Quantum Advantage from One-Way Functions

Tomoyuki Morimae[1] and Takashi Yamakawa[1,2]

[1]Yukawa Institute for Theoretical Physics, Kyoto University, Kyoto, Japan
tomoyuki.morimae@yukawa.kyoto-u.ac.jp
[2]NTT Social Informatics Laboratories, Tokyo, Japan
takashi.yamakawa.ga@hco.ntt.co.jp

Abstract

Showing quantum advantage based on weaker and standard classical complexity assumptions is one of the most important goals in quantum information science. In this paper, we demonstrate quantum advantage with several basic assumptions, specifically based on only the existence of classically-secure one-way functions. We introduce *inefficient-verifier proofs of quantumness* (IV-PoQ), and construct it from statistically-hiding and computationally-binding classical bit commitments. IV-PoQ is an interactive protocol between a verifier and a quantum polynomial-time prover consisting of two phases. In the first phase, the verifier is classical probabilistic polynomial-time, and it interacts with the quantum polynomial-time prover over a classical channel. In the second phase, the verifier becomes inefficient, and makes its decision based on the transcript of the first phase. If the quantum prover is honest, the inefficient verifier accepts with high probability, but any classical probabilistic polynomial-time malicious prover only has a small probability of being accepted by the inefficient verifier. In our construction, the inefficient verifier can be a classical deterministic polynomial-time algorithm that queries an $\mathbf{NP}$ oracle. Our construction demonstrates the following results based on the known constructions of statistically-hiding and computationally-binding commitments from one-way functions or distributional collision-resistant hash functions:

- If one-way functions exist, then IV-PoQ exist.
- If distributional collision-resistant hash functions exist (which exist if hard-on-average problems in $\mathbf{SZK}$ exist), then constant-round IV-PoQ exist.

We also demonstrate quantum advantage based on worst-case-hard assumptions. We define *auxiliary-input IV-PoQ* (AI-IV-PoQ) that only require that for any malicious prover, there exist infinitely many auxiliary inputs under which the prover cannot cheat. We construct AI-IV-PoQ from an auxiliary-input version of commitments in a similar way, showing that

- If auxiliary-input one-way functions exist (which exist if $\mathbf{CZK} \not\subseteq \mathbf{BPP}$), then AI-IV-PoQ exist.
- If auxiliary-input collision-resistant hash functions exist (which is equivalent to $\mathbf{PWPP} \not\subseteq \mathbf{FBPP}$) or $\mathbf{SZK} \not\subseteq \mathbf{BPP}$, then constant-round AI-IV-PoQ exist.

Finally, we also show that some variants of PoQ can be constructed from quantum-evaluation one-way functions (QE-OWFs), which are similar to classically-secure classical one-way functions except that the evaluation algorithm is not classical but quantum. QE-OWFs appear to be weaker than classically-secure classical one-way functions.

Contents

1 Introduction

Quantum advantage means that quantum computing outperforms classical one for some computational tasks. Showing quantum advantage based on weaker and standard classical complexity assumptions is one of the most important goals in quantum information science.

One approach to demonstrate quantum advantage is the sampling-based one. In the sampling-based quantum advantage, quantum polynomial-time (QPT) algorithms can sample certain probability distributions but no classical probabilistic polynomial-time (PPT) algorithm can. A great merit of the approach is that relatively simple quantum computing models are enough, such as the Boson Sampling model [AA11], the IQP model [BJS11], the random circuit model [BFNV19], and the one-clean-qubit model [FKM+18]. [1] Output probability distributions of these restricted quantum computing models cannot be sampled by any PPT algorithm within a constant multiplicative error[2] unless the polynomial-time hierarchy collapses to the third [AA11, BJS11] or the second level [FKM+18].[3] The assumption that the polynomial-time hierarchy does not collapse is a widely-believed assumption in classical complexity theory, but one disadvantage of these results is that the multiplicative-error sampling is unrealistic. The requirement of the multiplicative-error sampling can be relaxed to that of the constant additive-error sampling [AA11, BMS16, Mor17, BFNV19],[4] but the trade-off is that the underlying classical complexity assumptions become less standard: some ad-hoc assumptions about average-case $\#\mathbf{P}$-hardness of some problems, which were not studied before, have to be introduced.

Another disadvantage of the sampling-based approach is that it is not known to be verifiable. For the multiplicative-error case, we do not know how to verify quantum advantage even with a computationally-unbounded verifier. Also for the additive-error case, we do not know how to verify the quantum advantage efficiently. (For example, there is a negative result that suggets that exponentially-many samples are necessary to verify the correctness of the sampling [HKEG19].) At least, we can say that if there exists a sampling-based quantum advantage in the additive-error case, there exists an inefficiently-verifiable quantum advantage for a certain search problem [Aar14].[5]

Some inefficiently-verifiable search problems that exhibit quantum advantage have been introduced. For example, for the random circuit model, [AC17, AG19] introduced so-called Heavy Output Generation (HOG) and Linear Cross-Entropy Heavy Output Generation (XHOG) where given a quantum circuit C it is required to output bit strings that satisfy certain relations about C. The relations can be verified inefficiently. The classical hardnesses of these problems are, however, based on new assumptions introduced by the authors. [Aar10] constructed an inefficiently-verifiable search problem (Fourier Fishing), but its quantum advantage is relative to random oracles. [ACC+22] constructed another inefficiently-verifiable search problem (Collision Hashing), but its quantum advantage is also relative to random oracles.

There is another approach of demonstrating quantum advantage where the verification is efficient, namely, proofs of quantumness (PoQ) [BCM+21]. In PoQ, we have a QPT prover and a PPT verifier. They interact over a classical channel, and the verifier finally makes the decision. If the QPT prover behaves honestly, the verifier accepts with high probability, but for any malicious PPT prover, the verifier accepts with only small probability. The simplest way of realizing PoQ is to let the prover solve an $\mathbf{NP}$ problem that is quantumly easy but classically hard, such as factoring [Sho94]. Such a simplest way is, however, based on specific assumptions that certain specific problems are hard for PPT algorithms.

The first construction of PoQ based on a general assumption was given in [BCM+21] where (noisy) trapdoor claw-free functions with the adaptive-hardcore-bit property[6] is assumed. Such functions can be instantiated with

[1] The Boson Sampling model is a quantum computing model that uses non-interacting bosons, such as photons. The IQP (Instantaneous Quantum Polytime) model is a quantum computing model where only commuting quantum gates are used. The random circuit model is a quantum computing model where each gate is randomly chosen. The one-clean-qubit model is a quantum computing model where the input is $|0\rangle\langle 0| \otimes \frac{I^{\otimes m}}{2^m}$.

[2] We say that the output probability distribution of a quantum algorithm is sampled by a classical algorithm within a constant multiplicative error ϵ if $|q_z - p_z| \le \epsilon p_z$ is satisfied for all z, where q_z is the probability that the quantum algorithm outputs the bit string z, and p_z is the probability that the classical algorithm outputs the bit string z.

[3] [TD04] previously showed that output probability distributions of constant-depth quantum circuits cannot be sampled classically unless $\mathbf{BQP} \subseteq \mathbf{AM}$. Their assumption can be easily improved to the assumption that the polynomial-time hierarchy does not collapse to the second level.

[4] We say that the output probability distribution of a quantum algorithm is sampled by a classical algorithm within a constant additive error ϵ if $\sum_z |q_z - p_z| \le \epsilon$ is satisfied, where q_z is the probability that the quantum algorithm outputs the bit string z, and p_z is the probability that the classical algorithm outputs the bit string z.

[5] [Aar14, Theorem 21] showed that if there exists an additive-error sampling problem that is quantumly easy but classically hard, then there exists a search problem that is quantumly easy but classically hard. The relation of the search problem is verified inefficiently. Note that the search problem depends on the time-complexity of the classical adversary, and therefore it is incomparable to our (AI-)IV-PoQ.

[6] The adaptive-hardcore-bit property very roughly means that it is hard to find x_b ($b \in \{0,1\}$) and $d \neq \mathbf{0}$ such that $f_0(x_0) = f_1(x_1)$ and $d \cdot (x_0 \oplus x_1) = 0$, given a claw-free pair (f_0, f_1).

the LWE assumption, for example [BCM+21]. The adaptive-hardcore-bit property was removed in [KMCVY22], where only trapdoor 2-to-1 collision-resistant hash functions are assumed. In [MY23], PoQ was constructed from (full-domain) trapdoor permutations. PoQ can also be constructed from quantum homomorphic encryptions (QHE) [KLVY22] for a certain class of quantum operations (such as controlled-Hadamard gates), which can be instantiated with the LWE assumption [Mah18]. These contructions are interactive, i.e., the verifier and the prover have to exchange many rounds of messages. Recently, a non-interactive PoQ has been realized with only random oracles [YZ22]. This result demonstrates efficiently-verifiable quantum advantage with an "unstructured" problem for the first time. However, it is known that hardness relative to a random oracle does not necessarily imply hardness in the unrelativized world where the random oracle is replaced with a real-world hash function [CGH04]. Thus, [YZ22] does not give quantum advantage under a standard assumption in the unrelativized world.

We therefore have the following open problem.

Can we construct PoQ from weaker and standard assumptions, such as the existence of one-way functions (OWFs)?

Note that this open problem is highly non-trivial even if we give up the efficient verification. As we have explained, all previous results on inefficiently-verifiable quantum advantage assume (random) oracles or some ad-hoc assumptions newly introduced by the authors themselves. It is therefore highly non-trivial to answer even the following question.

Can we demonstrate inefficiently-verifiable quantum advantage with weaker and standard assumptions, such as the existence of OWFs?

1.1 Our Results

In this paper, we answer the second question affirmatively. We demonstrate inefficiently-verifiable quantum advantage with several basic assumptions, specifically based on only the existence of OWFs. To our knowledge, this is the first time that quantum advantage is shown based only on OWFs. More precisely, we construct what we call *inefficient-verifier proofs of quantumness* (IV-PoQ) from statistically-hiding and computationally-binding classical bit commitments. IV-PoQ is an interactive protocol between a verifier and a QPT prover, which is divided into two phases. In the first phase, the verifier is PPT, and it interacts with the QPT prover over a classical channel. In the second phase, the verifier becomes inefficient, and makes the decision based on the transcript of the first phase.[7] If the QPT prover is honest, the inefficient verifier accepts with high probability, but for any PPT malicious prover, the inefficient verifier accepts with only small probability. The new notion of IV-PoQ captures both the standard PoQ and inefficiently-verifiable quantum advantage (including search problems that exhibit quantum advantage).

Our main result is the following:

Theorem 1.1. *$(k+6)$-round IV-PoQ exist if statistically-hiding and computationally-binding classical bit commitments with k-round commit phase exist.*

A proof of Theorem 1.1 is given in Section 6. Note that we actually need the statistical hiding property only for the honest receiver, because the receiver corresponds to the verifier. Moreover, note that in our construction, the inefficient verifier in the second phase is enough to be a classical deterministic polynomial-time algorithm that queries the $\mathbf{NP}$ oracle. (See Section 6.3.)

Because statistically-hiding and computationally-binding classical bit commitments can be constructed from OWFs [HNO+09], we have the following result.

Theorem 1.2. *IV-PoQ exist if OWFs exist.*

Moreover, it is known that constant-round statistically-hiding and computationally-binding bit commitments can be constructed from distributional collision resistant hash functions [BHKY19][8], which exist if there is an hard-on-average problem in $\mathbf{SZK}$ [KY18]. Therefore we also have the following result.[9]

[7]The inefficient verifier could also take the efficient verifier's secret information as input in addition to the transcript. However, without loss of generality, we can assume that the inefficient verifier takes only the transcript as input, because we can always modify the protocol of the first phase in such a way that the efficient verifier sends its secret information to the prover at the end of the first phase.

[8]A distributional collision-resistant hash function [DI06] is a weaker variant of a collision-resistant hash function that requires the hardness of sampling a collision (x, y) where x is uniformly random and y is uniformly random conditioned on colliding with x.

[9]It is also known that constant-round statistically-hiding and computationally-binding commitments can be constructed from multi-collision resistant hash functions [BDRV18, KNY18], and therefore we have constant-round IV-PoQ from multi-collision resistant hash functions as well.

Theorem 1.3. *Constant-round IV-PoQ exist if there exist distributional collision-resistant hash functions, which exist if there is an hard-on-average problem in* $\mathbf{SZK}$.

The assumptions in Theorems 1.2 and 1.3 are average-case-hard assumptions. We can further weaken the assumptions to worst-case-hard ones if we require only worst-case soundness for IV-PoQ. Namely, we define *auxiliary-input IV-PoQ* (AI-IV-PoQ) that only requires that for any malicious prover, there exist infinitely many auxiliary inputs under which the prover cannot cheat. We can show the following:

Theorem 1.4. $(k+6)$*-round AI-IV-PoQ exist if auxiliary-input statistically-hiding and computationally-binding classical bit commitments with* k*-round commit phase exist.*

Its proof is omitted because it is similar to that of Theorem 1.1. Although AI-IV-PoQ is weaker than IV-PoQ, we believe that it still demonstrates a meaningful notion of quantum advantage, because it shows "worst-case quantum advantage" in the sense that no PPT algorithm can simulate the QPT honest prover on all auxiliary inputs.

Auxiliary-input OWFs[10] exist if $\mathbf{CZK} \not\subseteq \mathbf{BPP}$ [OW93].[11] Moreover, the construction of statistically-hiding and computationally-binding commitments from OWFs in [HNO+09] can be modified for the auxiliary-input setting. We therefore have the following result.

Theorem 1.5. *AI-IV-PoQ exist if there exist auxiliary-input OWFs, which exist if* $\mathbf{CZK} \not\subseteq \mathbf{BPP}$.

Furthermore, relying on the known constructions of constant-round (auxiliary-input) statistically-hiding commitments [HM96, OV08], we obtain the following result.

Theorem 1.6. *Constant-round AI-IV-PoQ exist if auxiliary-input collision-resistant hash functions exist (which is equivalent to* $\mathbf{PWPP} \not\subseteq \mathbf{FBPP}$*)*[12] *or* $\mathbf{SZK} \not\subseteq \mathbf{BPP}$.

Finally, we can also define another variant of IV-PoQ that we call *infinitely-often IV-PoQ* (IO-IV-PoQ) where the soundness is satisfied for infinitely many values of the security parameter. We note that IO-IV-PoQ lie between IV-PoQ and AI-IV-PoQ. It is known that infinitely-often OWFs exist if $\mathbf{SRE} \not\subseteq \mathbf{BPP}$ [AR16].[13] Therefore we also have the following result.

Theorem 1.7. *IO-IV-PoQ exist if infinitely-often OWFs exist, which exist if* $\mathbf{SRE} \not\subseteq \mathbf{BPP}$.

A comparison table among existing and our results on quantum advantage can be found in Table 1.

Remarks on completeness-soundness gap. We remark that the above theorems consider (AI-/IO-) IV-PoQ that only have an inverse-polynomial completeness-soundness gap, i.e., the honest QPT prover passes verification with probability at least c and any PPT cheating prover passes verification with probability at most s where $c - s \geq 1/\mathrm{poly}(\lambda)$ for the security parameter λ. Due to the inefficiency of verification, it is unclear if we can generically amplify the gap *even by sequential repetition*.[14] Fortunately, we find a stronger definition of soundness called strong soundness which our constructions satisfy and enables us to amplify the gap by sequential repetition. Roughly speaking, strong soundness requires that soundness holds for almost all fixed cheating prover's randomness rather than on average. See Definition 4.8 for the formal definition. This enables us to amplify the completeness-soundness gap to be optimal for any of our constructions. However, we remark that this increases the round complexity and in particular, the schemes of Theorems 1.3 and 1.6 are no longer constant-round if we amplify the completeness-soundness gap. This issue could be resolved if we could prove that parallel repetition amplifies the gap, but we do not know how to prove this. Remark

[10]Roughly speaking, auxiliary-input OWFs are keyed functions such that for each adversary there exist infinitely many keys on which the adversary fails to invert the function.

[11]$\mathbf{CZK}$ is the class of promise problems that have computational zero-knowledge proofs. By abuse of notation, we write $\mathbf{BPP}$ to mean the class of promise problems (instead of languages) that are decidable in PPT.

[12]See Appendix B.1 for the definitions of $\mathbf{PWPP}$ and $\mathbf{FBPP}$.

[13]$\mathbf{SRE}$ is the class of problems that admit statistically-private randomized encoding with polynomial-time client and computationally-unbounded server.

[14]If we assume soundness against *non-uniform* PPT adversaries, then it is easy to show that sequential repetition generically amplifies the gap. However, we consider the uniform model of adversaries in this paper since otherwise we would need non-uniform assumptions like one-way functions against non-uniform adversaries which is stronger than the mere existence of one-way functions against uniform adversaries.

Table 1: Comparison among results on quantum advantage. In column "Verification", "No" means that the verification is not known to be possible. (Actually, it seems to be impossible.) In column "Assumption", PH stands for the polynomial-time hierarchy, seOWFs stands for subexponentially secure one-way functions, 2-1 TDCRHFs stands for 2-to-1 trapdoor collision-resistant hash functions, QHE stands for quantum homomorphic encryption, fdTDPs stands for full-domain trapdoor permutations, OWFs stands for one-way functions, dCRHFs stands for distributional collision-resistant hash functions, and CRHFs stands for collision-resistant hash functions. In column "Misc", Mult.err. and Add.err. stand for multiplicative and additive errors, respectively. In the row of [Sho94], the number of rounds is two, because the verifier sends a composite number to the prover, and the prover returns its factorization. It can be considered as a non-interactive if the composite number is given as an auxiliary input.

