon \rfloor})) \leq \text{SD}(\{\mathcal{Q}_{C_i}\}_{C_i \leftarrow \mathcal{B}^{\mathcal{A}}(x, 1^{\lfloor 1/\epsilon \rfloor})}, \{\mathcal{A}(C_i, 1^{\lfloor N/\epsilon \rfloor})\}_{C_i \leftarrow \mathcal{B}^{\mathcal{A}}(x, 1^{\lfloor 1/\epsilon \rfloor})}) \quad (60)$$

$$\leq \mathbb{E}_{C_i \leftarrow \mathcal{B}^{\mathcal{A}}(x, 1^{\lfloor 1/\epsilon \rfloor})} \text{SD}(\mathcal{Q}_{C_i}, \mathcal{A}(C_i, 1^{\lfloor N/\epsilon \rfloor})) \quad (61)$$

$$\leq \frac{\epsilon}{N}, \quad (62)$$

where C_i denotes the classical description of the quantum circuit that $\mathcal{B}^{\mathcal{A}(\cdot, 1^{\lfloor N/\epsilon \rfloor})}(x, 1^{\lfloor 1/\epsilon \rfloor})$ queries on the i th query. By combining Equations (59) and (62) and using the triangle inequality, for all x and for all $\epsilon > 0$,

$$\text{SD}(\mathcal{B}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{B}_T(x, 1^{\lfloor 1/\epsilon \rfloor})) \leq \epsilon. \quad (63)$$

Note that $\mathcal{B}_T(x, 1^{\lfloor 1/\epsilon \rfloor})$ corresponds to the output distribution of QPT algorithm $\mathcal{B}^{\mathcal{A}(\cdot, 1^{\lfloor N/\epsilon \rfloor})}(x, 1^{\lfloor 1/\epsilon \rfloor})$. We consider the QPT algorithm $\mathcal{C}$ that on input $(x, 1^{\lfloor 1/\epsilon \rfloor})$, runs $\mathcal{B}^{\mathcal{A}(\cdot, 1^{\lfloor 2N/\epsilon \rfloor})}(x, 1^{\lfloor 2/\epsilon \rfloor})$. Then by Equations (57) and (63), for all x and for all $\epsilon > 0$,

$$\text{SD}(\mathcal{C}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{D}_x) \leq \text{SD}(\mathcal{C}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{B}^{\mathcal{Q}}(x, 1^{\lfloor 2/\epsilon \rfloor})) + \text{SD}(\mathcal{D}_x, \mathcal{B}^{\mathcal{Q}}(x, 1^{\lfloor 2/\epsilon \rfloor})) \quad (64)$$

$$\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \quad (65)$$

Therefore, we have $\{\mathcal{D}_x\}_{x \in \{0,1\}^*} \in \mathbf{SampBQP}$ and complete the proof. $\square$

Next, we show that the worst-case hardness of **SampPDQP** implies auxiliary-input **OWPuzzs**.

Theorem 5.6. *If **SampPDQP** $\not\subseteq$ **BQP**, then auxiliary-input **OWPuzzs** exist.*

Proof of Theorem 5.6. Because of the equivalence between auxiliary-input **OWPuzzs** and auxiliary-input **DistOWPuzzs**, it suffices to construct auxiliary-input **DistOWPuzzs**. Assume that **SampPDQP** $\not\subseteq$ **SampBQP** and let $S = \{\mathcal{D}_x\}_{x \in \{0,1\}^*}$ be a sampling problem in **SampPDQP** but not in **SampBQP**. Then, by the definition of **SampPDQP**, there exist a classical deterministic polynomial-time algorithm $\mathcal{R}$ that makes a single query to the non-collapsing measurement oracle $\mathcal{Q}$ such that for all $x \in \{0,1\}^*$ and for all $\epsilon > 0$,

$$\text{SD}(\mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{D}_x) \leq \epsilon. \quad (66)$$

For each $x \in \{0,1\}^*$ and $\epsilon > 0$, let $C_{x,\epsilon} := (U_1, M_1, \dots, U_{T_{x,\epsilon}}, M_{T_{x,\epsilon}})$ be the classical description of the quantum circuit that $\mathcal{R}(x, 1^{\lfloor 1/\epsilon \rfloor})$ queries to $\mathcal{Q}$. Here let $C_{x,\epsilon}$ act on ℓ qubits, U_i be a unitary operator, and M_i be a computational basis projective measurement on m_i qubits such that $0 \leq m_i \leq \ell$. Note that all of U_i , M_i , m_i , and ℓ also depend on x and ϵ , but we omit this dependence for notational simplicity. Define

$$T'(\lambda) := \max_{\substack{x \in \{0,1\}^{\leq \lambda}, \\ \epsilon > 0: \lfloor 1/\epsilon \rfloor \leq \lambda}} \{T_{x,\epsilon}\}. \quad (67)$$

Here $\{0,1\}^{\leq \lambda}$ is a set of bit strings x such that $|x| \leq \lambda$.

By using $\mathcal{R}$, we construct an auxiliary-input $\frac{1}{\lambda T'(\lambda)}$ -**DistOWPuzz Samp** as follows:

1. Take $z \in \{0,1\}^*$ as input, where $z = (x, 1^{\lfloor 1/\epsilon \rfloor})$ for some $x \in \{0,1\}^*$, $\epsilon > 0$.
2. Let $C_{x,\epsilon} := (U_1, M_1, \dots, U_{T_{x,\epsilon}}, M_{T_{x,\epsilon}})$ be a classical description of a quantum circuit that $\mathcal{R}(x, 1^{\lfloor 1/\epsilon \rfloor})$ queries to $\mathcal{Q}$.
3. Sample $t \leftarrow [T_{x,\epsilon}]$.
4. Run $(U_1, M_1, \dots, U_t, M_t)$ on $|0^\ell\rangle$. Obtain $\tau_t := (u_1, \dots, u_t)$ and the resulting state $|\psi_t^{\tau_t}\rangle$, where $u_i \in \{0,1\}^{m_i}$ be the measurement result of M_i .
5. Measure all qubits of $|\psi_t^{\tau_t}\rangle$ in the computational basis and obtain v_t , where $v_t = u_i \| w_i$ for some $w_i \in \{0,1\}^{\ell-m_i}$.
6. Let $\text{puzz} := (t, \tau_t)$ and $\text{ans} := w_t$. Output $(\text{puzz}, \text{ans})$.

For the sake of contradiction, we assume that **Samp** is not an auxiliary-input $\frac{1}{\lambda T'(\lambda)}$ -**DistOWPuzz**. Then by the definition of auxiliary-input **DistOWPuzzs**, there exists a QPT adversary $\mathcal{A}$ such that for all but finitely many $z \in \{0,1\}^*$,

$$\text{SD}\left(\{\text{puzz}, \text{ans}\}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(z)}, \{\text{puzz}, \mathcal{A}(z, \text{puzz})\}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(z)}\right) < \frac{1}{|z| T'(|z|)}. \quad (68)$$

Let $G := \{0,1\}^* \setminus \text{Bad}$ be a set of such $z \in \{0,1\}^*$, where $\text{Bad} \subseteq \{0,1\}^*$ is a finite subset. If $z = (x, 1^{\lfloor 1/\epsilon \rfloor})$ for some $x \in \{0,1\}^*$, $\epsilon > 0$, then

$$\Pr[(t, \tau_t, w_t) \leftarrow \text{Samp}(z)] = \frac{1}{T_{x,\epsilon}} \Pr[(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_{x,\epsilon})], \quad (69)$$

where $\mathcal{Q}^t$ is the following (unbounded) algorithm:

1. Take (a classical description of) a quantum circuit $C_{x,\epsilon} = (U_1, M_1, \dots, U_{T_{x,\epsilon}}, M_{T_{x,\epsilon}})$.
2. Sample $(v_1, \dots, v_{T_{x,\epsilon}}) \leftarrow \mathcal{Q}(C_{x,\epsilon})$, where $v_i = u_i \| w_i$ and u_i is a measurement result of M_i for each $i \in [T_{x,\epsilon}]$.
3. Let $\tau_t := (u_1, \dots, u_t)$. Output (τ_t, w_t) .