Ref.	Verification	#Rounds	Assumption	Misc
[TD04, AA11, BJS11, FKM+18]	No	1	PH does not collapse	Mult.err. sampling
[AA11, BMS16, BFNV19, Mor17]	No	1	Ad hoc	Add.err. sampling
[AA15]	No	1	Random oracle	Fourier Sampling
[AC17]	No	1	seOWFs+$\mathbf{P}/\mathbf{poly}$-oracle	Fourier Sampling
[AC17, AG19]	Inefficient	1	Ad hoc	HOG, XHOG
[Aar10]	Inefficient	1	Random oracle	Fourier Fishing
[AC17]	Inefficient	1	seOWFs+$\mathbf{P}/\mathbf{poly}$-oracle	Fourier Fishing
[ACC+22]	Inefficient	1	Random oracle	Collision Hashing
[Sho94]	Efficient	2	Factoring/Discrete-log	
[YZ22]	Efficient	1	Random oracle	
[BCM+21, KMCVY22]	Efficient	$O(1)$	(Noisy) 2-1 TDCRHFs	
[KLVY22]	Efficient	$O(1)$	QHE	
[MY23]	Efficient	$\mathrm{poly}(\lambda)$	fdTDPs	
Theorem 1.2	Inefficient	$\mathrm{poly}(\lambda)$	OWFs	
Theorem 1.3	Inefficient	$O(1)$	dCRHFs	
Theorem 1.5	Inefficient	$\mathrm{poly}(\lambda)$	Auxiliary-input OWFs / $\mathbf{CZK} \not\subseteq \mathbf{BPP}$	AI-IV-PoQ
Theorem 1.6	Inefficient	$O(1)$	Auxiliary-input CRHFs / $\mathbf{SZK} \not\subseteq \mathbf{BPP}$	AI-IV-PoQ
Theorem 1.7	Inefficient	$\mathrm{poly}(\lambda)$	Infinitely-often OWFs / $\mathbf{SRE} \not\subseteq \mathbf{BPP}$	IO-IV-PoQ

that we cannot use existing parallel repetition theorems for interactive arguments because verification is inefficient. Indeed, it is observed in [CHS05] that parallel repetition may not amplify the gap when verification is inefficient even for two-round arguments. Thus, we believe that it is very challenging or even impossible to prove a general parallel repetition theorem for (AI-/IO-)IV-PoQ. Nonetheless, it may be still possible to prove a parallel repetition theorem for our particular constructions, which we leave as an interesting open problem.

Implausibility of two-round AI-IV-PoQ. It is natural to ask how many rounds of interaction are needed. As already mentioned, it is trivial to construct two-round PoQ if we assume the existence of classically-hard and quantumly-easy problems such as factoring. We show evidence that it is inevitable to rely on such an assumption for constructing two-round (AI-/IO-)IV-PoQ. In the following, we state theorems for AI-IV-PoQ, but they immediately imply similar results for IV-PoQ and IO-IV-PoQ because they are stronger than AI-IV-PoQ.

First, we prove that there is no classical black-box reduction from security of two-round AI-IV-PoQ to standard cryptographic assumptions unless the assumptions do not hold against QPT adversaries.

Theorem 1.8 (Informal). *For a two-round AI-IV-PoQ, if its soundness can be reduced to a game-based assumption by a classical black-box reduction, then the assumption does not hold against QPT adversaries.*

The formal version of the theorem is given in Theorem 7.5. Here, game-based assumptions are those formalized as a game between the adversary and challenger that include (but not limited to) general assumptions such as security of OWFs, public key encryption, digital signatures, oblivious transfers, indistinguishability obfuscation, succinct arguments etc. as well as concrete assumptions such as the hardness of factoring, discrete-logarithm, LWE etc.[15] See Definition 7.1 for a formal definition. In particular, since we believe that quantumly-secure OWFs exist, the above theorem can be interpreted as a negative result on constructing two-round AI-IV-PoQ from general OWFs.

The proof idea is quite simple: Suppose that there is a classical black-box reduction algorithm R that is given a malicious prover as an oracle and breaks an assumption. Intuitively, the reduction should still work even if it is given the honest quantum prover $\mathcal{P}$ as an oracle. By considering the combination of R and $\mathcal{P}$ as a single quantum adversary, the assumption is broken. We remark this can be seen as an extension of an informal argument in [BKVV20] where they argue that it is unlikely that a two-round PoQ can be constructed from the hardness of the LWE problem.[16]

Note that Theorem 1.8 only rules out classical reductions. One may think that the above argument extends to rule out quantum reductions, but there is some technical difficulty. Roughly speaking, the problem is that a coherent execution of the honest quantum prover may generate an entanglement between its message register and internal register unlike a coherent execution of a classical cheating prover (see Remark 7.6 for more explanations).[17] To complement this, we prove another negative result that also captures some class of quantum reductions.

Theorem 1.9 (Informal). *If a cryptographic primitive* $\mathtt{P}$ *has a quantumly-secure construction (possibly relative to a classical oracle), then there is a randomized classical oracle relative to which two-round AI-IV-PoQ do not exist but a quantumly-secure construction of* $\mathtt{P}$ *exists.*

The formal version of the theorem is given in Theorem 7.13. The above theorem can be interpreted as a negative evidence on constructing two-round IV-PoQ from a cryptographic primitive for which we believe that quantumly-secure constructions exist (e.g., OWFs, public key encryption, indistinguishability obfuscation etc.) In particular, the above theorem rules out any constructions that work relative to randomized classical oracles.[18] Theorem 1.9 is incomparable to Theorem 1.8 since Theorem 1.9 does not require the reduction to be classical unlike Theorem 1.8, but requires that the construction and reduction work relative to randomized classical oracles.

Again, the proof idea is simple. Suppose that a quantumly-secure construction f of a primitive $\mathtt{P}$ exists relative to an oracle O. Then we introduce an additional oracle Q^O that takes a description of a quantum circuit C^O with O-gates and its input x as input and outputs a classical string according to the distribution of $C^O(x)$. Relative to oracles (O, Q^O), there do not exist AI-IV-PoQ since a classical malicious prover can query the description of the honest quantum prover to Q^O to get a response that passes the verification with high probability. On the other hand, f is quantumly-secure relative to (O, Q^O) since we assume that it is quantumly-secure relative to O and the additional oracle Q^O is useless for quantum adversaries since they can simulate it by themselves.

We remark that the above theorems do not completely rule out black-box constructions of two-round AI-IV-PoQ from quantumly-hard assumptions. For example, consider a quantum black-box reduction that queries a cheating prover with a fixed randomness multiple times. Such a reduction is not captured by Theorem 1.8 because it is quantum. Moreover, it is not captured by Theorem 1.9 because it does not work relative to randomized classical oracles since we cannot fix the randomness of the randomized classical oracle. It is a very interesting open problem to study if such a reduction is possible.

Quantum advantage based on quantum primitives weaker than OWFs. The existence of OWFs is the most fundamental assumption in classical cryptography. Interestingly, it has been realized recently that it is not necessarily the case in quantum cryptography [JLS18, Kre21, MY22b, AQY22, BCQ23, AGQY22, CX22, MY22a, KQST22]. Many quantum cryptographic tasks can be realized with new quantum primitives, which seem to be weaker than OWFs, such as pseudorandom states generators [JLS18], one-way states generators [MY22b], and EFI [BCQ23]. Can we construct PoQ (or its variants) from quantum primitives that seem to be weaker than OWFs? We show that variants of PoQ

[15] This is similar to falsifiable assumptions [Nao03, GW11] but there is an important difference that we do not restrict the challenger to be efficient.

[16] They use one-round PoQ to mean what we call two-round PoQ by counting interaction from the verifier to prover and from the prover to verifier as a single round.

[17] This observation is due to Mark Zhandry.

[18] Note that reductions that work relative to *deterministic* classical oracles do not necessarily work relative to *randomized* classical oracles [Aar08, Section 5].

can be constructed from (classically-secure) quantum-evaluation OWFs (QE-OWFs). QE-OWFs is the same as the standard classically-secure classical OWFs except that the function evaluation algorithm is not deterministic classical polynomial-time but quantum polynomial-time. (Its definition is given in Section 8.) QE-OWFs seem to be weaker than classically-secure classical OWFs. (For example, consider the function f that on input (x, y) outputs $\Pi_L(x)\|g(y)$, where L is any language in $\mathbf{BQP} \setminus \mathbf{BPP}$, Π_L is a function such that $\Pi_L(x) = 1$ if $x \in L$ and $\Pi_L(x) = 0$ if $x \notin L$, and g is any classically-secure classical OWF. f is a QE-OWF, and f cannot be evaluated in classical polynomial-time if $\mathbf{BQP} \neq \mathbf{BPP}$. For details, see Section 8.) We show the following result.

Theorem 1.10. *If QE-OWFs exist, then quantum-verifier PoQ (QV-PoQ) exist or infinitely-often classically-secure classical OWFs exist.*

A proof of the theorem is given in Section 8. QV-PoQ is the same as PoQ except that the verifier is a QPT algorithm. Such a new notion of PoQ will be useful, for example, when many local quantum computers are connected over classical internet: A quantum local machine may want to check whether it is interacting with a quantum computer or not over a classical channel.

The proof idea of Theorem 1.10 is as follows. Let f be a QE-OWF. We construct QV-PoQ as follows: The verifier first chooses $x \leftarrow \{0,1\}^n$ and sends it to the prover. The prover then returns y. The verifier finally evaluates $f(x)$ by himself, and accepts if it is equal to y. If the soundness holds, we have QV-PoQ. On the other hand, if the soundness does not hold, then it means that f can be evaluated in PPT, which means that f is a classical OWF. It is an interesting open problem whether PoQ or its variants can be constructed from pseudorandom quantum states generators, one-way states generators, or EFI.

Infinitely-often classically-secure OWFs imply IO-IV-PoQ (Theorem 1.7), and therefore Theorem 1.10 shows that the existence of QE-OWFs anyway implies quantum advantage (i.e., QV-PoQ or IO-IV-PoQ). Moreover, QV-PoQ in Theorem 1.10 implies IV-PoQ (and therefore IO-IV-PoQ). (In general, QV-PoQ does not necessarily imply IV-PoQ, but in our case, it does because our construction of QV-PoQ is a two-round protocol with the verifier's first message being a uniformly-randomly-chosen classical bit string.) Hence we have the result that QE-OWFs implies IO-IV-PoQ in either case.

1.2 Technical Overview

In this subsection, we provide technical overview of our main result, Theorem 1.1, namely, the construction of IV-PoQ from statistically-hiding commitments. (The construction of AI-IV-PoQ is similar.) Our construction is based on PoQ of [KMCVY22]. Let us first review their protocol. Their protocol can be divided into two phases. In the first phase, the verifier first generates a pair of a trapdoor and a trapdoor 2-to-1 collision resistant hash function F. The verifier sends F to the prover. The prover generates the quantum state $\sum_{x\in\{0,1\}^\ell} |x\rangle |F(x)\rangle$, and measures the second register in the computational basis to obtain the measurement result y. The post-measurement state is $|x_0\rangle + |x_1\rangle$, where $F(x_0) = F(x_1) = y$. This is the end of the first phase.

In the second phase, the verifier chooses a challenge bit $c \in \{0,1\}$ uniformly at random. If $c = 0$, the verifier asks the prover to measure the state in the computational basis. The verifier accepts and halts if the prover's measurement result is x_0 or x_1. (The verifier can compute x_0 and x_1 from y, because it has the trapdoor.) The verifier rejects and halts if the prover's measurement result is not correct. If $c = 1$, the verifier sends the prover a bit string $\xi \in \{0,1\}^\ell$ which is chosen uniformly at random. The prover changes the state $|x_0\rangle + |x_1\rangle$ into the state $|\xi \cdot x_0\rangle |x_0\rangle + |\xi \cdot x_1\rangle |x_1\rangle$, and measures the second register in the Hadamard basis. If the measurement result is $d \in \{0,1\}^\ell$, the post-measurement state is $|\xi \cdot x_0\rangle + (-1)^{d\cdot(x_0\oplus x_1)} |\xi \cdot x_1\rangle$, which is one of the BB84 states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. The verifier then asks the prover to measure this single-qubit state in a certain basis, and accepts if the measurement result is appropriate. This is the end of the second phase. Intuitively, the soundness comes from the collision resistance of F: If a malicious PPT prover is accepted by the verifier with some high probability for both challenges, $c = 0$ and $c = 1$, we can construct a PPT adversary that can find both x_0 and x_1 with non-negligible probability, which contradicts the collision resistance.

Therefore, once we can construct an interactive protocol where a verifier can let a prover generate $|x_0\rangle + |x_1\rangle$ in such a way that no malicious PPT prover can learn both x_0 and x_1, we can construct PoQ by running the second phase of [KMCVY22] on it. Can we do that with only OWFs? Our key idea is to *coherently* execute statistically-hiding classical bit commitments, which can be constructed from OWFs [HNO+09]. (A similar idea was also used in [MY23].) The prover plays the role of the sender of the commitment scheme, and the verifier plays the role of the receiver of the commitment scheme. The prover first generates the state $\sum_{b\in\{0,1\}}\sum_{x\in\{0,1\}^\ell}|b\rangle|x\rangle$, which is the superposition of the bit $b\in\{0,1\}$ to commit and sender's random seed $x\in\{0,1\}^\ell$. The prover and the verifier then run the interactive commitment phase. When the prover computes its message, it coherently computes the message on its state, and measures a register to obtain the measurement result.[19] The prover sends the measurement result as the sender's message to the verifier. The verifier runs classical receiver's algorithm, and sends classical message to the prover. At the end of the commit phase, the honest prover possesses the state

$$|0\rangle\sum_{x\in X_{0,t}}|x\rangle+|1\rangle\sum_{x\in X_{1,t}}|x\rangle, \tag{1}$$

where $X_{b,t}$ is the set of sender's random seeds that are consistent with the committed bit b and the transcript t, which is the sequence of all classical messages exchanged between the prover and the verifier.

If $|X_{0,t}|=|X_{1,t}|=1$, Equation (1) is $|0\rangle|x_0\rangle+|1\rangle|x_1\rangle$, where x_b is the unique element of $X_{b,t}$ for each $b\in\{0,1\}$. In that case, we can run the second phase of [KMCVY22] on it.[20] However, in general, $|X_{0,t}|=|X_{1,t}|=1$ is not always satisfied, and if it is not satisfied, we do not know how to realize PoQ from the state of Equation (1). This is our first problem. Moreover, even if $|X_{0,t}|=|X_{1,t}|=1$ is satisfied, we have the second problem: The efficient verifier cannot compute (x_0,x_1), because there is no trapdoor. The efficient verifier therefore cannot check whether the prover passes the tests or not.[21]

Unfortunately, we do not know how to solve the second problem, and therefore we have to give up the efficient verification. On the other hand, we can solve the first problem by introducing a new hashing technique, which may have further applications. First, we notice that $|X_{0,t}|\simeq|X_{1,t}|$ with overwhelming probability, because otherwise the statistical-hiding of the classical bit commitment scheme is broken. Next, let $\mathcal{H}:=\{h:\mathcal{X}\to\mathcal{Y}\}$ be a pairwise-independent hash family with $\mathcal{X}=\{0,1\}^\ell$. The verifier chooses $h_0,h_1\in\mathcal{H}$ uniformly at random, and sends (h_0,h_1) to the prover. The prover changes the state of Equation (1) into

$$|0\rangle\sum_{x\in X_{0,t}}|x\rangle|h_0(x)\rangle+|1\rangle\sum_{x\in X_{1,t}}|x\rangle|h_1(x)\rangle, \tag{2}$$

and measures the third register in the computational-basis to obtain the measurement result y. We show that if $|\mathcal{Y}|$ is chosen so that $|\mathcal{Y}|\simeq 2|X_{b,t}|$, the state collapses by the measurement to $|0\rangle|x_0\rangle+|1\rangle|x_1\rangle$ with constant probability, where $x_b\in X_{b,t}\cap h_b^{-1}(y)$ for $b\in\{0,1\}$. The remaining problem is that the efficient verifier cannot compute $|X_{b,t}|$, and therefore it cannot find the appropriate $|\mathcal{Y}|$. This problem is solved by noticing that even if the verifier chooses $|\mathcal{Y}|$ randomly, it is $\simeq 2|X_{b,t}|$ with non-negligible probability. More precisely, let m be an integer such that $(1+\epsilon)^m\ge 2^{\ell+1}$, where $0<\epsilon<1$ is a small constant (which we take $\epsilon=1/100$, for example). Then, we show that there exists a $j^*\in\{0,1,...,m-1\}$ such that $\lceil(1+\epsilon)^{j^*}\rceil\simeq 2|X_{b,t}|$. Therefore, if the efficient verifier chooses $j\in\{0,1,...,m-1\}$ uniformly at random, and sets $\mathcal{Y}:=[\lceil(1+\epsilon)^j\rceil]$, then $|\mathcal{Y}|\simeq 2|X_{b,t}|$ is satisfied with probability $1/m=1/\mathrm{poly}(\lambda)$.

In summary, the efficient verifier can let the honest prover generate $|0\rangle|x_0\rangle+|1\rangle|x_1\rangle$ with non-negligible probability. Fortunately, the second phase of [KMCVY22] is a public coin one, which means that all messages from the verifier are uniformly-chosen random bit strings, and therefore our efficient verifier can send all its messages without doing any inefficient computation (such as finding an element of $X_{b,t}\cap h_b^{-1}(y)$, etc.). All verifications are later done by the inefficient verifier.

The soundness of our construction is shown from the computational-binding of the classical bit commitment scheme. In the soundness proof of [KMCVY22], they use the fact that no PPT malicious prover can find both x_0 and x_1, which comes from the collision resistance. In our case, we have that property from the computational-binding of the classical

[19]For example, in the prover's jth round, if the prover possesses a state $\sum_{b\in\{0,1\}}\sum_{x\in X_b}|b\rangle|x\rangle$, where X_b is a certain set, it changes the state into $\sum_{b\in\{0,1\}}\sum_{x\in X_b}|b\rangle|x\rangle|f_j(b,x,t_j)\rangle$, and measures the third register to obtain the measurement result α_j, where f_j is the function that computes sender's jth message, and t_j is the transcript obtained before the jth round. The prover sends α_j to the verifier as the sender's jth message.

[20]Strictly speaking, $|0\rangle|x_0\rangle+|1\rangle|x_1\rangle$ is not equal to $|x_0\rangle+|x_1\rangle$, but the protocol can be easily modified. Given ξ, the prover has only to change $|0\rangle|x_0\rangle+|1\rangle|x_1\rangle$ to $|\xi\cdot x_0\rangle|x_0\rangle+|1\oplus(\xi\cdot x_1)\rangle|x_1\rangle$.

[21]In [MY23], they resolve the first problem by using a specific commitment scheme of [NOVY93] and resolve the second problem by simply assuming the existence of a trapdoor. However, since the commitment scheme of [NOVY93] relies on one-way *permutations*, their idea does not work based on OWFs even if we give up efficient verification.

bit commitment scheme. In a similar way as the soundness proof of [KMCVY22], we can construct a PPT adversary $\mathcal{A}$ that can find both x_0 and x_1 from a PPT malicious prover that passes both challenges with some high probability. We can then construct a PPT adversary $\mathcal{B}$ that breaks computational-binding of the classical bit commitment scheme from $\mathcal{A}$.

There is, however, a large difference in our case from that of [KMCVY22]. In the protocol of [KMCVY22], the honest prover's state is always $|x_0\rangle + |x_1\rangle$, but in our case $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not always satisfied. In order to keep the $1/\mathrm{poly}$ completeness-soundness gap in our protocol, we need a trick for the algorithm of the inefficient verifier. The inefficient verifier first checks whether $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is satisfied or not. If it is satisfied, the inefficient verifier computes the unique element $x_b \in X_{b,t} \cap h_b^{-1}(y)$ for each $b \in \{0,1\}$, and checks whether the transcript passes the second phase of the protocol of [KMCVY22] or not. On the other hand, if $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied, we need some trick. A naive attempt would be to always accept in such a case. Intuitively, this would give a $1/\mathrm{poly}$ completeness-soundness gap because we have a constant completeness-soundness gap conditioned on $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ by [KMCVY22] and such an event occurs with probability $1/\mathrm{poly}$ as explained above. However, there is a flaw in the argument because a malicious prover may change the probability that $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ holds. For example, if it can control the probability to be 1, then it passes the verification with probability 1, which is even higher than the honest quantum prover's success probability! Due to a similar reason, an attempt to let the inefficient verifier always reject when $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied also does not work. Our idea is to take the middle of the two attempts: If $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied, the inefficient verifier accepts with probability s and rejects with probability $1-s$, where s is the soundness parameter of the PoQ protocol of [KMCVY22], i.e., for any malicious prover, the verifier accepts with probability at most $s + \mathsf{negl}(\lambda)$. Let $p_{\mathbf{good}}$ be the probability that $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is satisfied in the interaction between the honest prover and the verifier. Then, the probability that the inefficient verifier accepts the honest prover is at least $p_{\mathbf{good}}c + (1-p_{\mathbf{good}})s$, where c is the completeness parameter of the PoQ protocol of [KMCVY22], i.e., the verifier accepts the honest prover with probability at least c. On the other hand, we show that the soundness parameter of our protocol is also s. (Intuitively, this is because if $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is satisfied, then a malicious prover can pass the verification with probability at most $s + \mathsf{negl}(\lambda)$ by the soundness of the PoQ protocol of [KMCVY22], and if $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied, the verifier accepts with probability s regardless of the prover's behavior.) Therefore, we have $p_{\mathbf{good}}c + (1-p_{\mathbf{good}})s - s = p_{\mathbf{good}}(c-s) \geq 1/\mathrm{poly}$, because $p_{\mathbf{good}} \geq 1/\mathrm{poly}$ as we have explained. In this way, we can achieve the $1/\mathrm{poly}$ completeness-soundness gap.

Finally, in our construction, the inefficient verifier is enough to be a classical deterministic polynomial-time algorithm that queries the $\mathbf{NP}$ oracle, because as we have explained above, inefficient computations that the inefficient verifier has to do are verifying $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ and finding the single element $x_b \in X_{b,t} \cap h_b^{-1}(y)$ for each $b \in \{0,1\}$.

1.3 Related Works

IV-PoQ from random oracles was constructed in [ACC+22], which they call Collision Hashing. Their construction is based on the observation that if the state $\sum_x |x\rangle |g(x)\rangle$ is generated, where g is a random oracle, and the second register is measured in the computational basis, the post-measurement state $\sum_{x \in g^{-1}(y)} |x\rangle$ corresponding to the measurement result y is a superposition of two computational-basis states with some probability on which the second phase of [KMCVY22] can be run. (Actually, because they assume random oracles, the non-interactive protocol of [BKVV20] can be run instead of [KMCVY22].) This idea seems to be somehow related to our idea.

[AA15] studied a sampling problem, Fourier Sampling, where given an oracle $f : \{0,1\}^n \to \{+1,-1\}$, it is required to sample from the distribution $\{p_y\}_y$, where $p_y := 2^{-n}\hat{f}(y)^2 = (\frac{1}{2^n}\sum_{x \in \{0,1\}^n} f(x)(-1)^{x \cdot y})^2$ within an additive error. It needs exponentially-many queries to classically solve it relative to a random oracle. [Aar10] also introduced a search problem, Fourier Fishing, where given an oracle $f : \{0,1\}^n \to \{+1,-1\}$, find $z \in \{0,1\}^n$ such that $|\hat{f}(z)| \geq 1$. It needs exponentially-many queries to classically solve it relative to a random oracle. The verification of Fourier Fishing can be done inefficiently. [Aar10] also introduced a decision problem, Fourier Checking, and show that it requires exponentially-many queries to solve it classically relative to a certain oracle. Whether $\mathbf{BQP} \neq \mathbf{BPP}$ relative to a random oracle is an open problem, and given the Aaronson-Ambainis conjecture [AA14], showing it seems to be difficult.

[AC17] showed that if OWFs exist, then there are oracles $A \in \mathbf{P}/\mathbf{poly}$ such that $\mathbf{BPP}^A \neq \mathbf{BQP}^A$ (and even $\mathbf{BQP}^A \not\subset \mathbf{SZK}^A$). The paper also showed that if there exist subexponentially-secure OWFs, then Fourier Sampling and Fourier Fishing are classically hard relative to oracles in $\mathbf{P}/\mathbf{poly}$. Regarding the possibility of removing the oracles, the authors say that "... *in the unrelativized world, there seems to be no hope at present of proving* $\mathbf{BPP} \neq \mathbf{BQP}$ *under any hypothesis nearly as weak as the existence of one-way functions*", which suggests the difficulty of demonstrating quantum advantage based only on one-way functions. We bypass the difficulty by considering interactive protocols.

It was pointed out in [LLQ22] that the complexity assumption of $\mathbf{PP} \neq \mathbf{BPP}$ is necessary for the existence of PoQ. A similar idea can be applied to show that $\mathbf{PP} \neq \mathbf{BPP}$ is necessary for the existence of (AI-/IO-)IV-PoQ. (For the convenience of readers, we provide a proof in Appendix A.) We remark that the proof holds even if we allow the honest prover to perform post-selection. Moreover, it holds even if the verifier in the first phase is unbounded-time.

Unconditional quantum advantage over restricted classical computing was also studied [BGK18, BGKT20, WKST19, GS20]. Unconditional separations between quantum and classical computing are appealing, but in this paper we do not focus on the setups of restricting classical computing. Note that showing unconditional quantum advantage without restricting classical computing is at least as hard as proving $\mathbf{PP} \neq \mathbf{BPP}$ ([LLQ22] and Appendix A), which is a major open problem in complexity theory.

The idea of coherently running statistically-hiding commitments was first introduced in [MY23]. However, they could apply the idea only to the specific commitment scheme of [NOVY93] whereas we can apply it to *any* statistically-hiding commitments. This is made possible by our new hashing technique as explained in Section 1.2.

2 Preliminaries

2.1 Basic Notations

We use the standard notations of quantum computing and cryptography. We use λ as the security parameter. $[n]$ means the set $\{1, 2, ..., n\}$. For any set S, $x \leftarrow S$ means that an element x is sampled uniformly at random from the set S. For a set S, $|S|$ means the cardinality of S. We write negl to mean a negligible function and poly to mean a polynomial. PPT stands for (classical) probabilistic polynomial-time and QPT stands for quantum polynomial-time. For an algorithm A, $y \leftarrow A(x)$ means that the algorithm A outputs y on input x. For two bit strings x and y, $x\|y$ means the concatenation of them. For simplicity, we sometimes omit the normalization factor of a quantum state. (For example, we write $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$ just as $|x_0\rangle + |x_1\rangle$.) $I := |0\rangle\langle 0| + |1\rangle\langle 1|$ is the two-dimensional identity operator. For the notational simplicity, we sometimes write $I^{\otimes n}$ just as I when the dimension is clear from the context.

2.2 Pairwise-Independent Hash Family

Definition 2.1. *A family of hash functions $\mathcal{H} := \{h : \mathcal{X} \to \mathcal{Y}\}$ is pairwise-independent if for any two $x \neq x' \in \mathcal{X}$ and any two $y, y' \in \mathcal{Y}$,*

$$\Pr_{h \leftarrow \mathcal{H}}[h(x) = y \wedge h(x') = y'] = \frac{1}{|\mathcal{Y}|^2}. \tag{3}$$

2.3 OWFs

Definition 2.2 (OWFs). *A function $f : \{0,1\}^* \to \{0,1\}^*$ is a (classically-secure) OWF if it is computable in classical deterministic polynomial-time, and for any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that for any λ,*

$$\Pr[f(x') = f(x) : x' \leftarrow \mathcal{A}(1^\lambda, f(x)), x \leftarrow \{0,1\}^\lambda] \leq \mathsf{negl}(\lambda). \tag{4}$$

Definition 2.3 (Infinitely-often OWFs). *A function $f : \{0,1\}^* \to \{0,1\}^*$ is a (classically-secure) infinitely-often OWF if it is computable in classical deterministic polynomial-time, and there exists an infinite set $\Lambda \subseteq \mathbb{N}$ such that for any PPT adversary $\mathcal{A}$,*

$$\Pr[f(x') = f(x) : x' \leftarrow \mathcal{A}(1^\lambda, f(x)), x \leftarrow \{0,1\}^\lambda] \leq \mathsf{negl}(\lambda) \tag{5}$$

for all $\lambda \in \Lambda$.

Definition 2.4 (Auxiliary-input function ensemble). *An auxiliary-input function ensemble is a collection of functions* $\mathcal{F} := \{f_\sigma : \{0,1\}^{p(|\sigma|)} \to \{0,1\}^{q(|\sigma|)}\}_{\sigma \in \{0,1\}^*}$*, where* p *and* q *are polynomials. We call* $\mathcal{F}$ *polynomial-time computable if there is a classical deterministic polynomial-time algorithm* F *such that for every* $\sigma \in \{0,1\}^*$ *and* $x \in \{0,1\}^{p(|\sigma|)}$*, we have* $F(\sigma, x) = f_\sigma(x)$*.*

Definition 2.5 (Auxiliary-input OWFs). *A (classically-secure) auxiliary-input OWF is a polynomial-time computable auxiliary-input function ensemble* $\mathcal{F} := \{f_\sigma : \{0,1\}^{p(|\sigma|)} \to \{0,1\}^{q(|\sigma|)}\}_{\sigma \in \{0,1\}^*}$ *such that for every uniform PPT adversary* $\mathcal{A}$ *and a polynomial* $\mathrm{poly}(\lambda)$*, there exists an infinite set* $\Lambda \subseteq \{0,1\}^*$ *such that,*

$$\Pr[f_\sigma(x') = f_\sigma(x) : x' \leftarrow \mathcal{A}(\sigma, f_\sigma(x)), x \leftarrow \{0,1\}^{p(|\sigma|)}] \le \frac{1}{\mathrm{poly}(|\sigma|)} \tag{6}$$

for all $\sigma \in \Lambda$*.*

Remark 2.6. It is easy to see that OWFs imply infinitely-often OWFs, and infinitely-often OWFs imply auxiliary-input OWFs.

Theorem 2.7 ([OW93]). *Auxiliary-input OWFs exist if* $\mathbf{CZK} \not\subseteq \mathbf{BPP}$*.*

Remark 2.8. As is pointed out in [Vad06], auxiliary-input OWFs secure against non-uniform PPT adversaries exist if $\mathbf{CZK} \not\subseteq \mathbf{P/poly}$.

2.4 Commitments

Definition 2.9 (Statistically-hiding and computationally-binding classical bit commitments). *A statistically-hiding and computationally-binding classical bit commitment scheme is an interactive protocol* $\langle \mathcal{S}, \mathcal{R} \rangle$ *between two PPT algorithms* $\mathcal{S}$ *(the sender) and* $\mathcal{R}$ *(the receiver) such that*

- *In the commit phase,* $\mathcal{S}$ *takes* $b \in \{0,1\}$ *and* 1^λ *as input and* $\mathcal{R}$ *takes* 1^λ *as input.* $\mathcal{S}$ *and* $\mathcal{R}$ *exchange classical messages. The transcript* t*, i.e., the sequence of all classical messages exchanged between* $\mathcal{S}$ *and* $\mathcal{R}$*, is called a commitment. At the end of the commit phase,* $\mathcal{S}$ *privately outputs a decommitment* decom*.*
- *In the open phase,* $\mathcal{S}$ *sends* (b, decom) *to* $\mathcal{R}$*.* $\mathcal{R}$ *on input* (t, b, decom) *outputs* $\top$ *or* $\bot$*.*

We require the following three properties.

Perfect Correctness: *For all* $\lambda \in \mathbb{N}$ *and* $b \in \{0,1\}$*, if* $\mathcal{S}(1^\lambda, b)$ *and* $\mathcal{R}(1^\lambda)$ *behave honestly,* $\Pr[\top \leftarrow \mathcal{R}] = 1$*.*

Statistical Hiding: *Let us consider the following security game between the honest sender* $\mathcal{S}$ *and a malicious receiver* $\mathcal{R}^*$*:*

1. $\mathcal{S}(b, 1^\lambda)$ *and* $\mathcal{R}^*(1^\lambda)$ *run the commit phase.*
2. $\mathcal{R}^*$ *outputs* $b' \in \{0,1\}$*.*

We say that the scheme is statistically hiding if for any computationally unbounded adversary $\mathcal{R}^*$*,*

$$|\Pr[0 \leftarrow \mathcal{R}^* | b = 0] - \Pr[0 \leftarrow \mathcal{R}^* | b = 1]| \le \mathsf{negl}(\lambda). \tag{7}$$

Computational Binding: *Let us consider the following security game between a malicious sender* $\mathcal{S}^*$ *and the honest receiver* $\mathcal{R}$*:*

1. $\mathcal{S}^*(1^\lambda)$ *and* $\mathcal{R}(1^\lambda)$ *run the commit phase to generate a commitment* t*.*
2. $\mathcal{S}^*$ *sends* $(0, \mathsf{decom}_0)$ *and* $(1, \mathsf{decom}_1)$ *to* $\mathcal{R}$*.*

We say that the scheme is computationally binding if for any PPT malicious $\mathcal{S}^*$,

$$\Pr[\top \leftarrow \mathcal{R}(0, \mathsf{decom}_0) \wedge \top \leftarrow \mathcal{R}(1, \mathsf{decom}_1)] \leq \mathsf{negl}(\lambda). \tag{8}$$

Statistically-hiding and computationally-binding bit commitments can be constructed from OWFs.

Theorem 2.10 ([HNO+09]). *If OWFs exist, then statistically-hiding and computationally-binding bit commitments exist.*

Moreover, constant-round schemes are known from collision-resistant hash functions [HM96]. The assumption is further weakened to the existence of distributional collision-resistant hash functions, which exist if there is an hard-on-average problem in **SZK**.

Theorem 2.11 ([KY18, BHKY19]). *If distributional collision-resistant hash functions exist, which exist if there is an hard-on-average problem in* **SZK***, then constant-round statistically-hiding and computationally-binding bit commitments exist.*

We define an infinitely-often variant of statistically-hiding and computationally-binding commitments as follows.

Definition 2.12 (Infinitely-often statistically-hiding and computationally-binding commitments). *Infinitely-often statistically-hiding and computationally-binding commitments are defined similarly to Definition 2.9 except that we require the existence of an infinite set $\Lambda \subseteq \mathbb{N}$ such that statistical hiding and computational binding hold for all $\lambda \in \Lambda$ instead of for all $\lambda \in \mathbb{N}$.*

By using infinitely-often OWFs instead of OWFs in the commitment scheme of [HNO+09], we obtain the following theorem. Since the construction and proof are almost identical to those of [HNO+09], we omit the details.

Theorem 2.13 (Infinitely-often variant of [HNO+09]). *If infinitely-often OWFs exist, then infinitely-often statistically-hiding and computationally-binding bit commitments exist.*

We also define an auxiliary-input variant of statistically-hiding and computationally-binding commitments. Intuitively, it is a family of commitment schemes indexed by an auxiliary input where correctness and statistical hiding hold for all auxiliary inputs and an "auxiliary-input" version of computational binding holds, i.e., for any PPT cheating sender $\mathcal{S}^*$, there is an infinite set of auxiliary inputs under which computational binding holds.