By Equation (68), for all $z \in G$ such that $z = (x, 1^{\lfloor 1/\epsilon \rfloor})$,

$$\frac{1}{|z|T'(|z|)} > \text{SD} \left(\{\text{puzz}, \text{ans}\}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(z)}, \{\text{puzz}, \mathcal{A}(z, \text{puzz})\}_{(\text{puzz}, \text{ans}) \leftarrow \text{Samp}(z)} \right) \quad (70)$$

$$= \frac{1}{T_{x,\epsilon}} \sum_{t \in [T_{x,\epsilon}]} \text{SD}(\{\tau_t, w_t\}, \{\tau_t, \mathcal{A}(z, t, \tau_t)\}), \quad (71)$$

where $(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_{x,\epsilon})$. Thus by Equation (67), for all $z \in G$ such that $z = (x, 1^{\lfloor 1/\epsilon \rfloor})$,

$$\sum_{t \in [T_{x,\epsilon}]} \text{SD}(\{\tau_t, w_t\}, \{\tau_t, \mathcal{A}(z, t, \tau_t)\}) < \epsilon, \quad (72)$$

where $(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_{x,\epsilon})$.

Our goal is to construct a QPT algorithm $\mathcal{F}$ on input $(x, 1^{\lfloor 1/\epsilon \rfloor})$ such that for all $x \in \{0, 1\}^*$ and for all $\epsilon > 0$

$$\text{SD}(\mathcal{F}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{D}_x) \leq \epsilon. \quad (73)$$

We define $\mathcal{F}$ as follows:

1. Take $x \in \{0, 1\}^*$ and $1^{\lfloor 1/\epsilon \rfloor}$ as input.
2. Let $b := \max\{|z| : z \in \text{Bad}\}$. If $|x| + \lfloor 1/\epsilon \rfloor > b$, then do the following: Run $\mathcal{R}(x, 1^{\lfloor 2/\epsilon \rfloor})$. Let $C_{x,\epsilon/2} := (U_1, M_1, \dots, U_{T_{x,\epsilon/2}}, M_{T_{x,\epsilon/2}})$ be the query that $\mathcal{R}$ makes to $\mathcal{Q}$. Instead of the query to $\mathcal{Q}$, run the following QPT algorithm $V \leftarrow \mathcal{Q}^*(x, 1^{\lfloor 2/\epsilon \rfloor}, C_{x,\epsilon/2})$ and use V as the outcome of $\mathcal{Q}$:
 - (a) Take $x, 1^{\lfloor 2/\epsilon \rfloor}$, and (a classical description of) a quantum circuit $C_{x,\epsilon/2} = (U_1, M_1, \dots, U_{T_{x,2/\epsilon}}, M_{T_{x,2/\epsilon}})$ that acts on ℓ qubits as input.
 - (b) Run $C_{x,\epsilon/2}$ on $|0^\ell\rangle$. Obtain $(u_1, \dots, u_{T_{x,\epsilon/2}})$, where $u_i \in \{0, 1\}^{m_i}$ is the measurement outcome of M_i .
 - (c) For each $i \in [T_{x,\epsilon/2}]$, run $w_i \leftarrow \mathcal{A}((x, 1^{\lfloor 2/\epsilon \rfloor}), i, u_1, \dots, u_i)$. Let $V := (u_1 \| w_1, \dots, u_{T_{x,\epsilon/2}} \| w_{T_{x,\epsilon/2}})$.
 - (d) Output V .
3. If $|x| + \lfloor 1/\epsilon \rfloor \leq b$, do the following: Run $\mathcal{R}(x, 1^{\lfloor 1/\epsilon \rfloor})$. Let $C_{x,\epsilon} := (U_1, M_1, \dots, U_{T_{x,\epsilon}}, M_{T_{x,\epsilon}})$ be the query that $\mathcal{R}$ makes to $\mathcal{Q}$. Instead of the query to $\mathcal{Q}$, run the following algorithm:
 - (a) Take (a classical description of) a quantum circuit $C_{x,\epsilon} = (U_1, M_1, \dots, U_{T_{x,\epsilon}}, M_{T_{x,\epsilon}})$ that acts on ℓ qubits as input.
 - (b) For each $i \in [T_{x,\epsilon}]$ do the following:
 - i. Run $(U_1, M_1, \dots, U_i, M_i)$ on $|0^\ell\rangle$. Obtain $\tau_i := (u'_1, \dots, u'_i)$, where $u_t \in \{0, 1\}^{m_t}$ is the measurement outcome of M_t . Let $|\psi_i^{\tau_i}\rangle$ be the resulting state.

- ii. If $(u'_1, \dots, u'_{i-1}) = (u_1, \dots, u_{i-1})$, then proceed to the next step. Otherwise, go back to the previous step.
 - iii. Let $u_i = u'_i$. Measure all qubits of $|\psi_i^{\tau_i}\rangle$ in the computational basis and obtain $v_i \in \{0, 1\}^\ell$, where $v_i = u_i \| w_i$ for some $w_i \in \{0, 1\}^{\ell-m_i}$.
- (c) Output $(v_1, \dots, v_{T_{x,\epsilon}})$.

Note that if x and $1^{\lfloor 1/\epsilon \rfloor}$ satisfies $|x| + \lfloor 1/\epsilon \rfloor \leq b$, then $\mathcal{F}(x, 1^{\lfloor 1/\epsilon \rfloor})$ runs in constant time. This is because Bad is the finite set and therefore b is constant that does not depend on $|x|$ and $\lfloor 1/\epsilon \rfloor$. Moreover, if $|x| + \lfloor 1/\epsilon \rfloor \leq b$, then the output distribution of $\mathcal{F}(x, 1^{\lfloor 1/\epsilon \rfloor})$ is equivalent to $\mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor})$. Thus, for all $x \in \{0, 1\}^*$ and $\epsilon > 0$ that satisfies $|x| + \lfloor 1/\epsilon \rfloor \leq b$,

$$\text{SD}(\mathcal{F}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{D}_x) = \text{SD}(\mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{D}_x) \leq \epsilon. \quad (74)$$

Next, we consider the case where $|x| + \lfloor 1/\epsilon \rfloor > b$. Later we will show that

$$\text{SD}(\mathcal{Q}^*(x, 1^{\lfloor 1/\epsilon \rfloor}, C_{x,\epsilon}), \mathcal{Q}(C_{x,\epsilon})) \leq \epsilon \quad (75)$$

for all $x \in \{0, 1\}^*$ and $\epsilon > 0$ such that $|x| + \lfloor 1/\epsilon \rfloor > b$. Then,

$$\text{SD}(\mathcal{F}(x, 1^{\lfloor 1/\epsilon \rfloor}), \mathcal{D}_x) \leq \text{SD}(\mathcal{R}^{\mathcal{Q}^*}(x, 1^{\lfloor 2/\epsilon \rfloor}), \mathcal{R}^{\mathcal{Q}}(x, 1^{\lfloor 2/\epsilon \rfloor})) + \frac{\epsilon}{2} \quad (76)$$

$$\leq \text{SD}(\mathcal{Q}^*(x, 1^{\lfloor 2/\epsilon \rfloor}, C_{x,\epsilon/2}), \mathcal{Q}(C_{x,\epsilon/2})) + \frac{\epsilon}{2} \quad (77)$$

$$\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} \leq \epsilon. \quad (78)$$

Therefore, $\mathcal{F}$ satisfies Equation (73).