Definition 2.14 (Auxiliary-input statistically-hiding and computationally-binding classical bit commitments). *An auxiliary-input statistically-hiding and computationally-binding classical bit commitment scheme is an interactive protocol $\langle \mathcal{S}, \mathcal{R} \rangle$ between two PPT algorithms $\mathcal{S}$ (the sender) and $\mathcal{R}$ (the receiver) associated with an infinite subset $\Sigma \subseteq \{0,1\}^*$ such that*

- *In the commit phase, $\mathcal{S}$ takes $b \in \{0,1\}$ and the auxiliary input $\sigma \in \Sigma$ as input and $\mathcal{R}$ takes the auxiliary input σ as input. $\mathcal{S}$ and $\mathcal{R}$ exchange classical messages. The transcript t, i.e., the sequence of all classical messages exchanged between $\mathcal{S}$ and $\mathcal{R}$, is called a commitment. At the end of the commit phase, $\mathcal{S}$ privately outputs a decommitment* decom.
- *In the open phase, $\mathcal{S}$ sends (b, decom) to $\mathcal{R}$. $\mathcal{R}$ on input (t, b, decom) outputs $\top$ or $\bot$.*

We require the following properties:

Perfect Correctness: *For all $\sigma \in \Sigma$ and $b \in \{0,1\}$, if $\mathcal{S}(b, \sigma)$ and $\mathcal{R}(\sigma)$ behave honestly,* $\Pr[\top \leftarrow \mathcal{R}] = 1$.

Statistical Hiding: *Let us consider the following security game between the honest sender $\mathcal{S}$ and a malicious receiver $\mathcal{R}^*$:*

1. *$\mathcal{S}(b, \sigma)$ and $\mathcal{R}^*(\sigma)$ run the commit phase.*
2. *$\mathcal{R}^*$ outputs $b' \in \{0,1\}$.*

We say that the scheme is statistically hiding if for all $\sigma \in \Sigma$ and any computationally unbounded adversary $\mathcal{R}^*$,

$$|\Pr[0 \leftarrow \mathcal{R}^* | b = 0] - \Pr[0 \leftarrow \mathcal{R}^* | b = 1]| \leq \mathsf{negl}(|\sigma|). \tag{9}$$

Computational Binding: *Let us consider the following security game between a malicious sender $\mathcal{S}^*$ and the honest receiver $\mathcal{R}$:*

1. *$\mathcal{S}^*(\sigma)$ and $\mathcal{R}(\sigma)$ run the commit phase to generate a commitment t.*
2. *$\mathcal{S}^*$ sends $(0, \mathsf{decom}_0)$ and $(1, \mathsf{decom}_1)$ to $\mathcal{R}$.*

We say that the scheme is computationally binding if for any PPT malicious sender $\mathcal{S}^*$ and a polynomial poly*, there exists an infinite subset $\Lambda \subseteq \Sigma$ such that for any $\sigma \in \Lambda$,*

$$\Pr[\top \leftarrow \mathcal{R}(t, 0, \mathsf{decom}_0) \wedge \top \leftarrow \mathcal{R}(t, 1, \mathsf{decom}_1)] \leq \frac{1}{\mathrm{poly}(|\sigma|)}. \tag{10}$$

By using auxiliary-input OWFs instead of OWFs in the commitment scheme of [HNO+09], we obtain the following theorem. Since the construction and proof are almost identical to those of [HNO+09], we omit the details.

Theorem 2.15 (Auxiliary-input variant of [HNO+09]). *If auxiliary-input OWFs exist, then auxiliary-input statistically-hiding and computationally-binding bit commitments exist.*

Similarly, by using auxiliary-input collision-resistant hash functions instead of collision-resistant hash functions in the commitment scheme of [HM96], we obtain 2-round auxiliary-input statistically-hiding and computationally-binding bit commitments. As shown in Appendix B.1, auxiliary-input collision-resistant hash functions exist if and only if $\mathbf{PWPP} \nsubseteq \mathbf{FBPP}$. Thus, we obtain the following theorem.

Theorem 2.16 (Auxiliary-input variant of [HM96]). *If auxiliary-input collision-resistant hash functions exist, which exist if and only if $\mathbf{PWPP} \nsubseteq \mathbf{FBPP}$, then 2-round auxiliary-input statistically-hiding and computationally-binding bit commitments exist.*

In addition, we observe in Appendix B.2 that the instance-dependent commitments for $\mathbf{SZK}$ of [OV08] directly gives constant-round auxiliary-input statistically-hiding and computationally-binding bit commitments under the assumption that $\mathbf{SZK} \nsubseteq \mathbf{BPP}$.

Theorem 2.17 (Auxiliary-input variant of [OV08]). *If $\mathbf{SZK} \nsubseteq \mathbf{BPP}$, then constant-round auxiliary-input statistically-hiding and computationally-binding bit commitments exist.*

Remark 2.18. In the constructions for Theorems 2.15 and 2.16, we can set $\Sigma := \{0,1\}^*$. However, we do not know if this is possible for the construction for Theorem 2.17 given in Appendix B.2. This is why we introduce the subset Σ in Definition 2.14.

3 Hashing Lemmas

In this section, we show two useful lemmas, Lemma 3.1 and Lemma 3.2. Lemma 3.1 is used to show Lemma 3.2, and Lemma 3.2 is used in the proof of our main result.

Lemma 3.1. *Let $\mathcal{H} := \{h : \mathcal{X} \to \mathcal{Y}\}$ be a pairwise-independent hash family such that $|\mathcal{X}| \geq 2$. Let $S \subseteq \mathcal{X}$ be a subset of $\mathcal{X}$. For any $y \in \mathcal{Y}$,*

$$\Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| \geq 1] \geq \frac{|S|}{|\mathcal{Y}|} - \frac{|S|^2}{2|\mathcal{Y}|^2}. \tag{11}$$

Proof of Lemma 3.1. First, if $|S| = 0$, Equation (11) trivially holds. Second, let us consider the case when $|S| = 1$. In that case,

$$\Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| \geq 1] = \frac{1}{|\mathcal{Y}|} \tag{12}$$

$$\geq \frac{1}{|\mathcal{Y}|} - \frac{1}{2|\mathcal{Y}|^2} \tag{13}$$

$$= \frac{|S|}{|\mathcal{Y}|} - \frac{|S|^2}{2|\mathcal{Y}|^2}, \tag{14}$$

and therefore Equation (11) is satisfied. Here, the first equality comes from the fact that the probability that the unique element of S is mapped to y is $1/|\mathcal{Y}|$.

Finally, let us consider the case when $|S| \geq 2$. The following argument is based on [Sta22]. First, for each $y \in \mathcal{Y}$,

$$\sum_{j=1}^{|S|} j \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = j] = \mathop{\mathbb{E}}_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)|] \tag{15}$$

$$= \mathop{\mathbb{E}}_{h \leftarrow \mathcal{H}}[|\{x \in S : h(x) = y\}|] \tag{16}$$

$$= \sum_{x \in S} \Pr_{h \leftarrow \mathcal{H}}[h(x) = y] \tag{17}$$

$$= \frac{|S|}{|\mathcal{Y}|}. \tag{18}$$

Second, for each $y \in \mathcal{Y}$,

$$\sum_{j=1}^{|S|} (j-1) \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = j] \leq \sum_{j=1}^{|S|} \binom{j}{2} \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = j] \tag{19}$$

$$= \mathop{\mathbb{E}}_{h \leftarrow \mathcal{H}}\left[\binom{|S \cap h^{-1}(y)|}{2}\right] \tag{20}$$

$$= \mathop{\mathbb{E}}_{h \leftarrow \mathcal{H}}[|\{\{x, x'\} \subseteq S : x \neq x', h(x) = h(x') = y\}|] \tag{21}$$

$$= \sum_{\{x,x'\} \subseteq S, x \neq x'} \Pr_{h \leftarrow \mathcal{H}}[h(x) = h(x') = y] \tag{22}$$

$$= \frac{1}{|\mathcal{Y}|^2} \binom{|S|}{2} \tag{23}$$

$$= \frac{|S|(|S|-1)}{2|\mathcal{Y}|^2} \tag{24}$$

$$\leq \frac{|S|^2}{2|\mathcal{Y}|^2}. \tag{25}$$

(Note that $\binom{n}{m} = 0$ for any $n < m$.) By extracting both sides of Equation (18) and Equation (25), we have

$$\sum_{j=1}^{|S|} \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = j] \quad \geq \quad \frac{|S|}{|\mathcal{Y}|} - \frac{|S|^2}{2|\mathcal{Y}|^2}, \tag{26}$$

which shows Lemma 3.1. □

Lemma 3.2. *Let $\mathcal{H} := \{h : \mathcal{X} \to \mathcal{Y}\}$ be a pairwise-independent hash family such that $|\mathcal{X}| \geq 2$. Let $S \subseteq \mathcal{X}$ be a subset of $\mathcal{X}$. For any $y \in \mathcal{Y}$,*

$$\Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = 1] \geq \frac{|S|}{|\mathcal{Y}|} - \frac{|S|^2}{|\mathcal{Y}|^2}. \tag{27}$$

Proof of Lemma 3.2. For any $y \in \mathcal{Y}$,

$$\frac{|S|}{|\mathcal{Y}|} = \sum_{j=1}^{|S|} j \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = j] \tag{28}$$

$$\geq \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = 1] + 2 \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| \geq 2] \tag{29}$$

$$= 2 \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| \geq 1] - \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = 1] \tag{30}$$

$$\geq \frac{2|S|}{|\mathcal{Y}|} - \frac{|S|^2}{|\mathcal{Y}|^2} - \Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = 1]. \tag{31}$$

Here, the first equality is from Equation (18), and in the last inequality we have used Lemma 3.1. Therefore,

$$\Pr_{h \leftarrow \mathcal{H}}[|S \cap h^{-1}(y)| = 1] \geq \frac{2|S|}{|\mathcal{Y}|} - \frac{|S|^2}{|\mathcal{Y}|^2} - \frac{|S|}{|\mathcal{Y}|} \tag{32}$$

$$= \frac{|S|}{|\mathcal{Y}|} - \frac{|S|^2}{|\mathcal{Y}|^2}, \tag{33}$$

which shows Lemma 3.2. □

4 Inefficient-Verifier Proofs of Quantumness

In this section, we define inefficient-verifier proofs of quantumness (IV-PoQ) and its variants. Then we show that sequential repetition amplifies the completeness-soundness gap assuming a special property of soundness, which we call a strong soundness, for the base scheme.

4.1 Definitions

We define IV-PoQ. It is identical to the definition of PoQ, which are implicitly defined in [BCM+21], except that we allow the verifier to be unbounded-time after completing interaction with the prover.

Definition 4.1 (Inefficient-verifier proofs of quantumness (IV-PoQ)). *An inefficient-verifier proof of quantumness (IV-PoQ) is an interactive protocol $(\mathcal{P}, \mathcal{V})$ between a QPT algorithm $\mathcal{P}$ (the prover) and an algorithm $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2)$ (the verifier) where $\mathcal{V}_1$ is PPT and $\mathcal{V}_2$ is unbounded-time. The protocol is divided into two phases. In the first phase, $\mathcal{P}$ and $\mathcal{V}_1$ take the security parameter 1^λ as input and interact with each other over a classical channel. Let I be the transcript, i.e., the sequence of all classical messages exchanged between $\mathcal{P}$ and $\mathcal{V}_1$. In the second phase, $\mathcal{V}_2$ takes I as input and outputs $\top$ or $\bot$. We require the following two properties for some functions c and s such that $c(\lambda) - s(\lambda) \geq 1/\mathrm{poly}(\lambda)$.*

c-completeness:

$$\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}(1^\lambda), \mathcal{V}_1(1^\lambda) \rangle] \geq c(\lambda) - \mathsf{negl}(\lambda). \tag{34}$$

s-soundness: *For any PPT malicious prover $\mathcal{P}^*$,*

$$\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}^*(1^\lambda), \mathcal{V}_1(1^\lambda) \rangle] \leq s(\lambda) + \mathsf{negl}(\lambda). \tag{35}$$

Remark 4.2. $\mathcal{V}_2$ could take $\mathcal{V}_1$'s secret information as input in addition to I, but without loss of generality, we can assume that $\mathcal{V}_2$ takes only I, because we can always modify the protocol of the first phase in such a way that $\mathcal{V}_1$ sends its secret information to $\mathcal{P}$ at the end of the first phase.

Remark 4.3. In our constructions, $\mathcal{V}_2$ is actually enough to be a classical deterministic polynomial-time algorithm that queries the $\mathbf{NP}$ oracle. (See Section 6.3.)

We define an infinitely-often version of IV-PoQ as follows.

Definition 4.4 (Infinitely-often inefficient-verifier proofs of quantumness (IO-IV-PoQ)). *An infinitely-often inefficient-verifier proofs of quantumness (IO-IV-PoQ) is defined similarly to IV-PoQ (Definition 4.1) except that we require the existence of an infinite set $\Lambda \subseteq \mathbb{N}$ such that c-completeness and s-soundness hold for all $\lambda \in \Lambda$ instead of for all $\lambda \in \mathbb{N}$.*

We also define an auxiliary-input variant of IV-PoQ as follows. It is defined similarly to IV-PoQ except that the prover and verifier take an auxiliary input instead of the security parameter and completeness should hold for all auxiliary inputs wheres soundness is replaced to auxiliary-input soundness, i.e., for any PPT cheating prover $\mathcal{P}^*$, there exists an infinite set of auxiliary inputs under which soundness holds.

Definition 4.5 (Auxiliary-input inefficient-verifier proofs of quantumness (AI-IV-PoQ)). *An auxiliary-input inefficient-verifier proof of quantumness (AI-IV-PoQ) is an interactive protocol $(\mathcal{P},\mathcal{V})$ between a QPT algorithm $\mathcal{P}$ (the prover) and an algorithm $\mathcal{V}=(\mathcal{V}_1,\mathcal{V}_2)$ (the verifier) where $\mathcal{V}_1$ is PPT and $\mathcal{V}_2$ is unbounded-time, associated with an infinite set $\Sigma \subseteq \{0,1\}^*$. The protocol is divided into two phases. In the first phase, $\mathcal{P}$ and $\mathcal{V}_1$ take an auxiliary input $\sigma \in \Sigma$ as input and interact with each other over a classical channel. Let I be the transcript, i.e., the sequence of all classical messages exchanged between $\mathcal{P}$ and $\mathcal{V}_1$. In the second phase, $\mathcal{V}_2$ takes I as input and outputs $\top$ or $\bot$. We require the following two properties for some functions c and s such that $c(|\sigma|)-s(|\sigma|)\geq 1/\mathrm{poly}(|\sigma|)$.*

c-completeness: *For any $\sigma\in\Sigma$,*

$$\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}(\sigma), \mathcal{V}_1(\sigma)\rangle] \geq c(|\sigma|) - \mathsf{negl}(|\sigma|). \tag{36}$$

s-soundness: *For any PPT malicious prover $\mathcal{P}^*$ and polynomial p, there exists an infinite set $\Lambda \subseteq \Sigma$ such that*

$$\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}^*(\sigma), \mathcal{V}_1(\sigma)\rangle] \leq s(|\sigma|) + \frac{1}{p(|\sigma|)} \tag{37}$$

for all $\sigma \in \Lambda$.

Remark 4.6. We can set $\Sigma := \{0,1\}^*$ for all our constructions of AI-IV-PoQ except for the one based on $\mathbf{SZK} \neq \mathbf{BPP}$. See also Remark 2.18.

Remark 4.7. It is easy to see that IV-PoQ imply IO-IV-PoQ, and IO-IV-PoQ imply AI-IV-PoQ.

Even though AI-IV-PoQ is weaker than IV-PoQ, we believe that it still demonstrates a meaningful notion of quantum advantage, because it shows "worst-case quantum advantage" in the sense that no PPT algorithm can simulate the QPT honest prover on all auxiliary inputs $\sigma\in\Sigma$.

4.2 Strong Soundness

Unfortunately, we do not know if parallel or even sequential repetition amplifies the completeness-soundness gap for general (AI-/IO-)IV-PoQ. Here, we define a stronger notion of soundness which we call strong soundness. In Section 4.3, we show that sequential repetition amplifies the completeness-soundness gap if the base scheme satisfies strong soundness. In Section 6, we show that our (AI-/IO-)IV-PoQ satisfies strong soundness. Thus, gap amplification by sequential repetition works for our particular constructions of (AI-/IO-)IV-PoQ.

Roughly, the s-strong-soundness requires that a PPT cheating prover can pass verification with probability at most $\approx s$ for *almost all fixed randomness*. The formal definition is given below.

Definition 4.8 (Strong soundness for IV-PoQ). *We say that an IV-PoQ $(\mathcal{P},\mathcal{V}=(\mathcal{V}_1,\mathcal{V}_2))$ satisfies s-strong-soundness if the following holds:*

s-strong-soundness: *For any PPT malicious prover $\mathcal{P}^*$ and any polynomial p,*

$$\Pr_{r\leftarrow\mathcal{R}}\left[\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}^*_r(1^\lambda), \mathcal{V}_1(1^\lambda)\rangle] \geq s(\lambda) + \frac{1}{p(\lambda)}\right] \leq \frac{1}{p(\lambda)} \tag{38}$$

for all sufficiently large λ where $\mathcal{R}$ is the randomness space for $\mathcal{P}^*$ and $\mathcal{P}^*_r$ is $\mathcal{P}^*$ with the fixed randomness r.

It is defined similarly for (AI-/IO-)IV-PoQ.

It is easy to see that s-strong-soundness implies s-soundness.