In the remaining part, we show Equation (75). To accomplish this, we define the following (unbounded) algorithm $\mathcal{B}$:

- $\mathcal{B}(k, z, C_{x,\epsilon}) \rightarrow (v_1, \dots, v_k, v'_{k+1}, \dots, v'_{T_{x,\epsilon}})$:
 1. Take $k \in \{0, \dots, T_{x,\epsilon}\}$, $z = (x, 1^{\lfloor 1/\epsilon \rfloor})$, and (a classical description of) a quantum circuit $C_{x,\epsilon} = (U_1, M_1, \dots, U_{T_{x,\epsilon}}, M_{T_{x,\epsilon}})$ that acts on ℓ qubits as input.
 2. Run $C_{x,\epsilon}$ on input $|0^\ell\rangle$ and let $u_t \in \{0, 1\}^{m_t}$ be the outcome of the measurement M_t for each $t \in [T_{x,\epsilon}]$. For each $t \in [T_{x,\epsilon}]$, let $\tau_t := (u_1, \dots, u_t)$ and let $|\psi_t^{\tau_t}\rangle$ be the (normalized) post-measurement state after the measurement M_t , i.e.,

$$|\psi_t^{\tau_t}\rangle := \frac{(|u_t\rangle\langle u_t| \otimes I)U_t|\psi_{t-1}^{\tau_{t-1}}\rangle}{\sqrt{\langle \psi_{t-1}^{\tau_{t-1}} | U_t^\dagger (|u_t\rangle\langle u_t| \otimes I) U_t | \psi_{t-1}^{\tau_{t-1}} \rangle}}. \quad (79)$$
 3. For $1 \leq i \leq k$, sample $v_i \in \{0, 1\}^\ell$ with probability $|\langle v_i | \psi_i^{\tau_i} \rangle|^2$.
 4. For $k+1 \leq i \leq T_{x,\epsilon}$, run $w'_i \leftarrow \mathcal{A}(z, i, \tau_i)$ and let $v'_i := u_i \| w'_i$.
 5. Output $(v_1, \dots, v_k, v'_{k+1}, \dots, v'_{T_{x,\epsilon}})$.

Then, the distribution $\mathcal{B}(T_{x,\epsilon}, z, C_{x,\epsilon})$ is equivalent to the distribution $\mathcal{Q}(C_{x,\epsilon})$ and the distribution $\mathcal{B}(0, z, C_{x,\epsilon})$ is equivalent to the distribution $\mathcal{Q}^*(x, 1^{\lfloor 1/\epsilon \rfloor}, C_{x,\epsilon})$. By the triangle inequality, it suffices to show

$$\sum_{t \in [T_{x,\epsilon}]} \text{SD}(\mathcal{B}(t-1, z, C_{x,\epsilon}), \mathcal{B}(t, z, C_{x,\epsilon})) \leq \epsilon \quad (80)$$

for all $x \in \{0, 1\}^*$ and $\epsilon > 0$ such that $|x| + \lfloor 1/\epsilon \rfloor > b$. Indeed, for all $x \in \{0, 1\}^*$, $\epsilon > 0$, and for all $t \in [T_{x,\epsilon}]$,

$$\text{SD}(\mathcal{B}(t-1, z, C_{x,\epsilon}), \mathcal{B}(t, z, C_{x,\epsilon})) \quad (81)$$

$$= \text{SD}\left(\{\tau_t, w_t\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_{x,\epsilon})}, \{\tau_t, \mathcal{A}(z, t, \tau_t)\}_{(\tau_t, w_t) \leftarrow \mathcal{Q}^t(C_{x,\epsilon})}\right), \quad (82)$$

where $z = (x, 1^{\lfloor 1/\epsilon \rfloor})$. We can obtain the above equality by the same way as Equation (53). If $|x| + \lfloor 1/\epsilon \rfloor > b$, then $|z| > b$ and therefore $z \in G$. By Equation (72), for all $x \in \{0, 1\}^*$ and $\epsilon > 0$ such that $|x| + \lfloor 1/\epsilon \rfloor > b$,

$$\sum_{t \in [T_{x,\epsilon}]} \text{SD}(\mathcal{B}(t-1, z, C_{x,\epsilon}), \mathcal{B}(t, z, C_{x,\epsilon})) \leq \epsilon. \quad (83)$$

□

By combining Lemma 5.5 and Theorem 5.6, we obtain the following corollary.

Corollary 5.7. *If $\text{SampAdPDQP} \not\subseteq \text{BQP}$, then auxiliary-input OWPuzzs exist.*

6 Distributional Collision-Resistant Puzzles

In this section, we introduce a quantum analogue of dCRH, namely, distributional collision-resistant puzzles (dCRPuzzs).

6.1 Definition of classical dCRH

Before introducing the quantum analogue, we first remind the definition of classical dCRH for the convenience of readers.

Definition 6.1 (Distributional Collision-Resistant Hashing (dCRH) [DI06, BHKY19]). *Let $\{\mathcal{H}_\lambda : \{0, 1\}^{n(\lambda)} \rightarrow \{0, 1\}^{m(\lambda)}\}_{\lambda \in \mathbb{N}}$ be an efficient function family ensemble. Here n and m are polynomials. We say that it is a distributional collision-resistant hash (dCRH) function family if there exists a polynomial p such that for any QPT algorithm $\mathcal{A}$,*

$$\text{SD}(\{h, \mathcal{A}(h)\}_{h \leftarrow \mathcal{H}_\lambda}, \{h, \text{Col}(h)\}_{h \leftarrow \mathcal{H}_\lambda}) \geq \frac{1}{p(\lambda)} \quad (84)$$

for all sufficiently large $\lambda \in \mathbb{N}$. Here $\text{Col}(h)$ is the following distribution.

1. Sample $x \leftarrow \{0, 1\}^{n(\lambda)}$.
2. Sample $x' \leftarrow h^{-1}(h(x))$.
3. Output (x, x') .

dCRH imply distributional OWFs [BHKY19], and therefore OWFs. Average-case hardness of SZK imply dCRH [KY18, BHKY19]. dCRH will not be constructed from one-way permutations (OWPs) in a black-box way [Sim98, DI06]. dCRH will not be constructed from iO plus OWPs in a black-box way [AS16]. dCRH implies constant-round statistically-hiding commitments [BHKY19]. Two-message statistically-hiding commitments imply dCRH [BHKY19].

6.2 Definition of dCRPuzzs

Next we introduce dCRPuzzs.

Definition 6.2 (Distributional Collision-Resistant Puzzles (dCRPuzzs)). A distributional collision-resistant puzzle ($dCRPuzz$) is a pair $(Setup, Samp)$ of algorithms such that

- $Setup(1^\lambda) \rightarrow pp$: It is a QPT algorithm that, on input the security parameter λ , outputs a classical public parameter pp .
- $Samp(pp) \rightarrow (puzz, ans)$: It is a QPT algorithm that, on input pp , outputs two bit strings $(puzz, ans)$.