Lemma 4.9. *For any s, s-strong-soundness implies s-soundness for (AI-/IO-)IV-PoQ.*

Algorithm 1 N-sequential-repetition version $\Pi^{N\text{-}\mathsf{seq}}$ of Π

The first phase: The QPT prover $\mathcal{P}^{N\text{-}\mathsf{seq}}$ and PPT verifier $\mathcal{V}_1^{N\text{-}\mathsf{seq}}$ run the first phase of Π sequentially N times (i.e., after they finish i-th execution, they start $(i+1)$-st execution) where $\mathcal{P}^{N\text{-}\mathsf{seq}}$ plays the role of $\mathcal{P}$ and $\mathcal{V}_1^{N\text{-}\mathsf{seq}}$ plays the role of $\mathcal{V}_1$. Let I_i be the transcript of i-th execution of Π for $i \in [N]$.

The second phase: The unbounded-time $\mathcal{V}_2^{N\text{-}\mathsf{seq}}$ takes the transcript $\{I_i\}_{i\in[N]}$ of the first phase as input. For $i \in [N]$, it runs $\mathcal{V}_2$ on I_i and sets $X_i := 1$ if it accepts and otherwise sets $X_i := 0$. If $\frac{\sum_{i\in[N]} X_i}{N} \geq \frac{c(\lambda)+s(\lambda)}{2}$, it outputs $\top$ and otherwise outputs $\bot$.

Proof. We focus on the case of IV-PoQ since the cases of (AI-/IO-)IV-PoQ are similar. If there is a PPT malicious prover $\mathcal{P}^*$ that breaks s-soundness of IV-PoQ, then there exists a polynomial p such that

$$\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}^*(1^\lambda), \mathcal{V}_1(1^\lambda)\rangle] \geq s(\lambda) + \frac{3}{p(\lambda)} \tag{39}$$

for infinitely many λ. By a standard averaging argument, this implies

$$\Pr_{r\leftarrow\mathcal{R}}\left[\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}_r^*(1^\lambda), \mathcal{V}_1(1^\lambda)\rangle] \geq s(\lambda) + \frac{1}{p(\lambda)}\right] \geq \frac{2}{p(\lambda)} \tag{40}$$

for infinitely many λ. This contradicts s-strong-soundness. Thus, Lemma 4.9 holds. $\square$

We remark that the other direction does not seem to hold. For example, suppose that a PPT malicious prover $\mathcal{P}^*$ passes the verification with probability 1 for 0.99-fraction of randomness and with probability 0 for the rest of randomness. In this case, $\mathcal{P}^*$ breaks 0.99-soundness. On the other hand, it does not break s-strong-soundness for any constant $s > 0$.

4.3 Gap Amplification

We prove that sequential repetition amplifies the completeness-soundness gap if the base scheme satisfies strong soundness.

Theorem 4.10 (Gap amplification theorem). *Let $\Pi = (\mathcal{P}, \mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2))$ be an (AI-/IO-)IV-PoQ that satisfies c-completeness and s-strong-soundness where $c(\lambda) - s(\lambda) \geq 1/\mathrm{poly}(\lambda)$ and c and s are computable in polynomial-time. Let $\Pi^{N\text{-}\mathsf{seq}} = (\mathcal{P}^{N\text{-}\mathsf{seq}}, \mathcal{V}^{N\text{-}\mathsf{seq}} = (\mathcal{V}_1^{N\text{-}\mathsf{seq}}, \mathcal{V}_2^{N\text{-}\mathsf{seq}}))$ be its N-sequential-repetition version as described in Algorithm 1. If $N \geq \frac{\lambda}{(c(\lambda)-s(\lambda))^2}$, then $\Pi^{N\text{-}\mathsf{seq}}$ satisfies 1-completeness and 0-soundness.*

Remark 4.11. Note that the meaning of "sequential repetition" is slightly different from that for usual interactive arguments: we defer the second phases of each execution to the end of the protocol so that inefficient computations are only needed after completing the interaction.

Remark 4.12. If we assume s-soundness against *non-uniform* PPT adversaries, we can easily prove a similar amplification theorem without introducing strong soundness. However, for proving soundness against non-uniform PPT adversaries, we would need non-uniform hardness assumptions such as non-uniformly secure OWFs. Since our motivation is to demonstrate quantum advantage from the standard notion of *uniformly* secure OWFs, we do not take the above approach.

Proof of Theorem 4.10. We focus on the case of IV-PoQ since it is almost identical for (AI-/IO-)IV-PoQ. First, 1-completeness of $\Pi^{N\text{-}\mathsf{seq}}$ immediately follows from Hoeffding's inequality. (Recall that 1-completeness means that the honest prover's acceptance probability is at least $1 - \mathsf{negl}(\lambda)$.) In the following, we prove that $\Pi^{N\text{-}\mathsf{seq}}$ satisfies

0-soundness. Suppose that it does not satisfy 0-soundness. Then, there is a PPT malicious prover $\mathcal{P}^{N\text{-seq}^*}$ against $\Pi^{N\text{-seq}}$ and a polynomial p such that

$$\Pr[\top \leftarrow \mathcal{V}_2(I_1,\ldots,I_N) : (I_1,\ldots,I_N) \leftarrow \langle \mathcal{P}^{N\text{-seq}^*}(1^\lambda), \mathcal{V}_1^{N\text{-seq}}(1^\lambda)\rangle] \geq \frac{1}{p(\lambda)} \tag{41}$$

for infinitely many λ. We define random variables $X_1,\ldots,X_N$ as in Algorithm 1, i.e., $X_i = 1$ if $\mathcal{V}_2$ accepts the i-th transcript I_i and otherwise $X_i = 0$. Using this notation, the above inequality can be rewritten as

$$\Pr\left[\frac{\sum_{i\in[N]} X_i}{N} \geq \frac{c(\lambda)+s(\lambda)}{2}\right] \geq \frac{1}{p(\lambda)} \tag{42}$$

for infinitely many λ. For $i \in [N]$, let X_i' be an independent random variable over $\{0,1\}$ such that $\Pr[X_i' = 1] = s(\lambda) + \frac{1}{2Np(\lambda)}$. Noting that $\frac{c(\lambda)+s(\lambda)}{2} - (s(\lambda) + \frac{1}{2Np(\lambda)}) \geq \frac{c(\lambda)-s(\lambda)}{4}$ for sufficiently large λ,[22] by Hoeffding's inequality, we have

$$\Pr\left[\frac{\sum_{i\in[N]} X_i'}{N} \geq \frac{c(\lambda)+s(\lambda)}{2}\right] \leq \mathsf{negl}(\lambda). \tag{43}$$

Moreover, we prove below that for any $k \in [N]$, we have

$$\Pr\left[\frac{\sum_{i=1}^{k} X_i + \sum_{i=k+1}^{N} X_i'}{N} \geq \frac{c(\lambda)+s(\lambda)}{2}\right] - \Pr\left[\frac{\sum_{i=1}^{k-1} X_i + \sum_{i=k}^{N} X_i'}{N} \geq \frac{c(\lambda)+s(\lambda)}{2}\right] \leq \frac{1}{2Np(\lambda)} \tag{44}$$

for sufficiciently large λ. By a standard hybrid argument, Equations (42) and (44) imply

$$\Pr\left[\frac{\sum_{i\in[N]} X_i'}{N} \geq \frac{c(\lambda)+s(\lambda)}{2}\right] \geq \frac{1}{2p(\lambda)}. \tag{45}$$

for infinitely many λ. This contradicts Equation (43). Thus, we only have to prove Equation (44) holds for all $k \in [N]$ and sufficiently large λ.

Proof of Equation (44). Let $\mathcal{P}_k^*$ be a malicious prover against Π that works as follows: $\mathcal{P}_k^*$ first simulates the interaction between $\mathcal{P}^{N\text{-seq}^*}$ and $\mathcal{V}_1^{N\text{-seq}}$ for the first $k-1$ executions of Π where it also simulates $\mathcal{V}_1^{N\text{-seq}}$ by itself in this phase. Then $\mathcal{P}_k^*$ starts interaction with the external verifier $\mathcal{V}_1$ of Π where it works similarly to $\mathcal{P}^{N\text{-seq}^*}$ in the kth execution of Π. Note that the randomness of $\mathcal{P}_k^*$ consists of the randomness r_P for $\mathcal{P}^{N\text{-seq}^*}$ and randomness r_V^{k-1} for $\mathcal{V}_1^{N\text{-seq}}$ for the first $k-1$ executions of Π. Therefore, by applying s-strong-soundness of Π for the above malicious prover $\mathcal{P}_k^*$, there is a set of $\left(1 - \frac{1}{2Np(\lambda)}\right)$-fraction of (r_P, r_V^{k-1}), which we denote by $\mathcal{G}_{k-1}$, such that for all $(r_P, r_V^{k-1}) \in \mathcal{G}_{k-1}$,

$$\Pr[X_k = 1 | r_P, r_V^{k-1}] \leq s(\lambda) + \frac{1}{2Np(\lambda)} \tag{46}$$

for sufficiently large λ, where $\Pr[X_k = 1 | r_P, r_V^{k-1}]$ means the conditional probability that $X_k = 1$ occurs conditioned on the fixed values of (r_P, r_V^{k-1}). On the other hand, because X_k' is independent of (r_P, r_V^{k-1}),

$$\Pr[X_k' = 1 | r_P, r_V^{k-1}] = \Pr[X_k' = 1] = s(\lambda) + \frac{1}{2Np(\lambda)} \tag{47}$$

[22]This can be seen as follows: $\frac{c(\lambda)+s(\lambda)}{2} - (s(\lambda) + \frac{1}{2Np(\lambda)}) = \frac{c(\lambda)-s(\lambda)}{2} - \frac{1}{2Np(\lambda)} \geq \frac{c(\lambda)-s(\lambda)}{2} - \frac{1}{2N} \geq \frac{c(\lambda)-s(\lambda)}{2} - \frac{(c(\lambda)-s(\lambda))^2}{2\lambda} \geq \frac{c(\lambda)-s(\lambda)}{2} - \frac{c(\lambda)-s(\lambda)}{4} = \frac{c(\lambda)-s(\lambda)}{4}$ for sufficiently large λ where the first inequality follows from the assumption $p(\lambda) \geq 1$, the second inequality follows from $N \geq \frac{\lambda}{(c(\lambda)-s(\lambda))^2}$, the third inequality follows from $2\lambda \geq 4$ and $0 \leq c(\lambda) - s(\lambda) \leq 1$ for sufficiently large λ.

for any fixed (r_P, r_V^{k-1}).

For notational simplicity, we denote the events that $\frac{\sum_{i=1}^{k} X_i + \sum_{i=k+1}^{N} X'_i}{N} \geq \frac{c(\lambda)+s(\lambda)}{2}$ and $\frac{\sum_{i=1}^{k-1} X_i + \sum_{i=k}^{N} X'_i}{N} \geq \frac{c(\lambda)+s(\lambda)}{2}$ by E_k and E_{k-1}, respectively. Then for any $(r_P, r_V^{k-1}) \in \mathcal{G}_{k-1}$,

$$\Pr\left[E_k \middle| (r_P, r_V^{k-1})\right] \tag{48}$$

$$= \Pr\left[\frac{X_k}{N} \geq \frac{c(\lambda)+s(\lambda)}{2} - \frac{\sum_{i=1}^{k-1} X_i + \sum_{i=k+1}^{N} X'_i}{N} \middle| (r_P, r_V^{k-1})\right] \tag{49}$$

$$\leq \Pr\left[\frac{X'_k}{N} \geq \frac{c(\lambda)+s(\lambda)}{2} - \frac{\sum_{i=1}^{k-1} X_i + \sum_{i=k+1}^{N} X'_i}{N} \middle| (r_P, r_V^{k-1})\right] \tag{50}$$

$$= \Pr\left[E_{k-1} \middle| (r_P, r_V^{k-1})\right] \tag{51}$$

for sufficiently large λ where Equation (50) follows from Equations (46) and (47) and the observations that $X_1, \ldots, X_{k-1}$ are determined by (r_P, r_V^{k-1}) and $X'_{k+1}, \ldots, X'_N$ are independent of X_k or X'_k.

Then, for sufficiently large λ, we have

$$\Pr[E_k] - \Pr[E_{k-1}] \tag{52}$$

$$= \left(\Pr\left[E_k \wedge (r_P, r_V^{k-1}) \in \mathcal{G}_{k-1}\right] - \Pr\left[E_{k-1} \wedge (r_P, r_V^{k-1}) \in \mathcal{G}_{k-1}\right]\right) \tag{53}$$

$$+ \left(\Pr\left[E_k \wedge (r_P, r_V^{k-1}) \notin \mathcal{G}_{k-1}\right] - \Pr\left[E_{k-1} \wedge (r_P, r_V^{k-1}) \notin \mathcal{G}_{k-1}\right]\right) \tag{54}$$

$$\leq \Pr\left[(r_P, r_V^{k-1}) \in \mathcal{G}_{k-1}\right] \cdot \left(\Pr\left[E_k \middle| (r_P, r_V^{k-1}) \in \mathcal{G}_{k-1}\right] - \Pr\left[E_{k-1} \middle| (r_P, r_V^{k-1}) \in \mathcal{G}_{k-1}\right]\right) \tag{55}$$

$$+ \Pr\left[(r_P, r_V^{k-1}) \notin \mathcal{G}_{k-1}\right] \tag{56}$$

$$\leq \frac{1}{2Np(\lambda)}, \tag{57}$$

where Equation (57) follows from Equations (48)-(51) and the fact that $\mathcal{G}_{k-1}$ consists of $\left(1 - \frac{1}{2Np(\lambda)}\right)$-fraction of (r_P, r_V^{k-1}). This implies Equation (44) and completes the proof of Theorem 4.10. □

5 Coherent Execution of Classical Bit Commitments

In this section, we explain our key concept, namely, executing classical bit commitments coherently.

Let $(\mathcal{S}, \mathcal{R})$ be a classical bit commitment scheme. If we explicitly consider the randomness, the commit phase can be described as in Algorithm 2.

Now let us consider the coherent execution of Algorithm 2, which is shown in Algorithm 3. Let $t :=$ $(\alpha_1, \beta_1, \alpha_2, \beta_2, ..., \alpha_L, \beta_L)$ be the transcript obtained in the execution of Algorithm 3. At the end of the execution of Algorithm 3, $\mathcal{S}$ possesses the state

$$\frac{1}{\sqrt{|X_{0,t}| + |X_{1,t}|}} \sum_{b \in \{0,1\}} \sum_{x \in X_{b,t}} |b\rangle |x\rangle, \tag{63}$$

where

$$X_{b,t} \coloneqq \bigcap_{j=0}^{L} X_b^j = \left\{x \in \{0,1\}^\ell : \bigwedge_{j=1}^{L} f_j(b, x, \alpha_1, \beta_1, ..., \alpha_{j-1}, \beta_{j-1}) = \alpha_j\right\}. \tag{64}$$

The probability $\Pr[t]$ that the transcript t is obtained in the execution of Algorithm 3 is

$$\Pr[t] = \frac{|R_t|}{2^\ell} \frac{|X_{0,t}| + |X_{1,t}|}{2^{\ell+1}}, \tag{65}$$

Algorithm 2 The commit phase of a classical bit commitment scheme

1. The sender $\mathcal{S}$ takes the comitted bit $b \in \{0,1\}$ and the security parameter 1^λ as input. The receiver $\mathcal{R}$ takes 1^λ as input.
2. $\mathcal{R}$ samples a random seed $r \leftarrow \{0,1\}^\ell$, where $\ell := \mathrm{poly}(\lambda)$.
3. $\mathcal{S}$ samples a random seed $x \leftarrow \{0,1\}^\ell$. (Without loss of generality, we assume that $\mathcal{R}$'s random seed and $\mathcal{S}$'s random seed are of equal length.)
4. Let L be the number of rounds. For $j = 1$ to L, $\mathcal{S}$ and $\mathcal{R}$ repeat the following.
   (a) $\mathcal{S}$ computes $\alpha_j := f_j(b, x, \alpha_1, \beta_1, \alpha_2, \beta_2, ..., \alpha_{j-1}, \beta_{j-1})$, and sends α_j to $\mathcal{R}$. Here, f_j is the function that computes $\mathcal{S}$'s jth message.
   (b) $\mathcal{R}$ computes $\beta_j := g_j(r, \alpha_1, \beta_1, \alpha_2, \beta_2, ..., \alpha_j)$, and sends β_j to $\mathcal{S}$. Here, g_j is the function that computes $\mathcal{R}$'s jth message.

where

$$R_t := \left\{ r \in \{0,1\}^\ell : \bigwedge_{j=1}^{L} g_j(r, \alpha_1, \beta_1, ..., \alpha_j) = \beta_j \right\}. \tag{66}$$

In the remaining of this section, we show two lemmas, Lemma 5.1 and Lemma 5.2, that will be used in the proofs of our main results.

The following Lemma 5.1 roughly claims that $|X_{0,t}|$ and $|X_{1,t}|$ are almost equal with overwhelming probability.

Lemma 5.1. *Let $0 < \epsilon < 1$ be a constant. Define the set*

$$T := \{t : (1-\epsilon)|X_{1,t}| < |X_{0,t}| < (1+\epsilon)|X_{1,t}|\}. \tag{67}$$

Then,

$$\sum_{t \in T} \Pr[t] \geq 1 - \mathsf{negl}(\lambda). \tag{68}$$

Here, $\Pr[t]$ is the probability that the transcript t is obtained in the execution of Algorithm 3 as is given in Equation (65).

Proof of Lemma 5.1. Intuitively, this follows from the statistical hiding property since whenever $t \notin T$, an unbounded adversary can guess the committed bit from the transcript t with probability $1/2 + \Omega(\epsilon)$. Below, we provide a formal proof.