We require the following property: there exists a polynomial p such that for any QPT adversary $\mathcal{A}$

$$SD(\{pp, \mathcal{A}(pp)\}_{pp \leftarrow Setup(1^\lambda)}, \{pp, Col(pp)\}_{pp \leftarrow Setup(1^\lambda)}) \geq \frac{1}{p(\lambda)} \quad (85)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where $Col(pp)$ is the following distribution:

1. Run $(puzz, ans) \leftarrow Samp(pp)$.
2. Sample ans' with the conditional probability $\Pr[ans' | puzz] := \frac{\Pr[(ans', puzz) \leftarrow Samp(pp)]}{\Pr[puzz \leftarrow Samp(pp)]}$.
3. Output $(puzz, ans, ans')$.

The following lemma is easy to show.

Lemma 6.3. If (quantumly-secure) dCRH exists, then dCRPuzzs exist.

6.3 dCRPuzzs imply average-case hardness of SampPDQP

Theorem 6.4. If dCRPuzzs exist, then SampPDQP is hard on average.

Proof. Let $(Setup, Samp)$ be a dCRPuzz. Without loss of generality, we can assume that $Samp(pp) \rightarrow (puzz, ans)$ runs as follows.

1. Apply a unitary V_{pp} on $|0\dots 0\rangle$ to generate a state,

$$V_{pp}|0\dots 0\rangle = \sum_{puzz, ans} c_{puzz, ans} |puzz\rangle_A |ans\rangle_B |junk_{puzz, ans}\rangle, \quad (86)$$

where $c_{puzz, ans}$ is a complex coefficient, and $|junk_{puzz, ans}\rangle$ is a “junk” state.

2. Measure the register **A** to get $puzz$.
3. Measure the register **B** to get ans .
4. Output $(puzz, ans)$.

Define the following distribution $\mathcal{D}_{pp}$.

1. Apply V_{pp} on $|0\dots 0\rangle$ to generate

$$V_{pp}|0\dots 0\rangle = \sum_{puzz, ans} c_{puzz, ans} |puzz\rangle_A |ans\rangle_B |junk_{puzz, ans}\rangle_C. \quad (87)$$

2. Measure the register **A**. The state is collapsed to

$$|\text{puzz}\rangle \otimes \left(\sum_{\text{ans}} c_{\text{puzz}, \text{ans}} |\text{ans}\rangle_{\text{B}} |junk_{\text{puzz}, \text{ans}}\rangle_{\text{C}} \right) \quad (88)$$

up to the normalization.

3. Perform the non-collapsing measurement on this state to sample $v_1 := (\text{puzz}, \text{ans}, junk)$.
4. Perform the non-collapsing measurement on this state to sample $v_2 := (\text{puzz}, \text{ans}', junk')$.
5. Output $(\text{puzz}, \text{ans}, \text{ans}')$.

It is clear that the sampling problem $\{\mathcal{D}_{\text{pp}}\}_{\text{pp}}$ is in **SampPDQP**. (The classical polynomial-time deterministic base algorithm has only to query (U_1, M_1, U_2, M_2) to the non-collapsing measurement oracle, where $U_1 = V_{\text{pp}}$, M_1 is the measurement after the application of V_{pp} , U_2 is the identity, and M_2 does not do any measurement.)

Assume that **SampPDQP** is not hard on average. Then, from the definition of average-case hardness of **SampPDQP** (Definition 3.9), we have that for any polynomial p there exists a QPT algorithm $\mathcal{F}$ such that

$$\text{SD}(\{\text{pp}, \mathcal{F}(\text{pp})\}_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}, \{\text{pp}, \mathcal{D}_{\text{pp}}\}_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}) \leq \frac{1}{p(\lambda)} \quad (89)$$

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

From such $\mathcal{F}$, we construct a QPT adversary $\mathcal{A}$ that breaks the dCRPuzz as follows.

1. Receive pp as input.
2. Run $(\text{puzz}, \text{ans}, \text{ans}') \leftarrow \mathcal{F}(\text{pp})$.
3. Output $(\text{puzz}, \text{ans}, \text{ans}')$.

Then,

$$\text{SD}(\{\text{pp}, \mathcal{A}(\text{pp})\}_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}, \{\text{pp}, \text{Col}(\text{pp})\}_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}) \quad (90)$$

$$= \text{SD}(\{\text{pp}, \mathcal{F}(\text{pp})\}_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}, \{\text{pp}, \text{Col}(\text{pp})\}_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}) \quad (91)$$

$$= \text{SD}(\{\text{pp}, \mathcal{F}(\text{pp})\}_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}, \{\text{pp}, \mathcal{D}_{\text{pp}}\}_{\text{pp} \leftarrow \text{Setup}(1^\lambda)}) \quad (92)$$

$$\leq \frac{1}{p(\lambda)} \quad (93)$$

for infinitely-many $\lambda \in \mathbb{N}$, which means that $\mathcal{A}$ breaks the dCRPuzz, but it is the contradiction. $\square$

7 One-Shot Signatures and MACs

7.1 Definitions

We first remind the definition of one-shot signatures.

Definition 7.1 (One-Shot Signatures [AGKZ20]). A one-shot signature scheme is a set $(\text{Setup}, \text{Gen}, \text{Sign}, \text{Ver})$ of algorithms such that

- $\text{Setup}(1^\lambda) \rightarrow \text{pp}$: It is a QPT algorithm that, on input the security parameter λ , outputs a public parameter pp .
- $\text{Gen}(\text{pp}) \rightarrow (\text{vk}, \text{sigk})$: It is a QPT algorithm that, on input pp , outputs a quantum signing key sigk and a classical verification key vk .
- $\text{Sign}(\text{sigk}, m) \rightarrow \sigma$: It is a QPT algorithm that, on input sigk and a message m , outputs a classical signature σ .
- $\text{Ver}(\text{pp}, \text{vk}, \sigma, m) \rightarrow \top/\perp$: It is a QPT algorithm that, on input pp , vk , σ , and m , outputs $\top/\perp$.

We require the following properties.

Correctness: For any m ,

$$\Pr \left[\begin{array}{l} \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, \sigma, m) : \\ \text{pp} \leftarrow \text{Setup}(1^\lambda) \\ (\text{vk}, \text{sigk}) \leftarrow \text{Gen}(\text{pp}) \\ \sigma \leftarrow \text{Sign}(\text{sigk}, m) \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (94)$$

Security: For any QPT adversary $\mathcal{A}$,

$$\Pr \left[\begin{array}{l} m_0 \neq m_1 \\ \wedge \\ \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, \sigma_0, m_0) : \\ \wedge \\ \top \leftarrow \text{Ver}(\text{pp}, \text{vk}, \sigma_1, m_1) \end{array} : \begin{array}{l} \text{pp} \leftarrow \text{Setup}(1^\lambda) \\ (\text{vk}, m_0, m_1, \sigma_0, \sigma_1) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \leq \text{negl}(\lambda). \quad (95)$$

One-shot MACs are a relaxation of one-shot signatures and two-tier one-shot signatures [MPY23], which have partial public verification. One-shot MACs can be constructed from the LWE assumption [MPY23, CKNY24].