Define

$$T^+ := \{t : (1+\epsilon)|X_{1,t}| \leq |X_{0,t}|\}, \tag{69}$$
$$T^- := \{t : |X_{0,t}| \leq (1-\epsilon)|X_{1,t}|\}. \tag{70}$$

In order to show the lemma, we want to show that

$$\sum_{t \in T^+ \cup T^-} \Pr[t] \leq \mathsf{negl}(\lambda). \tag{71}$$

To show this, assume that

$$\sum_{t \in T^+ \cup T^-} \Pr[t] \geq \frac{1}{\mathrm{poly}(\lambda)} \tag{72}$$

for infinitely many λ. Then the following computationally-unbounded malicious receiver $\mathcal{R}^*$ can break the statistical hiding of the classical bit commitment scheme in Algorithm 2.

Algorithm 3 Coherent execution of the commit phase of a classical bit commitment scheme

1. The sender $\mathcal{S}$ takes 1^λ as input. The receiver $\mathcal{R}$ takes 1^λ as input.
2. $\mathcal{R}$ samples a random seed $r \leftarrow \{0,1\}^\ell$.
3. $\mathcal{S}$ generates the state $\sum_{b\in\{0,1\}}\sum_{x\in\{0,1\}^\ell}|b\rangle|x\rangle$.
4. Let L be the number of rounds. For $j=1$ to L, $\mathcal{S}$ and $\mathcal{R}$ repeat the following.
   (a) $\mathcal{S}$ possesses the state
   $$\sum_{b\in\{0,1\}}\sum_{x\in\cap_{i=0}^{j-1}X_b^i}|b\rangle|x\rangle, \tag{58}$$
   where
   $$X_b^0 := \{0,1\}^\ell, \tag{59}$$
   and
   $$X_b^i := \{x\in\{0,1\}^\ell : f_i(b,x,\alpha_1,\alpha_2,...,\alpha_{i-1},\beta_{i-1})=\alpha_i\} \tag{60}$$
   for $i\geq 1$.
   (b) $\mathcal{S}$ generates the state
   $$\sum_{b\in\{0,1\}}\sum_{x\in\cap_{i=0}^{j-1}X_b^i}|b\rangle|x\rangle|f_j(b,x,\alpha_1,\beta_1,...,\alpha_{j-1},\beta_{j-1})\rangle, \tag{61}$$
   and measures the third register in the computational basis to obtain the measurement result α_j. The post-measurement state is
   $$\sum_{b\in\{0,1\}}\sum_{x\in\cap_{i=0}^{j}X_b^i}|b\rangle|x\rangle. \tag{62}$$
   $\mathcal{S}$ sends α_j to $\mathcal{R}$.
   (c) $\mathcal{R}$ computes $\beta_j := g_j(r,\alpha_1,\beta_1,...,\alpha_j)$, and sends β_j to $\mathcal{S}$.

---

1. $\mathcal{R}^*$ honestly executes the commit phase with $\mathcal{S}$. Let t be the transcript obtained in the execution.
2. If $t\in T^+$, $\mathcal{R}^*$ outputs 0. If $t\in T^-$, $\mathcal{R}^*$ outputs 1. If $t\in T$, $\mathcal{R}^*$ outputs 0 with probability $1/2$ and outputs 1 with probability $1/2$.

The probability that $\mathcal{R}^*$ outputs 0 when $\mathcal{S}$ commits $b\in\{0,1\}$ is

$$\begin{aligned}\Pr[0\leftarrow\mathcal{R}^*|b] &= \sum_{t\in T^+}\Pr[t|b]+\frac{1}{2}\sum_{t\in T}\Pr[t|b] &(73)\\ &= \sum_{t\in T^+}\frac{|R_t|}{2^\ell}\frac{|X_{b,t}|}{2^\ell}+\frac{1}{2}\sum_{t\in T}\frac{|R_t|}{2^\ell}\frac{|X_{b,t}|}{2^\ell}. &(74)\end{aligned}$$

Therefore

$$\Pr[0 \leftarrow \mathcal{R}^*|b=0] - \Pr[0 \leftarrow \mathcal{R}^*|b=1] \tag{75}$$

$$= \sum_{t\in T^+} \Pr[t|b=0] + \frac{1}{2}\sum_{t\in T}\Pr[t|b=0] - \sum_{t\in T^+}\Pr[t|b=1] - \frac{1}{2}\sum_{t\in T}\Pr[t|b=1] \tag{76}$$

$$= \sum_{t\in T^+} \Pr[t|b=0] + \frac{1}{2}\Big(1 - \sum_{t\in T^+}\Pr[t|b=0] - \sum_{t\in T^-}\Pr[t|b=0]\Big) \tag{77}$$

$$- \sum_{t\in T^+} \Pr[t|b=1] - \frac{1}{2}\Big(1 - \sum_{t\in T^+}\Pr[t|b=1] - \sum_{t\in T^-}\Pr[t|b=1]\Big) \tag{78}$$

$$= \frac{1}{2}\sum_{t\in T^+}(\Pr[t|b=0] - \Pr[t|b=1]) + \frac{1}{2}\sum_{t\in T^-}(\Pr[t|b=1] - \Pr[t|b=0]) \tag{79}$$

$$= \frac{1}{2}\sum_{t\in T^+}\frac{|R_t|}{2^\ell}\frac{|X_{0,t}|-|X_{1,t}|}{2^\ell} + \frac{1}{2}\sum_{t\in T^-}\frac{|R_t|}{2^\ell}\frac{|X_{1,t}|-|X_{0,t}|}{2^\ell} \tag{80}$$

$$\geq \frac{1}{2}\sum_{t\in T^+}\frac{|R_t|}{2^\ell}\frac{|X_{0,t}|-\frac{|X_{0,t}|}{1+\epsilon}}{2^\ell} + \frac{1}{2}\sum_{t\in T^-}\frac{|R_t|}{2^\ell}\frac{|X_{1,t}|-(1-\epsilon)|X_{1,t}|}{2^\ell} \tag{81}$$

$$= \frac{1}{2}\Big(1-\frac{1}{1+\epsilon}\Big)\sum_{t\in T^+}\frac{|R_t|}{2^\ell}\frac{|X_{0,t}|}{2^\ell} + \frac{\epsilon}{2}\sum_{t\in T^-}\frac{|R_t|}{2^\ell}\frac{|X_{1,t}|}{2^\ell} \tag{82}$$

$$= \frac{1}{2}\frac{\epsilon}{1+\epsilon}\sum_{t\in T^+}\frac{|R_t|}{2^\ell}\frac{2|X_{0,t}|}{2^{\ell+1}} + \frac{\epsilon}{2}\sum_{t\in T^-}\frac{|R_t|}{2^\ell}\frac{2|X_{1,t}|}{2^{\ell+1}} \tag{83}$$

$$\geq \frac{1}{2}\frac{\epsilon}{1+\epsilon}\sum_{t\in T^+}\frac{|R_t|}{2^\ell}\frac{|X_{0,t}|+|X_{1,t}|}{2^{\ell+1}} + \frac{\epsilon}{2}\sum_{t\in T^-}\frac{|R_t|}{2^\ell}\frac{|X_{0,t}|+|X_{1,t}|}{2^{\ell+1}} \tag{84}$$

$$\geq \frac{1}{2}\frac{\epsilon}{1+\epsilon}\sum_{t\in T^+}\frac{|R_t|}{2^\ell}\frac{|X_{0,t}|+|X_{1,t}|}{2^{\ell+1}} + \frac{\epsilon}{2(1+\epsilon)}\sum_{t\in T^-}\frac{|R_t|}{2^\ell}\frac{|X_{0,t}|+|X_{1,t}|}{2^{\ell+1}} \tag{85}$$

$$= \frac{1}{2}\frac{\epsilon}{1+\epsilon}\sum_{t\in T^+\cup T^-}\frac{|R_t|}{2^\ell}\frac{|X_{0,t}|+|X_{1,t}|}{2^{\ell+1}} \tag{86}$$

$$= \frac{1}{2}\frac{\epsilon}{1+\epsilon}\sum_{t\in T^+\cup T^-}\Pr[t] \tag{87}$$

$$\geq \frac{1}{2}\frac{\epsilon}{1+\epsilon}\frac{1}{\mathrm{poly}(\lambda)} \tag{88}$$

$$= \frac{1}{\mathrm{poly}(\lambda)} \tag{89}$$

for infinitely many λ, which breaks the statistical hiding. □

The following Lemma 5.2 roughly claims that whenever $t \in T$ a good approximation k of $2|X_{0,t}|$ and $2|X_{1,t}|$ up to a small constant multiplicative error can be chosen with probability $1/m = 1/\mathrm{poly}(\lambda)$.

Lemma 5.2. *Let $0 < \epsilon < 1$ be a constant. Let*

$$T := \{t : (1-\epsilon)|X_{1,t}| < |X_{0,t}| < (1+\epsilon)|X_{1,t}|\}. \tag{90}$$

Let m be an integer such that $(1+\epsilon)^m \geq 2^{\ell+1}$. For any $t \in T$, there exists an integer $j \in \{0,1,2,...,m-1\}$ such that

$$k \leq 2|X_{0,t}| \leq (1+\epsilon)k \tag{91}$$

and

$$\frac{k}{1+\epsilon} \leq 2|X_{1,t}| \leq \frac{1+\epsilon}{1-\epsilon}k. \tag{92}$$

Here, $k \coloneqq \lceil (1+\epsilon)^j \rceil$.

Proof of Lemma 5.2. Let $t \in T$. Because $0 \leq (1-\epsilon)|X_{1,t}| < |X_{0,t}|$, we have $|X_{0,t}| \geq 1$. Because $2|X_{0,t}| \leq 2^{\ell+1}$, there exists an integer $j \in \{0,1,2,...,m-1\}$ such that $(1+\epsilon)^j < 2|X_{0,t}| \leq (1+\epsilon)^{j+1}$. Let us take $k = \lceil (1+\epsilon)^j \rceil$. Because $2|X_{0,t}|$ is an integer, $k \leq 2|X_{0,t}|$. Moreover, we have

$$\begin{aligned} 2|X_{0,t}| &\leq (1+\epsilon)^{j+1} & (93)\\ &= (1+\epsilon) \times (1+\epsilon)^j & (94)\\ &\leq (1+\epsilon) \times \lceil (1+\epsilon)^j \rceil & (95)\\ &= (1+\epsilon) \times k. & (96) \end{aligned}$$

In summary, we have $k \leq 2|X_{0,t}| \leq (1+\epsilon)k$. We also have $2|X_{1,t}| < \frac{2|X_{0,t}|}{1-\epsilon} \leq \frac{1+\epsilon}{1-\epsilon}k$, and $2|X_{1,t}| > \frac{2|X_{0,t}|}{1+\epsilon} \geq \frac{k}{1+\epsilon}$. □

6 Construction of IV-PoQ

In this section, we prove Theorem 1.1. That is, we construct IV-PoQ from statistically-hiding and computationally-binding classical bit commitments.

Let $(\mathcal{S}, \mathcal{R})$ be a statistically-hiding and computationally-binding classical bit commitment scheme. The first phase where the PPT verifier $\mathcal{V}_1$ and the QPT prover $\mathcal{P}$ interact is given in Algorithm 4. The second phase where the inefficient verifier $\mathcal{V}_2$ runs is given in Algorithm 5.

We can show the completeness and soundness of our IV-PoQ as follows.

Theorem 6.1 (Completeness). *Our IV-PoQ satisfies* $(\frac{7}{8}+\frac{1}{\mathrm{poly}(\lambda)})$*-completeness.*

Theorem 6.2 (Soundness). *Our IV-PoQ satisfies* $\frac{7}{8}$*-strong-soundness, which in particular implies* $\frac{7}{8}$*-soundness.*

The above only gives an inverse-polynomial completeness-soundness gap, but we can amplify the gap to 1 by sequential repetition by Theorem 4.10. Theorem 6.1 is shown in Section 6.1. Theorem 6.2 is shown in Section 6.2. By combining Theorem 6.1 and Theorem 6.2, we obtain Theorem 1.1.

6.1 Completeness

In this subsection, we show $(\frac{7}{8}+\frac{1}{\mathrm{poly}(\lambda)})$-completeness.

Proof of Theorem 6.1. Let p_{good} be the probability that Equation (97) in the Item 4 of Algorithm 4 is in the form of $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$. Then

$$p_{\mathsf{good}} \geq (1-\mathsf{negl}(\lambda))\frac{0.1}{m} \tag{102}$$

because of the following reasons.

- In Step 1 of Algorithm 4, the probability that the transcript t such that $t \in T$ is obtained is at least $1-\mathsf{negl}(\lambda)$ from Lemma 5.1 where T is defined in Lemma 5.1.
- Given $t \in T$, in Step 3 of Algorithm 4, the probability that $\mathcal{V}_1$ chooses j such that k satisfies Equation (91) and Equation (92) of Lemma 5.2 is $\frac{1}{m}$.
- Given that k satisfies Equation (91) and Equation (92) of Lemma 5.2, in Step 4 of Algorithm 4, the probability that y such that $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is obtained is at least 0.1 from Lemma 6.3 shown below.

Algorithm 4 The first phase by $\mathcal{V}_1$ and $\mathcal{P}$

1. The PPT verifier $\mathcal{V}_1$ and the QPT prover $\mathcal{P}$ coherently execute the commit phase of the classical bit commitment scheme $(\mathcal{S}, \mathcal{R})$. (See Algorithm 3.) $\mathcal{V}_1$ plays the role of the receiver $\mathcal{R}$. $\mathcal{P}$ plays the role of the sender $\mathcal{S}$. Let t be the transcript obtained in the execution.

2. $\mathcal{P}$ has the state $\sum_{b\in\{0,1\}} \sum_{x\in X_{b,t}} |b\rangle |x\rangle$.

3. Let $0 < \epsilon < 1$ be a small constant. (We set $\epsilon := \frac{1}{100}$ for clarity.) Let m be an integer such that $(1+\epsilon)^m \geq 2^{\ell+1}$. (Such an m can be computed in $\mathrm{poly}(\lambda)$ time. In fact, we have only to take the minimum integer m such that $m \geq \frac{\ell+1}{\log_2(1+\epsilon)}$.) $\mathcal{V}_1$ chooses $j \leftarrow \{0,1,2,...,m-1\}$. Define $k := \lceil (1+\epsilon)^j \rceil$. Let $\mathcal{H} := \{h : \mathcal{X} \to \mathcal{Y}\}$ be a pairwise-independent hash family with $\mathcal{X} := \{0,1\}^\ell$ and $\mathcal{Y} := [k]$. $\mathcal{V}_1$ chooses $h_0, h_1 \leftarrow \mathcal{H}$, and sends (h_0, h_1) to $\mathcal{P}$.

4. $\mathcal{P}$ changes its state into $\sum_{b\in\{0,1\}} \sum_{x\in X_{b,t}} |b\rangle |x\rangle |h_b(x)\rangle$, and measures the third register in the computational basis to obtain the result $y \in [k]$. $\mathcal{P}$ sends y to $\mathcal{V}_1$. The post-measurement state is
$$\sum_{b\in\{0,1\}} \sum_{x\in X_{b,t}\cap h_b^{-1}(y)} |b\rangle |x\rangle . \tag{97}$$
(If there is only a single x_b such that $x_b \in X_{b,t} \cap h_b^{-1}(y)$ for each $b \in \{0,1\}$, Equation (97) is $|0\rangle |x_0\rangle + |1\rangle |x_1\rangle$. We will show later that it occurs with a non-negligible probability.)

5. From now on, $\mathcal{V}_1$ and $\mathcal{P}$ run the protocol of [KMCVY22]. $\mathcal{V}_1$ chooses $v_1 \leftarrow \{0,1\}$. $\mathcal{V}_1$ chooses $\xi \leftarrow \{0,1\}^\ell$. $\mathcal{V}_1$ sends v_1 and ξ to $\mathcal{P}$.

6.
   - If $v_1 = 0$: $\mathcal{P}$ measures all qubits of the state of Equation (97) in the computational basis, and sends the measurement result $(b', x') \in \{0,1\} \times \{0,1\}^\ell$ to $\mathcal{V}_1$. $\mathcal{V}_1$ halts.
   - If $v_1 = 1$: $\mathcal{P}$ changes the state of Equation (97) into
   $$\sum_{b\in\{0,1\}} \sum_{x\in X_{b,t}\cap h_b^{-1}(y)} |b \oplus (\xi \cdot x)\rangle |x\rangle , \tag{98}$$
   measures its second register in the Hadamard basis to obtain the measurement result $d \in \{0,1\}^\ell$, and sends d to $\mathcal{V}_1$. The post-measurement state is
   $$\sum_{b\in\{0,1\}} \sum_{x\in X_{b,t}\cap h_b^{-1}(y)} (-1)^{d\cdot x} |b \oplus (\xi \cdot x)\rangle . \tag{99}$$
   (If there is only a single x_b such that $x_b \in X_{b,t} \cap h_b^{-1}(y)$ for each $b \in \{0,1\}$, Equation (98) is $|\xi \cdot x_0\rangle |x_0\rangle + |1 \oplus (\xi \cdot x_1)\rangle |x_1\rangle$, and Equation (99) is $|\xi \cdot x_0\rangle + (-1)^{d\cdot(x_0 \oplus x_1)} |1 \oplus (\xi \cdot x_1)\rangle$.)

7. $\mathcal{V}_1$ chooses $v_2 \leftarrow \{0,1\}$. $\mathcal{V}_1$ sends v_2 to $\mathcal{P}$.

8. If $v_2 = 0$, $\mathcal{P}$ measures Equation (99) in the basis $\left\{\cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle, \sin\frac{\pi}{8}|0\rangle - \cos\frac{\pi}{8}|1\rangle\right\}$. If $v_2 = 1$, $\mathcal{P}$ measures Equation (99) in the basis $\left\{\cos\frac{\pi}{8}|0\rangle - \sin\frac{\pi}{8}|1\rangle, \sin\frac{\pi}{8}|0\rangle + \cos\frac{\pi}{8}|1\rangle\right\}$. Let $\eta \in \{0,1\}$ be the measurement result. (For the measurement in the basis $\{|\phi\rangle, |\phi^\perp\rangle\}$, the result 0 corresponds to $|\phi\rangle$ and the result 1 corresponds to $|\phi^\perp\rangle$.) $\mathcal{P}$ sends η to $\mathcal{V}_1$.

Algorithm 5 The second phase by $\mathcal{V}_2$

1. $\mathcal{V}_2$ takes $(t, h_0, h_1, y, v_1 = 0, \xi, b', x')$ or $(t, h_0, h_1, y, v_1 = 1, \xi, d, v_2, \eta)$ as input.

2. If $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied, $\mathcal{V}_2$ outputs $\top$ with probability $7/8$ and outputs $\perp$ with probability $1/8$. Then $\mathcal{V}_2$ halts. If $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is satisfied, $\mathcal{V}_2$ computes (x_0, x_1) and goes to the next step. Here, x_b is the single element of $X_{b,t} \cap h_b^{-1}(y)$ for each $b \in \{0,1\}$.

3. If $v_1 = 0$ and $x' = x_{b'}$, $\mathcal{V}_2$ outputs $\top$ and halts. Otherwise, $\mathcal{V}_2$ outputs $\perp$ and halts.

4. If $v_1 = 1$, $\mathcal{V}_2$ outputs $\top$ if

$$(\xi \cdot x_0 \neq \xi \cdot x_1) \wedge (\eta = \xi \cdot x_0), \tag{100}$$

or

$$(\xi \cdot x_0 = \xi \cdot x_1) \wedge (\eta = v_2 \oplus d \cdot (x_0 \oplus x_1)). \tag{101}$$

Otherwise, $\mathcal{V}_2$ outputs $\perp$.

Moreover, if Equation (97) in Step 4 of Algorithm 4 is in the form of $|0\rangle |x_0\rangle + |1\rangle |x_1\rangle$, the probability that $\mathcal{V}_2$ outputs $\top$ in Algorithm 5 is $\frac{1}{2} + \frac{1}{2}\cos^2\frac{\pi}{8} \geq 0.9$ as is shown in Appendix C.

Therefore, the probability that $\mathcal{V}_2$ outputs $\top$ is

$$\begin{aligned} &\Big(\frac{1}{2} + \frac{1}{2}\cos^2\frac{\pi}{8}\Big)p_{\mathsf{good}} + \frac{7}{8}(1 - p_{\mathsf{good}}) && (103)\\ &= \frac{7}{8} + \Big(\frac{1}{2} + \frac{1}{2}\cos^2\frac{\pi}{8} - \frac{7}{8}\Big)p_{\mathsf{good}} && (104)\\ &\geq \frac{7}{8} + \frac{1}{\mathrm{poly}(\lambda)}, && (105)\end{aligned}$$

which shows the completeness. □

Lemma 6.3. *Assume that k satisfies Equation (91) and Equation (92) of Lemma 5.2. In Step 4 of Algorithm 4, the probability that y such that $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is obtained is at least 0.1.*

Proof of Lemma 6.3. By using Lemma 3.2 with $S = X_{b,t}$, $h = h_b$, and $\mathcal{Y} = [k]$, we have, for any $b \in \{0,1\}$, t, and $y \in \mathcal{Y}$,

$$\begin{aligned} \Pr_{h_b \leftarrow \mathcal{H}}[|X_{b,t} \cap h_b^{-1}(y)| = 1] &\geq \frac{|X_{b,t}|}{k} - \frac{|X_{b,t}|^2}{k^2} && (106)\\ &\geq \frac{1}{2(1+\epsilon)} - \frac{(1+\epsilon)^2}{4(1-\epsilon)^2}. && (107)\end{aligned}$$

Here, in the last inequality, we have used Lemma 5.2.

In Step 4 of Algorithm 4, the probability that y is obtained is

$$\frac{|X_{0,t} \cap h_0^{-1}(y)| + |X_{1,t} \cap h_1^{-1}(y)|}{|X_{0,t}| + |X_{1,t}|}. \tag{108}$$

Let us define

$$G_{b,t,h_b} := \{y \in [k] : |X_{b,t} \cap h_b^{-1}(y)| = 1\}. \tag{109}$$

Then, the probability that we obtain y such that $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is

$$\begin{aligned}
&\mathop{\mathbb{E}}_{h_0,h_1\leftarrow\mathcal{H}}\left[\sum_{y\in G_{0,t,h_0}\cap G_{1,t,h_1}} \frac{2}{|X_{0,t}|+|X_{1,t}|}\right] && (110)\\
&= \mathop{\mathbb{E}}_{h_0,h_1\leftarrow\mathcal{H}}\left[\frac{2|G_{0,t,h_0}\cap G_{1,t,h_1}|}{|X_{0,t}|+|X_{1,t}|}\right] && (111)\\
&\geq \mathop{\mathbb{E}}_{h_0,h_1\leftarrow\mathcal{H}}\left[\frac{2|G_{0,t,h_0}\cap G_{1,t,h_1}|}{\frac{(1+\epsilon)k}{1-\epsilon}}\right] && (112)\\
&= \frac{2(1-\epsilon)}{1+\epsilon}\mathop{\mathbb{E}}_{h_0,h_1\leftarrow\mathcal{H}}\left[\frac{|G_{0,t,h_0}\cap G_{1,t,h_1}|}{k}\right] && (113)\\
&= \frac{2(1-\epsilon)}{1+\epsilon}\frac{1}{|\mathcal{H}|^2}\sum_{h_0,h_1\in\mathcal{H}}\frac{1}{k}\sum_{y\in[k]}\delta_{y\in G_{0,t,h_0}}\delta_{y\in G_{1,t,h_1}} && (114)\\
&= \frac{2(1-\epsilon)}{1+\epsilon}\frac{1}{k}\sum_{y\in[k]}\left(\frac{1}{|\mathcal{H}|}\sum_{h_0\in\mathcal{H}}\delta_{y\in G_{0,t,h_0}}\right)\left(\frac{1}{|\mathcal{H}|}\sum_{h_1\in\mathcal{H}}\delta_{y\in G_{1,t,h_1}}\right) && (115)\\
&= \frac{2(1-\epsilon)}{1+\epsilon}\frac{1}{k}\sum_{y\in[k]}\left(\Pr_{h_0\leftarrow\mathcal{H}}[y\in G_{0,t,h_0}]\right)\left(\Pr_{h_1\leftarrow\mathcal{H}}[y\in G_{1,t,h_1}]\right) && (116)\\
&\geq \frac{2(1-\epsilon)}{1+\epsilon}\frac{1}{k}\sum_{y\in[k]}\left[\frac{1}{2(1+\epsilon)}-\frac{(1+\epsilon)^2}{4(1-\epsilon)^2}\right]^2 && (117)\\
&= \frac{2(1-\epsilon)}{1+\epsilon}\left[\frac{1}{2(1+\epsilon)}-\frac{(1+\epsilon)^2}{4(1-\epsilon)^2}\right]^2 && (118)\\
&> 0.1. && (119)
\end{aligned}$$

Here, δ_α is 1 if the statement α is true, and is 0 if not. In Equation (112), we have used Lemma 5.2, and in Equation (117), we have used Equation (107). In the last inequality, we have taken $\epsilon = \frac{1}{100}$.

□

6.2 Soundness

In this subsection, we show $\frac{7}{8}$-strong-soundness.

Proof of Theorem 6.2. Our goal is to prove that for any PPT malicious prover $\mathcal{P}^*$ and any polynomial p,

$$\Pr_{r\leftarrow\mathcal{R}}\left[\Pr[\top\leftarrow\mathcal{V}_2(I):I\leftarrow\langle\mathcal{P}_r^*(1^\lambda),\mathcal{V}_1(1^\lambda)\rangle]\geq\frac{7}{8}+\frac{1}{p(\lambda)}\right]\leq\frac{1}{p(\lambda)} \tag{120}$$

for sufficiently large λ where $\mathcal{R}$ is the randomness space for $\mathcal{P}^*$ and $\mathcal{P}_r^*$ is $\mathcal{P}^*$ with the fixed randomness r.

Toward contradiction, suppose that there are a PPT prover $\mathcal{P}^*$ and a polynomial p such that

$$\Pr_{r\leftarrow\mathcal{R}}\left[\Pr[\top\leftarrow\mathcal{V}_2(I):I\leftarrow\langle\mathcal{P}_r^*(1^\lambda),\mathcal{V}_1(1^\lambda)\rangle]\geq\frac{7}{8}+\frac{1}{p(\lambda)}\right]>\frac{1}{p(\lambda)} \tag{121}$$

for infinitely many λ. Then we prove the following lemma.

Lemma 6.4. *There is an oracle-aided PPT algorithm $\mathcal{B}$ that breaks the computational binding property of the commitment scheme if it is given black-box access to $\mathcal{P}_r^*$ such that*

$$\Pr[\top\leftarrow\mathcal{V}_2(I):I\leftarrow\langle\mathcal{P}_r^*(1^\lambda),\mathcal{V}_1(1^\lambda)\rangle]\geq\frac{7}{8}+\frac{1}{p(\lambda)} \tag{122}$$

for infinitely many λ.

By combining Equation (121) and Lemma 6.4, $\mathcal{B}^{\mathcal{P}_r^*}$ for random $r \leftarrow \mathcal{R}$ breaks the computational binding property, which is a contradiction. Thus, we only have to prove Lemma 6.4 for completing the proof of Theorem 6.2.

Proof of Lemma 6.4. The proof is very similar to that of [MY23], which in turn is based on [KMCVY22]. Nonetheless, there is a difference that we have to deal with the case where $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied. Thus, we provide the full proof even though we sometimes repeat the same arguments as those in [MY23] where some sentences are taken verbatim from there with notational adaptation.

We fix r and an infinite set $\Gamma \subseteq \mathbb{N}$ such that Equation (122) holds for all $\lambda \in \Gamma$. In the following, we simply write $\mathcal{P}^*$ to mean $\mathcal{P}_r^*$ and $\Pr[\top \leftarrow \mathcal{V}_2]$ to mean $\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}_r^*(1^\lambda), \mathcal{V}_1(1^\lambda) \rangle]$. We also often omit to say "for all $\lambda \in \Gamma$", but whenever we refer to some inequality where λ appears, we always mean it holds for all $\lambda \in \Gamma$.

Define

$$\mathsf{Good} := \left\{ (t, h_0, h_1, y) : \Pr[\top \leftarrow \mathcal{V}_2 \mid (t, h_0, h_1, y)] \geq \frac{7}{8} + \frac{1}{2p(\lambda)} \right\}, \tag{123}$$

where $\Pr[\top \leftarrow \mathcal{V}_2 \mid (t, h_0, h_1, y)]$ denotes $\mathcal{V}_2$'s acceptance probability conditioned on a fixed (t, h_0, h_1, y), and define

$$p_{\mathsf{Good}} := \Pr[(t, h_0, h_1, y) \in \mathsf{Good}]. \tag{124}$$

Note that we have $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ for all $(t, h_0, h_1, y) \in \mathsf{Good}$ since otherwise $\Pr[\top \leftarrow \mathcal{V}_2 \mid (t, h_0, h_1, y)] = \frac{7}{8}$. Then we have

$$\begin{aligned} \Pr[\top \leftarrow \mathcal{V}_2] &= \Pr[\top \leftarrow \mathcal{V}_2 \wedge (t, h_0, h_1, y) \in \mathsf{Good}] + \Pr[\top \leftarrow \mathcal{V}_2 \wedge (t, h_0, h_1, y) \notin \mathsf{Good}] && (125) \\ &\leq p_{\mathsf{Good}} + (1 - p_{\mathsf{Good}}) \cdot \left(\frac{7}{8} + \frac{1}{2p(\lambda)} \right). && (126) \end{aligned}$$

By Equations (122) and (126), we have

$$p_{\mathsf{Good}} \geq \frac{1}{2p(\lambda)}. \tag{127}$$

We fix $(t, h_0, h_1, y) \in \mathsf{Good}$ until Equation (137).

For $b \in \{0,1\}$, let $x_b \in \{0,1\}^\ell$ be the unique element in $X_{b,t} \cap h_b^{-1}(y)$. Note that it is well-defined since we assume $(t, h_0, h_1, y) \in \mathsf{Good}$, which implies $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$.

We define the following probabilities all of which are conditioned on the fixed value of (t, h_0, h_1, y):

p_0: The probability that $\mathcal{V}_2$ returns $\top$ conditioned on $v_1 = 0$.

p_1: The probability that $\mathcal{V}_2$ returns $\top$ conditioned on $v_1 = 1$.

$p_{1,0}$: The probability that $\mathcal{V}_2$ returns $\top$ conditioned on $v_1 = 1$ and $v_2 = 0$.

$p_{1,1}$: The probability that $\mathcal{V}_2$ returns $\top$ conditioned on $v_1 = 1$ and $v_2 = 1$.

Clearly, we have

$$\Pr[\top \leftarrow \mathcal{V}_2 | (t, h_0, h_1, y)] = \frac{p_0 + p_1}{2} \tag{128}$$

and

$$p_1 = \frac{p_{1,0} + p_{1,1}}{2}. \tag{129}$$

By $(t, h_0, h_1, y) \in \mathsf{Good}$, Equations (123) and (128), and a trivial inequality $p_0, p_1 \leq 1$, we have

$$p_0 \geq \frac{3}{4} + \frac{1}{p(\lambda)} \tag{130}$$

and

$$p_1 \geq \frac{3}{4} + \frac{1}{p(\lambda)}. \tag{131}$$

Let $\mathcal{A}$ be a classical deterministic polynomial-time algorithm that works as follows:

1. $\mathcal{A}$ takes (t, h_0, h_1, y) and $\xi \in \{0,1\}^\ell$ as input.
2. $\mathcal{A}$ runs Step 6 of $\mathcal{P}^*$ where the transcript of Step 1-4 is set to be (t, h_0, h_1, y) and the transcript of Step 5 is set to be $(v_1 = 1, \xi)$. Let $d \in \{0,1\}^\ell$ be the message sent from $\mathcal{P}^*$ to $\mathcal{V}_1$. Note that $\mathcal{P}^*$'s message is determined by the previous transcript since $\mathcal{P}^*$ is deterministic. (Recall that $\mathcal{P}^*$ is a shorthand of $\mathcal{P}^*_r$ for a fixed randomness r.)
3. $\mathcal{A}$ runs Step 8 of $\mathcal{P}^*$ where the transcript of Step 1-4 is set to be (t, h_0, h_1, y), the transcript of Step 5 is set to be $(v_1 = 1, \xi)$, the transcript of Step 6 is set to be d, and the transcript of Step 7 is set to be $v_2 = 0$. Let $\eta_{1,0}$ be the message sent from $\mathcal{P}^*$ to $\mathcal{V}_1$.
4. $\mathcal{A}$ runs Step 8 of $\mathcal{P}^*$ where the transcript of Step 1-4 is set to be (t, h_0, h_1, y), the transcript of Step 5 is set to be $(v_1 = 1, \xi)$, the transcript of Step 6 is set to be d, and the transcript of Step 7 is set to be $v_2 = 1$. Let $\eta_{1,1}$ be the message sent from $\mathcal{P}^*$ to $\mathcal{V}_1$.
5. $\mathcal{A}$ outputs $\eta_{1,0} \oplus \eta_{1,1} \oplus 1$.

By the union bound, the probability that both $(d, \eta_{1,0})$ and $(d, \eta_{1,1})$ pass the verification is at least

$$1 - (1 - p_{1,0}) - (1 - p_{1,1}) = -1 + 2p_1 \geq \frac{1}{2} + \frac{1}{p(\lambda)}, \tag{132}$$

where the equation follows from Equation (129) and the inequality follows from Equation (131). When this occurs, for each $v_2 \in \{0,1\}$, we have

$$(\xi \cdot x_0 \neq \xi \cdot x_1) \wedge (\eta_{1,v_2} = \xi \cdot x_0), \tag{133}$$

or

$$(\xi \cdot x_0 = \xi \cdot x_1) \wedge (\eta_{1,v_2} = v_2 \oplus d \cdot (x_0 \oplus x_1)). \tag{134}$$

(Remark that the same d is used for both cases of $v_2 = 0$ and $v_2 = 1$.) In particular, if $\xi \cdot x_0 \neq \xi \cdot x_1$ then $\eta_{1,0} = \eta_{1,1}$, and if $\xi \cdot x_0 = \xi \cdot x_1$ then $\eta_{1,0} = \eta_{1,1} \oplus 1$. This implies that

$$\eta_{1,0} \oplus \eta_{1,1} \oplus 1 = \xi \cdot (x_0 \oplus x_1). \tag{135}$$

Therefore, we have

$$\Pr_{\xi \leftarrow \{0,1\}^\ell}[\mathcal{A}((t, h_0, h_1, y), \xi) = \xi \cdot (x_0 \oplus x_1)] \geq \frac{1}{2} + \frac{1}{p(\lambda)}. \tag{136}$$

Thus, by the Goldreich-Levin theorem [GL89], there is a PPT algorithm $\mathcal{E}$ such that

$$\Pr[\mathcal{E}(t, h_0, h_1, y) = x_0 \oplus x_1] \geq \frac{1}{p'(\lambda)} \tag{137}$$

for some polynomial p'. (Remark that what we have shown so far is that the above holds for any fixed $(t, h_0, h_1, y) \in$ Good.)