Definition 7.2 (One-Shot MACs [CKNY24]). A one-shot message authentication code (MAC) scheme is a set $(\text{Setup}, \text{Gen}, \text{Sign}, \text{Ver})$ of algorithms such that

- $\text{Setup}(1^\lambda) \rightarrow (\text{pp}, \text{mvk})$: It is a QPT algorithm that, on input the security parameter λ , outputs a public parameter pp and a master verification key mvk .
- $\text{Gen}(\text{pp}) \rightarrow (\text{vk}, \text{sigk})$: It is a QPT algorithm that, on input pp , outputs a quantum signing key sigk and a classical verification key vk .
- $\text{Sign}(\text{sigk}, m) \rightarrow \sigma$: It is a QPT algorithm that, on input sigk and a message m , outputs a classical signature σ .
- $\text{Ver}(\text{pp}, \text{mvk}, \text{vk}, \sigma, m) \rightarrow \top/\perp$: It is a QPT algorithm that, on input pp , mvk , vk , σ , and m , outputs $\top/\perp$.

We require the following properties.

Correctness: For any m ,

$$\Pr \left[\begin{array}{l} \top \leftarrow \text{Ver}(\text{pp}, \text{mvk}, \text{vk}, \sigma, m) : \\ \begin{array}{l} (\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{sigk}, \text{vk}) \leftarrow \text{Gen}(\text{pp}) \\ \sigma \leftarrow \text{Sign}(\text{sigk}, m) \end{array} \end{array} \right] \geq 1 - \text{negl}(\lambda). \quad (96)$$

Security: For any QPT adversary $\mathcal{A}$,

$$\Pr \left[\begin{array}{l} m_0 \neq m_1 \\ \wedge \\ \top \leftarrow \text{Ver}(\text{pp}, \text{mvk}, \text{vk}, \sigma_0, m_0) : \\ \wedge \\ \top \leftarrow \text{Ver}(\text{pp}, \text{mvk}, \text{vk}, \sigma_1, m_1) \end{array} \begin{array}{l} (\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda) \\ (\text{vk}, m_0, m_1, \sigma_0, \sigma_1) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] \leq \text{negl}(\lambda). \quad (97)$$

7.2 One-shot MACs imply dCRPuzzs

Theorem 7.3. If one-shot MACs exist, then dCRPuzzs exist.

Proof. Let $(\text{Setup}, \text{Gen}, \text{Sign}, \text{Ver})$ be a one-shot MAC. Define the algorithm $\mathcal{C}$ as follows:

1. Get pp as input.
2. Run $(\text{vk}, \text{sigk}) \leftarrow \text{Gen}(\text{pp})$.
3. Choose $m_0 \leftarrow \{0, 1\}^\ell$. Run $\sigma_0 \leftarrow \text{Sign}(\text{sigk}, m_0)$.
4. Run $\text{Gen}(\text{pp})$ until vk is obtained.
5. Choose $m_1 \leftarrow \{0, 1\}^\ell$. Run $\sigma_1 \leftarrow \text{Sign}(\text{sigk}, m_1)$.
6. Output $(\text{vk}, m_0, \sigma_0, m_1, \sigma_1)$.

Let Π be a POVM element corresponding to the event that the challenger of the security game of one-shot MACs accepts. From the correctness of the one-shot MAC, we have

$$\sum_{\text{pp}, \text{mvk}} \Pr[(\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda)] \text{Tr}[\Pi(|\text{mvk}\rangle\langle\text{mvk}| \otimes |\text{pp}\rangle\langle\text{pp}| \otimes \mathcal{C}(\text{pp}))] \geq 1 - \text{negl}(\lambda). \quad (98)$$

Let q be a polynomial. Define the set G as

$$G := \left\{ (\text{pp}, \text{mvk}) : \text{Tr}[\Pi(|\text{mvk}\rangle\langle\text{mvk}| \otimes |\text{pp}\rangle\langle\text{pp}| \otimes \mathcal{C}(\text{pp}))] \geq 1 - \frac{1}{q(\lambda)} \right\}. \quad (99)$$

Then, from the standard average argument,

$$\sum_{(\text{pp}, \text{mvk}) \in G} \Pr[(\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda)] \geq 1 - \text{negl}(\lambda). \quad (100)$$

We construct a dCRPuzz $(\text{d.Setup}, \text{d.Samp})$ as follows.

- $\text{d.Setup}(1^\lambda) \rightarrow \text{d.pp} : \text{Run } (\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda)$. Output $\text{d.pp} := \text{pp}$.

- $d.\text{Samp}(d.\text{pp}) \rightarrow (\text{puzz}, \text{ans})$: Parse $d.\text{pp} = \text{pp}$. Run $(\text{vk}, \text{sigk}) \leftarrow \text{Gen}(\text{pp})$. Choose $m \leftarrow \{0, 1\}^\ell$. Run $\sigma \leftarrow \text{Sign}(\text{sigk}, m)$. Output $\text{puzz} := \text{vk}$ and $\text{ans} := (m, \sigma)$.

For the sake of contradiction, assume that this is not a $d\text{CRPuzz}$. Then, for any polynomial p , there exists a QPT adversary $\mathcal{A}$ such that

$$\text{SD}((d.\text{pp}, \mathcal{A}(d.\text{pp}))_{d.\text{pp} \leftarrow d.\text{Setup}(1^\lambda)}, (d.\text{pp}, \text{Col}(d.\text{pp}))_{d.\text{pp} \leftarrow d.\text{Setup}(1^\lambda)}) \leq \frac{1}{p(\lambda)} \quad (101)$$

for infinitely many $\lambda \in \mathbb{N}$. Here $\text{Col}(d.\text{pp})$ is the following distribution.

1. Run $(\text{puzz}, \text{ans}) \leftarrow d.\text{Samp}(d.\text{pp})$.
2. Sample ans' with the conditional probability $\Pr[\text{ans}' | \text{puzz}] := \frac{\Pr[(\text{ans}', \text{puzz}) \leftarrow d.\text{Samp}(d.\text{pp})]}{\Pr[\text{puzz} \leftarrow d.\text{Samp}(d.\text{pp})]}$.
3. Output $(\text{puzz}, \text{ans}, \text{ans}')$.

From Equation (101), we have

$$\frac{1}{p(\lambda)} \geq \text{SD}((d.\text{pp}, \mathcal{A}(d.\text{pp}))_{d.\text{pp} \leftarrow d.\text{Setup}(1^\lambda)}, (d.\text{pp}, \text{Col}(d.\text{pp}))_{d.\text{pp} \leftarrow d.\text{Setup}(1^\lambda)}) \quad (102)$$

$$\geq \sum_{d.\text{pp}} \Pr[d.\text{pp} \leftarrow d.\text{Setup}(1^\lambda)] \text{TD}(|d.\text{pp}\rangle\langle d.\text{pp}| \otimes \mathcal{A}(d.\text{pp}), |d.\text{pp}\rangle\langle d.\text{pp}| \otimes \text{Col}(d.\text{pp})) \quad (103)$$

$$= \sum_{\text{pp}} \Pr[\text{pp} \leftarrow \text{Setup}(1^\lambda)] \text{TD}(|\text{pp}\rangle\langle \text{pp}| \otimes \mathcal{A}(\text{pp}), |\text{pp}\rangle\langle \text{pp}| \otimes \mathcal{C}(\text{pp})). \quad (104)$$

If we define the set S as

$$S := \left\{ \text{pp} : \text{TD}(|\text{pp}\rangle\langle \text{pp}| \otimes \mathcal{A}(\text{pp}), |\text{pp}\rangle\langle \text{pp}| \otimes \mathcal{C}(\text{pp})) \leq \frac{1}{\sqrt{p(\lambda)}} \right\}, \quad (105)$$

we have

$$\sum_{\text{pp} \in S} \Pr[\text{pp} \leftarrow \text{Setup}(1^\lambda)] \geq 1 - \frac{1}{\sqrt{p(\lambda)}} \quad (106)$$

from the standard average argument.

From $\mathcal{A}$, we can construct a QPT adversary $\mathcal{B}$ that breaks the security of the one-shot MAC as follows.

1. Receive pp as input.
2. Run $(\text{vk}, m_0, \sigma_0, m_1, \sigma_1) \leftarrow \mathcal{A}(\text{pp})$.
3. Output $(\text{vk}, m_0, \sigma_0, m_1, \sigma_1)$.

The probability that $\mathcal{B}$ wins is

$$\sum_{\text{pp}, \text{mvk}} \Pr[(\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda)] \text{Tr}[\Pi(|\text{mvk}\rangle\langle\text{mvk}| \otimes |\text{pp}\rangle\langle\text{pp}| \otimes \mathcal{B}(\text{pp}))] \quad (107)$$

$$\geq \sum_{(\text{pp}, \text{mvk}) \in G \wedge \text{pp} \in S} \Pr[(\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda)] \text{Tr}[\Pi(|\text{mvk}\rangle\langle\text{mvk}| \otimes |\text{pp}\rangle\langle\text{pp}| \otimes \mathcal{B}(\text{pp}))] \quad (108)$$

$$= \sum_{(\text{pp}, \text{mvk}) \in G \wedge \text{pp} \in S} \Pr[(\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda)] \text{Tr}[\Pi(|\text{mvk}\rangle\langle\text{mvk}| \otimes |\text{pp}\rangle\langle\text{pp}| \otimes \mathcal{A}(\text{pp}))] \quad (109)$$

$$\geq \sum_{(\text{pp}, \text{mvk}) \in G \wedge \text{pp} \in S} \Pr[(\text{pp}, \text{mvk}) \leftarrow \text{Setup}(1^\lambda)] \text{Tr}[\Pi(|\text{mvk}\rangle\langle\text{mvk}| \otimes |\text{pp}\rangle\langle\text{pp}| \otimes \mathcal{C}(\text{pp}))] - \frac{1}{\sqrt{p(\lambda)}} \quad (110)$$

$$\geq \left(1 - \frac{1}{\sqrt{p(\lambda)}}\right) \left(1 - \frac{1}{q(\lambda)}\right) - \frac{1}{\sqrt{p(\lambda)}} \quad (111)$$

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

8 Commitments

8.1 Definitions

We first remind the definition of commitments we consider.

Definition 8.1 (Two-Message Honest-Statistically-Hiding Computationally-Binding Bit Commitments with Classical Communication). A two-message honest-statistically-hiding and computationally-binding bit commitment scheme with classical communication is a set (S_1, S_2, R_1, R_2) of algorithms such that

1. $R_1(1^\lambda) \rightarrow (r_1, \psi_R)$: It is a QPT algorithm that, on input the security parameter λ , outputs a classical bit string r_1 and an internal quantum state ψ_R .
2. $S_1(r_1, b) \rightarrow (s_1, \psi_S)$: It is a QPT algorithm that, on input r_1 and a bit $b \in \{0, 1\}$, outputs a bit string s_1 and an internal state ψ_S .
3. $S_2(b, \psi_S) \rightarrow s_2$: It is a QPT algorithm that, on input b and ψ_S , outputs a bit string s_2 .
4. $R_2(\psi_R, s_1, s_2, b) \rightarrow \top / \perp$: It is a QPT algorithm that, on input ψ_R , s_1 , s_2 , and b , outputs $\top / \perp$.

We require the following properties.

Correctness. For all $b \in \{0, 1\}$,

$$\Pr[\top \leftarrow R_2(\psi_R, s_1, s_2, b) : (r_1, \psi_R) \leftarrow R_1(1^\lambda), (s_1, \psi_S) \leftarrow S_1(r_1, b), s_2 \leftarrow S_2(b, \psi_S)] \geq 1 - \text{negl}(\lambda). \quad (112)$$

Honest statistical hiding. For all $b \in \{0, 1\}$ and for any (not-necessarily-efficient) algorithm $\mathcal{A}$,

$$\Pr[b \leftarrow \mathcal{A}(\psi_R, s_1) : (r_1, \psi_R) \leftarrow R_1(1^\lambda), s_1 \leftarrow S_1(r_1, b)] \leq \frac{1}{2} + \text{negl}(\lambda). \quad (113)$$

Computational binding. For any QPT algorithm $\mathcal{A}$,

$$\Pr \left[\begin{array}{c} \top \leftarrow R_2(\psi_R, s_1, s_2, 0) \\ \wedge \\ \top \leftarrow R_2(\psi_R, s_1, s'_2, 1) \end{array} : \begin{array}{c} (r_1, \psi_R) \leftarrow R_1(1^\lambda) \\ (s_1, s_2, s'_2) \leftarrow \mathcal{A}(r_1) \end{array} \right] \leq \text{negl}(\lambda). \quad (114)$$

8.2 Commitments imply dCRPuzzs

Theorem 8.2. *If two-message honest-statistically-hiding computationally-binding bit commitments with classical communication exist, then dCRPuzzs exist.*

Proof. Let (R_1, R_2, S_1, S_2) be a two-message honest-statistically-hiding computationally-binding bit commitment scheme with classical communication.

Define the following algorithm $\mathcal{C}$:

1. Get r_1 as input.
2. Run $(s_1, \psi_S) \leftarrow S_1(r_1, 0)$. Run $s_2 \leftarrow S_2(0, \psi_S)$.
3. Generate ψ_S . Run $s'_2 \leftarrow S_2(1, \psi_S)$.
4. Output (s_1, s_2, s'_2) .

Let Π be a POVM element corresponding to the event that the challenger of the security game of binding accepts. Then, from the correctness and statistical hiding of the commitment scheme,

$$\sum_{r_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \mathcal{C}(r_1))] \geq 1 - \frac{1}{q(\lambda)} \quad (115)$$

for a certain polynomial q . We will show it later. If we define the set

$$V := \left\{ r_1 : \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \mathcal{C}(r_1))] \geq 1 - \frac{1}{\sqrt{q(\lambda)}} \right\}, \quad (116)$$

we have

$$\sum_{r_1 \in V} \Pr[r_1 \leftarrow R_1(1^\lambda)] \geq 1 - \frac{1}{\sqrt{q(\lambda)}} \quad (117)$$

from the standard average argument.

From the commitment scheme, we construct a dCRPuzz (Setup, Samp) as follows.

- Setup(1^λ) $\rightarrow$ pp : Run $(r_1, \psi_R) \leftarrow R_1(1^\lambda)$. Output pp $:= r_1$.
- Samp(pp) $\rightarrow$ (puzz, ans) :
 1. Parse pp $= r_1$.
 2. Run $(s_1, \psi_S) \leftarrow S_1(r_1, 0)$.
 3. Choose $b \leftarrow \{0, 1\}$.
 4. Run $s_2 \leftarrow S_2(b, \psi_S)$.

5. Output $\text{puzz} := s_1$ and $\text{ans} := (b, s_2)$.

For the sake of contradiction, assume that it is not a dCRPuzz. Then for any polynomial p , there exists a QPT algorithm $\mathcal{A}$ such that

$$\text{SD}((\text{pp}, \mathcal{A}(\text{pp}))_{\text{pp} \leftarrow \text{Samp}(1^\lambda)}, (\text{pp}, \text{Col}(\text{pp}))_{\text{pp} \leftarrow \text{Samp}(1^\lambda)}) \leq \frac{1}{p(\lambda)} \quad (118)$$

for infinitely-many $\lambda \in \mathbb{N}$. This means that

$$\frac{1}{p(\lambda)} \geq \sum_{r_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{TD}[\mathcal{A}(r_1), \text{Col}(r_1)]. \quad (119)$$

If we define the set

$$G := \{r_1 : \text{TD}[\mathcal{A}(r_1), \text{Col}(r_1)] \leq \frac{1}{\sqrt{p(\lambda)}}\}, \quad (120)$$

we have

$$\sum_{r_1 \in G} \Pr[r_1 \leftarrow R_1(1^\lambda)] \geq 1 - \frac{1}{\sqrt{p(\lambda)}} \quad (121)$$

from the standard average argument.

From $\mathcal{A}$, we construct a QPT adversary $\mathcal{B}$ that breaks the binding of the commitment scheme as follows.

1. Get r_1 as input.
2. Run $(\text{puzz}, \text{ans}, \text{ans}') \leftarrow \mathcal{A}(r_1)$.
3. Parse $\text{puzz} = s_1$, $\text{ans} = b \| s_2$, and $\text{ans}' = b' \| s'_2$.
4. Output (s_1, s_2, s'_2) .

Let Π be a POVM element corresponding to the event that the challenger of the security game of the binding accepts. The probability that $\mathcal{B}$ wins is

$$\sum_{r_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \mathcal{B}(r_1))] \quad (122)$$

$$= \sum_{r_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \mathcal{A}(r_1))] \quad (123)$$

$$\geq \sum_{r_1 \in G} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \mathcal{A}(r_1))] \quad (124)$$

$$\geq \sum_{r_1 \in G} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \text{Col}(r_1))] - \frac{1}{\sqrt{p(\lambda)}} \quad (125)$$

$$\geq \frac{1}{4} \sum_{r_1 \in G} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \mathcal{C}(r_1))] - \frac{1}{\sqrt{p(\lambda)}} \quad (126)$$

$$\geq \frac{1}{4} \sum_{r_1 \in G \cap V} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \mathcal{C}(r_1))] - \frac{1}{\sqrt{p(\lambda)}} \quad (127)$$

$$\geq \frac{1}{4} \left(1 - \frac{1}{\sqrt{q(\lambda)}} - \frac{1}{\sqrt{p(\lambda)}}\right) \left(1 - \frac{1}{\sqrt{q(\lambda)}}\right) - \frac{1}{\sqrt{p(\lambda)}} \quad (128)$$

$$\geq \frac{1}{\text{poly}(\lambda)}. \quad (129)$$

Let us show Equation (117). For each $b \in \{0, 1\}$, because of the correctness,

$$\sum_{r_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \sum_{s_1} \Pr[s_1 \leftarrow S_1(r_1, b)] \sum_{s_2} \Pr[s_2 \leftarrow S_2(\psi_S, b)] \Pr[\top \leftarrow R_2(\psi_R, s_1, s_2, b)] \geq 1 - \text{negl}(\lambda). \quad (130)$$

Let p be a polynomial. For each $b \in \{0, 1\}$, if we define the set

$$G_b := \left\{ (r_1, s_1) : \sum_{s_2} \Pr[s_2 \leftarrow S_2(\psi_S, b)] \Pr[\top \leftarrow R_2(\psi_R, s_1, s_2, b)] \geq 1 - \frac{1}{p(\lambda)} \right\}, \quad (131)$$

we have

$$\sum_{(r_1, s_1) \in G_b} \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, b)] \geq 1 - \text{negl}(\lambda) \quad (132)$$

from the standard average argument. Because of the statistical hiding,

$$\text{negl}(\lambda) \geq \sum_{(r_1, s_1)} \left| \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 0)] - \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 1)] \right| \quad (133)$$

$$\geq \sum_{(r_1, s_1) \in G_1} \left| \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 0)] - \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 1)] \right| \quad (134)$$

$$\geq \left| \sum_{(r_1, s_1) \in G_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 0)] - \sum_{(r_1, s_1) \in G_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 1)] \right|. \quad (135)$$

Therefore from the last inequality and Equation (132) with $b = 1$, we have

$$\sum_{(r_1, s_1) \in G_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 0)] \geq 1 - \text{negl}(\lambda). \quad (136)$$

Hence

$$\sum_{r_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \text{Tr}[\Pi(\psi_R^{\otimes 2} \otimes \mathcal{C}(r_1))] \quad (137)$$

$$= \sum_{(r_1, s_1)} \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 0)] \sum_{s_2} \Pr[s_2 \leftarrow S_2(\psi_S, 0)] \Pr[\top \leftarrow R_2(\psi_R, s_1, s_2, 0)] \quad (138)$$

$$\times \sum_{s'_2} \Pr[s'_2 \leftarrow S_2(\psi_S, 1)] \Pr[\top \leftarrow R_2(\psi_R, s_1, s'_2, 1)] \quad (139)$$

$$\geq \sum_{(r_1, s_1) \in G_0 \cap G_1} \Pr[r_1 \leftarrow R_1(1^\lambda)] \Pr[s_1 \leftarrow S_1(r_1, 0)] \sum_{s_2} \Pr[s_2 \leftarrow S_2(\psi_S, 0)] \Pr[\top \leftarrow R_2(\psi_R, s_1, s_2, 0)] \quad (140)$$

$$\times \sum_{s'_2} \Pr[s'_2 \leftarrow S_2(\psi_S, 1)] \Pr[\top \leftarrow R_2(\psi_R, s_1, s'_2, 1)] \quad (141)$$

$$\geq (1 - \text{negl}(\lambda)) \left(1 - \frac{1}{p(\lambda)} \right)^2. \quad (142)$$

□

Acknowledgements. TM is supported by JST CREST JPMJCR23I3, JST Moonshot R&D JPMJMS2061-5-1-1, JST FOREST, MEXT QLEAP, the Grant-in Aid for Transformative Research Areas (A) 21H05183, and the Grant-in-Aid for Scientific Research (A) No.22H00522. YS is supported by JST SPRING, Grant Number JPMJSP2110.

References

- [Aar14] Scott Aaronson. The equivalence of sampling and searching. *Theory of Computing Systems*, 55:281–298, 2014. (Cited on page 13.)
- [ABFL16] Scott Aaronson, Adam Bouland, Joseph Fitzsimons, and Mitchell Lee. The space “just above” BQP. In Madhu Sudan, editor, *ITCS 2016*, pages 271–280. ACM, January 2016. (Cited on page 3, 6, 7, 12, 13.)
- [ABK24] Scott Aaronson, Harry Buhrman, and William Kretschmer. A qubit, a coin, and an advice string walk into a relational problem. In Venkatesan Guruswami, editor, *ITCS 2024*, volume 287, pages 1:1–1:24. LIPIcs, January / February 2024. (Cited on page 13.)
- [AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 255–268. ACM Press, June 2020. (Cited on page 5, 26.)
- [AGQY22] Prabhanjan Ananth, Aditya Gulati, Luowen Qian, and Henry Yuen. Pseudorandom (function-like) quantum state generators: New definitions and applications. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 237–265. Springer, Cham, November 2022. (Cited on page 3.)
- [ALY24] Prabhanjan Ananth, Yao-Ting Lin, and Henry Yuen. Pseudorandom strings from pseudorandom quantum states. In Venkatesan Guruswami, editor, *ITCS 2024*, volume 287, pages 6:1–6:22. LIPIcs, January / February 2024. (Cited on page 3.)
- [AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 208–236. Springer, Cham, August 2022. (Cited on page 3.)
- [AS16] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. *SIAM Journal on Computing*, 45(6):2117–2176, 2016. (Cited on page 24.)
- [BBSS23] Amit Behera, Zvika Brakerski, Or Sattath, and Omri Shmueli. Pseudorandomness with proof of destruction and applications. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part IV*, volume 14372 of *LNCS*, pages 125–154. Springer, Cham, November / December 2023. (Cited on page 3.)
- [BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 467–496, Virtual Event, August 2021. Springer, Cham. (Cited on page 3.)

- [BCQ23] Zvika Brakerski, Ran Canetti, and Luowen Qian. On the computational hardness needed for quantum cryptography. In Yael Tauman Kalai, editor, *ITCS 2023*, volume 251, pages 24:1–24:21. LIPIcs, January 2023. (Cited on page 3.)
- [BGH⁺23] Khashayar Barooti, Alex B. Grilo, Loïs Huguenin-Dumittan, Giulio Malavolta, Or Sattath, Quoc-Huy Vu, and Michael Walter. Public-key encryption with quantum keys. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part IV*, volume 14372 of *LNCS*, pages 198–227. Springer, Cham, November / December 2023. (Cited on page 3.)
- [BHKY19] Nir Bitansky, Iftach Haitner, Ilan Komargodski, and Eylon Yogev. Distributional collision resistance beyond one-way functions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 667–695. Springer, Cham, May 2019. (Cited on page 24.)
- [BJ24] Rishabh Batra and Rahul Jain. Commitments are equivalent to statistically-verifiable one-way state generators. In *65th FOCS*, pages 1178–1192. IEEE Computer Society Press, October 2024. (Cited on page 3.)
- [CGG⁺23] Bruno Cavalar, Eli Goldin, Matthew Gray, Peter Hall, Yanyi Liu, and Angelos Pelecanos. On the computational hardness of quantum one-wayness, 2023. (Cited on page 6.)
- [CGG24] Kai-Min Chung, Eli Goldin, and Matthew Gray. On central primitives for quantum cryptography with classical communication. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 215–248. Springer, Cham, August 2024. (Cited on page 3, 7, 11.)
- [CGGH25] Bruno Pasqualotto Cavalar, Eli Goldin, Matthew Gray, and Peter Hall. A meta-complexity characterization of quantum cryptography. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part VII*, volume 15607 of *LNCS*, pages 82–107. Springer, Cham, May 2025. (Cited on page 6, 7.)
- [CKNY24] Jeffrey Champion, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Untelegraphable encryption and its applications, 2024. (Cited on page 5, 27.)
- [DI06] Bella Dubrov and Yuval Ishai. On the randomness complexity of efficient sampling. In Jon M. Kleinberg, editor, *38th ACM STOC*, pages 711–720. ACM Press, May 2006. (Cited on page 5, 24.)
- [GLSV21] Alex B. Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in MiniQCrypt. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 531–561. Springer, Cham, October 2021. (Cited on page 3.)
- [HM25] Taiga Hiroka and Tomoyuki Morimae. Quantum cryptography and meta-complexity. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part II*, volume 16001 of *LNCS*, pages 545–574. Springer, Cham, August 2025. (Cited on page 6, 7.)
- [IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th FOCS*, pages 230–235. IEEE Computer Society Press, October / November 1989. (Cited on page 3.)

- [IL90] Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *31st FOCS*, pages 812–821. IEEE Computer Society Press, October 1990. (Cited on page 7.)
- [JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 126–152. Springer, Cham, August 2018. (Cited on page 3.)
- [KQST23] William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in algorithmica. In Barna Saha and Rocco A. Servedio, editors, *55th ACM STOC*, pages 1589–1602. ACM Press, June 2023. (Cited on page 3.)
- [KQT25] William Kretschmer, Luowen Qian, and Avishay Tal. Quantum-computable one-way functions without one-way functions. In Michal Koucký and Nikhil Bansal, editors, *57th ACM STOC*, pages 189–200. ACM Press, June 2025. (Cited on page 3, 5.)
- [Kre21] W. Kretschmer. Quantum pseudorandomness and classical complexity. *TQC 2021*, 2021. (Cited on page 3.)
- [KT24] Dakshita Khurana and Kabir Tomer. Commitments from quantum one-wayness. In Bojan Mohar, Igor Shinkar, and Ryan O’Donnell, editors, *56th ACM STOC*, pages 968–978. ACM Press, June 2024. (Cited on page 3, 10.)
- [KT25] Dakshita Khurana and Kabir Tomer. Founding quantum cryptography on quantum advantage, or, towards cryptography from #p hardness. In Michal Koucký and Nikhil Bansal, editors, *57th ACM STOC*, pages 178–188. ACM Press, June 2025. (Cited on page 6, 7.)
- [KY18] Ilan Komargodski and Eylon Yogev. On distributional collision resistant hashing. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 303–327. Springer, Cham, August 2018. (Cited on page 5, 6, 24.)
- [LMW24] Alex Lombardi, Fermi Ma, and John Wright. A one-query lower bound for unitary synthesis and breaking quantum cryptography. In Bojan Mohar, Igor Shinkar, and Ryan O’Donnell, editors, *56th ACM STOC*, pages 979–990. ACM Press, June 2024. (Cited on page 3.)
- [MH25] Fermi Ma and Hsin-Yuan Huang. How to construct random unitaries. In Michal Koucký and Nikhil Bansal, editors, *57th ACM STOC*, pages 806–809. ACM Press, June 2025. (Cited on page 3.)
- [MPY23] Tomoyuki Morimae, Alexander Poremba, and Takashi Yamakawa. Revocable quantum digital signatures. *Cryptology ePrint Archive*, Paper 2023/1937, 2023. (Cited on page 5, 27.)
- [MX24] Tomoyuki Morimae and Keita Xagawa. Quantum group actions. *Cryptology ePrint Archive*, Paper 2024/1578, 2024. (Cited on page 3.)
- [MY22] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 269–295. Springer, Cham, August 2022. (Cited on page 3.)

- [MY24] Tomoyuki Morimae and Takashi Yamakawa. One-Wayness in Quantum Cryptography. In Frédéric Magniez and Alex Bredariol Grilo, editors, *19th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2024)*, volume 310 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:21, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. (Cited on page 3.)
- [MY24] Tomoyuki Morimae, Shogo Yamada, and Takashi Yamakawa. Quantum unpredictability. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IX*, volume 15492 of *LNCS*, pages 3–32. Springer, Singapore, December 2024. (Cited on page 3.)
- [Ost91] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. *Computational Complexity Conference*, 1991. (Cited on page 6, 7.)
- [OW93] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *The 2nd Israel Symposium on Theory and Computing Systems. 1993*, pp.3-17, 1993. (Cited on page 6, 7.)
- [Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 334–345. Springer, Berlin, Heidelberg, May / June 1998. (Cited on page 24.)
- [Yan22] Jun Yan. General properties of quantum bit commitments (extended abstract). In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 628–657. Springer, Cham, December 2022. (Cited on page 3.)