Then, we construct a PPT algorithm $\mathcal{B}$ that breaks the computational binding property of the classical bit commitment scheme as follows:

1. $\mathcal{B}$ interacts with the receiver $\mathcal{R}$ in the same way as $\mathcal{P}^*$ does in Step 1 of Algorithm 4, and let t be the transcript obtained from the execution.

2. $\mathcal{B}$ chooses hash functions h_0 and h_1 as in Step 3 of Algorithm 4, and send them to $\mathcal{P}^*$.

3. $\mathcal{P}^*$ returns y as a message of Step 4 in Algorithm 4. At this point, (x_0, x_1) is implicitly determined if $(t, h_0, h_1, y) \in \mathsf{Good}$.

4. $\mathcal{B}$ sends $v_1 = 0$ and $\xi \leftarrow \{0,1\}^\ell$ to $\mathcal{P}^*$ as a message of Step 5 in Algorithm 4.

5. $\mathcal{P}^*$ returns (b', x') as a message of the first case of Step 6 in Algorithm 4.

6. $\mathcal{B}$ runs $\mathcal{E}(t, h_0, h_1, y)$ and let z be the output.

7. $\mathcal{B}$ sets $x'_0 := x'$ and $x'_1 := x' \oplus z$ if $b' = 0$, and $x'_0 := x' \oplus z$ and $x'_1 := x'$ otherwise. For each $b \in \{0,1\}$, $\mathcal{B}$ generate a decommitment dcom_b corresponding to the sender's randomness x'_b and transcript t. $\mathcal{B}$ outputs $(0, \mathsf{decom}_0)$ and $(1, \mathsf{decom}_1)$.

Recall that we have shown that for any $(t, h_0, h_1, y) \in \mathsf{Good}$, Equations (130) and (137) hold. Thus, for any $(t, h_0, h_1, y) \in \mathsf{Good}$, we have

$$\Pr[x' = x_{b'} | (t, h_0, h_1, y)] \geq \frac{3}{4} + \frac{1}{p(\lambda)} \tag{138}$$

and

$$\Pr[z = x_0 \oplus x_1 | (t, h_0, h_1, y)] \geq \frac{1}{p'(\lambda)}. \tag{139}$$

Moreover, the two events $x' = x_{b'}$ and $z = x_0 \oplus x_1$ are independent once we fix (t, h_0, h_1, y). Therefore, for any $(t, h_0, h_1, y) \in \mathsf{Good}$, we have

$$\Pr[x' = x_{b'} \wedge z = x_0 \oplus x_1 | (t, h_0, h_1, y)] \geq \frac{3}{4p'(\lambda)}. \tag{140}$$

Combined with Equation (127), we have

$$\Pr[(t, h_0, h_1, y) \in \mathsf{Good} \wedge x'_0 = x_0 \wedge x'_1 = x_1] \geq \frac{3}{8p(\lambda)p'(\lambda)}. \tag{141}$$

By the definition of x_b, we have $x_b \in X_{b,t}$ for $b \in \{0,1\}$. Thus, by the perfect correctness of the commitment scheme, decom_b derived from (x_b, t) is a valid decommitment. Thus, Equation (141) implies that $\mathcal{B}$ outputs valid decommitments for both messages 0 and 1 with probability at least $\frac{3}{8p(\lambda)p'(\lambda)}$ (for all $\lambda \in \Gamma$). This completes the proof of Lemma 6.4. $\square$

This completes the proof of Theorem 6.2. $\square$

6.3 Computational Power of the Inefficient Verifier

In this subsection, we show that $\mathcal{V}_2$ can be a classical deterministic polynomial-time algorithm querying an $\mathbf{NP}$ oracle. (The inefficient verifier in our construction actually uses randomness when $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied, but the inefficient verifier can be deterministic if we let the first phase verifier append the randomness to the transcript.)

The inefficient parts of $\mathcal{V}_2$ are verifying $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ and finding (x_0, x_1), where x_b is the single element of $X_{b,t} \cap h_b^{-1}(y)$. We show that these two tasks can be done in classical deterministic polynomial-time querying an $\mathbf{NP}$ oracle. Note that the membership of $x \in X_{b,t} \cap h_b^{-1}(y)$ can be decided in a classical deterministic polynomial time. Therefore, the decision problem

Yes: There exists $x \in \{0,1\}^\ell$ such that $x \in X_{b,t} \cap h_b^{-1}(y)$.

No: For any $x \in \{0,1\}^\ell$, $x \notin X_{b,t} \cap h_b^{-1}(y)$.

is in $\mathbf{NP}$.

First, $\mathcal{V}_2$ queries the above decision problem to the $\mathbf{NP}$ oracle for each $b \in \{0,1\}$. If the answer is no for a $b \in \{0,1\}$, it means that $|X_{b,t} \cap h_b^{-1}(y)| = 0$. In that case, $\mathcal{V}_2$ concludes that $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied. If the answer is yes for both $b \in \{0,1\}$, $|X_{0,t} \cap h_0^{-1}(y)| \geq 1$ and $|X_{1,t} \cap h_1^{-1}(y)| \geq 1$ are guaranteed.

Then, $\mathcal{V}_2$ finds an element $x_b \in X_{b,t} \cap h_b^{-1}(y)$ for each $b \in \{0,1\}$. Finding such an element is just an $\mathbf{NP}$ search problem, which can be solved in classical deterministic polynomial-time by querying the $\mathbf{NP}$ oracle polynomially many times.

$\mathcal{V}_2$ finally queries the following decision problem to the $\mathbf{NP}$ oracle for each $b \in \{0,1\}$.

Yes: There exists $x \in \{0,1\}^\ell$ such that $x \in (X_{b,t} \cap h_b^{-1}(y)) \setminus \{x_b\}$.

No: For any $x \in \{0,1\}^\ell$, $x \notin (X_{b,t} \cap h_b^{-1}(y)) \setminus \{x_b\}$.

If the answer is yes for a $b \in \{0,1\}$, it means that $|X_{b,t} \cap h_b^{-1}(y)| \geq 2$. In that case, $\mathcal{V}_2$ concludes that $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is not satisfied. If the answer is no for both $b \in \{0,1\}$, $\mathcal{V}_2$ concludes that $|X_{0,t} \cap h_0^{-1}(y)| = |X_{1,t} \cap h_1^{-1}(y)| = 1$ is satisfied.

7 Implausibility of Two-Round AI-IV-PoQ

In this section, we prove Theorems 1.8 and 1.9.

7.1 Impossibility of Classical Reduction

In this subsection, we formally state Theorem 1.8 and prove it.

First, we define game-based assumptions. The definition is identical to *falsifiable assumptions* in [GW11] except for the important difference that the challenger can be unbounded-time.

Definition 7.1 (Game-based assumptions). *A game-based assumption consists of a possibly unbounded-time interactive machine $\mathcal{C}$ (the challenger) and a constant $t \in [0,1)$. On the security parameter 1^λ, the challenger $\mathcal{C}(1^\lambda)$ interacts with a classical or quantum machine $\mathcal{A}$ (the adversary) over a classical channel and finally outputs a bit b. We denote this execution by $b \leftarrow \langle \mathcal{A}(1^\lambda), \mathcal{C}(1^\lambda) \rangle$.*

We say that a game-based assumption $(\mathcal{C}, t)$ holds against classical (resp. quantum) adversaries if for any PPT (resp. QPT) adversary $\mathcal{A}$, $|\Pr[1 \leftarrow \langle \mathcal{A}(1^\lambda), \mathcal{C}(1^\lambda) \rangle] - t| \leq \mathsf{negl}(\lambda)$.

Remark 7.2 (Examples). As explained in [GW11], the above definition captures a very wide class of assumptions used in cryptography even if we restrict the challenger to be PPT. They include (but not limited to) general assumptions such as security of OWFs, public key encryption, digital signatures, oblivious transfers etc. as well as concrete assumptions such as the hardness of factoring, discrete-logarithm, LWE etc. In addition, since we allow the challenger to be unbounded-time, it also captures some non-falsifiable assumptions such as hardness of indistinguishability obfuscation [BGI+12, GGH+16] or succinct arguments [Mic00] etc. Examples of assumptions that are not captured by the above definition include so called knowledge assumptions [Dam92, CD08, CD09, BCCT12] and zero-knowledge proofs with non-black-box zero-knowledge [HT98, Bar01].

We clarify meanings of several terms used in the statement of our theorem.

Definition 7.3 (Classical oracle-access to a cheating prover). *Let $\Pi = (\mathcal{P}, \mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2))$ be a two-round AI-IV-PoQ. We say that a (possibly unbounded-time stateless) randomized classical machine $\mathcal{P}^*$ breaks s-soundness of Π if there is a polynomial* poly *such that* $\Pr[\top \leftarrow \mathcal{V}_2(I) : I \leftarrow \langle \mathcal{P}^*(\sigma), \mathcal{V}_1(\sigma) \rangle] \geq s(|\sigma|) + 1/\mathrm{poly}(|\sigma|)$ *for all but finitely many $\sigma \in \Sigma$. We say that an oracle-aided classical machine $\mathcal{R}$ is given oracle access to $\mathcal{P}^*$ if it can query an auxiliary input σ and the first-round message m_1 of Π and the oracle returns the second-round message m_2 generated by $\mathcal{P}^*$ with a fresh randomness r in each invocation.*

Remark 7.4. Since we consider two-round protocols, we can assume that $\mathcal{P}^*$ is stateless without loss of generality.

Then we state the formal version of Theorem 1.8.

Theorem 7.5. *Let Π be a two-round AI-IV-PoQ that satisfies c-completeness and s-soundness where $c(\lambda) - s(\lambda) \geq 1/\mathrm{poly}(\lambda)$ and let $(\mathcal{C}, t)$ be a game-based assumption. Suppose that there is an oracle-aided PPT machine $\mathcal{R}$ (the reduction algorithm) such that for any (possibly unbounded-time stateless) randomized classical machine $\mathcal{P}^*$ that breaks s-soundness of Π, $|\Pr[1 \leftarrow \langle \mathcal{R}^{\mathcal{P}^*}(1^\lambda), \mathcal{C}(1^\lambda)\rangle] - t|$ is non-negligible. Then the game-based assumption $(\mathcal{C}, t)$ does not hold against quantum adversaries.*

Proof. Let $\widetilde{\mathcal{P}}^*$ be an unbounded-time randomized classical machine that simulates the honest QPT prover $\mathcal{P}$. Then $\widetilde{\mathcal{P}}^*$ breaks s-soundness of Π because of c-soundness and $c(\lambda) - s(\lambda) \geq 1/\mathrm{poly}(\lambda)$. Thus, $|\Pr[1 \leftarrow \langle \mathcal{R}^{\widetilde{\mathcal{P}}^*}(1^\lambda), \mathcal{C}(1^\lambda)\rangle] - t|$ is non-negligible. Since $\widetilde{\mathcal{P}}^*$ is simulatable by a QPT machine, $\mathcal{R}^{\widetilde{\mathcal{P}}^*}$ is simulatable by a QPT machine. Thus, $(\mathcal{C}, t)$ does not hold against quantum adversaries. □

Remark 7.6. One might think that a similar proof extends to the case of quantum reductions. However, we believe that this is non-trivial. For example, suppose that a quantum reduction algorithm $\mathcal{R}$ queries a uniform superposition $\sum_{m_1} |m_1\rangle$ to the oracle where we omit an auxiliary input for simplicity. If the oracle is a classical randomized cheating prover $\mathcal{P}^*$, then it should return a state of the form $\sum_{m_1} |m_1\rangle |\mathcal{P}^*(m_1; r)\rangle$ for a randomly chosen r where $\mathcal{P}^*(m_1; r)$ is the second message m_2 sent by $\mathcal{P}^*$ given the first-round message m_1 and randomness r. On the other hand, if we try to simulate the oracle by using the honest QPT prover $\mathcal{P}$, then the resulting state is of the form $\sum_{m_1,m_2} |m_1\rangle |m_2\rangle |garbage_{m_1,m_2}\rangle$. Due to potential entanglement between the first two registers and the third register, this does not correctly simulate the situation with a classical prover.

7.2 Oracle Separation

In this subsection, we formally state Theorem 1.9 and prove it.

First, we define cryptographic primitives. The following definition is taken verbatim from [RTV04] except for the difference that we consider quantum security. We remark that we restrict primitives themselves to be classical and only allow the adversary (the machine M) to be quantum.

Definition 7.7 (Cryptographic primitives; quantumly-secure version of [RTV04, Definition 2.1]). *A primitive $\mathtt{P}$ is a pair $(F_{\mathtt{P}}, R_{\mathtt{P}})$, where $F_{\mathtt{P}}$ is a set of functions $f : \{0,1\}^* \to \{0,1\}^*$, and $R_{\mathtt{P}}$ is a relation over pairs (f, M) of a function $f \in F_{\mathtt{P}}$ and an interactive quantum machine M. The set $F_{\mathtt{P}}$ is required to contain at least one function which is computable by a PPT machine.*

A function $f : \{0,1\}^* \to \{0,1\}^*$ implements $\mathtt{P}$ or is an implementation of $\mathtt{P}$ if $f \in F_{\mathtt{P}}$. An efficient implementation of $\mathtt{P}$ is an implementation of $\mathtt{P}$ which is computable by a PPT machine. A machine M $\mathtt{P}$-breaks $f \in F_{\mathtt{P}}$ if $(f, M) \in R_{\mathtt{P}}$. A quantumly-secure implementation of $\mathtt{P}$ is an implementation of $\mathtt{P}$ such that no QPT machine $\mathtt{P}$-breaks f.

It was pointed out in [BBF13] that the above definition was too general and there are subtle logical gaps and counter examples in their claims. In particular, [RTV04] implicitly assumes that if two machines M and M' behave identically, then $(f, M) \in R_{\mathtt{P}}$ and $(f, M') \in R_{\mathtt{P}}$ are equivalent. We formalize this property following [BBF13] where the definitions are taken verbatim from there except for adaptation to the quantumly-secure setting.

Definition 7.8 (Output distribution [BBF13, Definition B.1 in the ePrint version]). *An interactive (oracle-aided) quantum Turing machine M together with its oracle defines an output distribution, namely, each fixed finite sequence of inputs fed to M induces a distribution on the output sequences by considering all random choices of M and its oracle. The output distribution of M is defined to be the set of these distributions, indexed by the finite sequences of input values.*

Definition 7.9 (Semantical cryptographic primitive [BBF13, Definition B.2 in the ePrint version]). *A cryptographic primitive $\mathtt{P} = (F_{\mathtt{P}}, R_{\mathtt{P}})$ is called semantical, if for all $f \in F_{\mathtt{P}}$ and all interactive (oracle-aided) quantum Turing machines M and M' (including their oracles), it holds: If M induces the same output distribution as M', then $(f, M) \in R_{\mathtt{P}}$ if and only if $(f, M') \in R_{\mathtt{P}}$.*

Remark 7.10 (Examples). As explained in [RTV04, BBF13], most cryptographic primitives considered in the literature are captured by semantical cryptographic primitives. They include (but not limited to) OWFs, public key encryption, digital signatures, oblivious transfers indistinguishability obfuscation etc. On the other hand, we note that it does not capture concrete assumptions such as the hardness of factoring, discrete-logarithm, LWE etc. unlike game-based assumptions defined in Definition 7.1. Similarly to game-based assumptions, semantical cryptographic primitives do not capture knowledge-type assumptions or zero-knowledge proofs (with non-black-box zero-knowledge).

Next, we define secure implementation relative to oracles following [RTV04].

Definition 7.11 (Secure implementation relative to oracles [RTV04, Definition 2.2]). *A quantumly-secure implementation of primitive* $\mathtt{P}$ *exists relative to an oracle* O *if there exists an implementation of* f *of* $\mathtt{P}$ *which is computable by a PPT oracle machine with access to* O *and such that no QPT oracle machine with access to* O $\mathtt{P}$*-breaks* f*.*

Remark 7.12 (Example). A quantumly-secure implementation of OWFs and collision-resistant hash functions exists relative to a random oracle [BBBV97, Zha15]. [HY20] implicitly proves that quantumly-secure implementation of trapdoor-permutations exist relative to a classical oracle. We believe that we can prove similar statements for most cryptographic primitives by appropriately defining oracles.

Now, we are ready to state the formal version of Theorem 1.9.

Theorem 7.13. *Suppose that a semantical cryptographic primitive* $\mathtt{P} = (F_{\mathtt{P}}, R_{\mathtt{P}})$ *has a quantumly-secure implementation relative to a classical oracle. Then there is a randomized classical oracle relative to which two-round AI-IV-PoQ do not exist but a quantumly-secure implementation of* $\mathtt{P}$ *exists.*

Remark 7.14. If we assume that a quantumly-secure implementation of a semantical cryptographic primitive $\mathtt{P}$ exists in the unrelativized world, then the assumption of the theorem is trivially satisfied relative to a trivial oracle that does nothing. Thus, the above theorem can be understood as a negative result on constructing two-round AI-IV-PoQ from any primitive whose quantumly-secure implementation is believed to exist.

Proof. Let f be a quantumly-secure implementation of $\mathtt{P}$ relative to a classical oracle O. Let Q^O be a randomized oracle that takes a description of an n-qubit input quantum circuit C^O with O-gates and a classical string $x \in \{0,1\}^n$ as input and returns a classical string according to the distribution of $C^O(x)$. We prove that AI-IV-PoQ do not exist but f is a quantumly-secure implementation relative to oracles (O, C^O).

Let $\Pi = (\mathcal{P}, \mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2))$ be an AI-IV-PoQ that satisfies s-soundness relative to (O, Q^O). Since Q^O is simulatable in QPT with oracle access to O, given the auxiliary input σ, one can generate a description of quantum circuit C^O with O-gates that simulates $\mathcal{P}^{O,Q^O}(\sigma)$ in a classical polynomial time. Then let us consider a classical cheating prover $\mathcal{P}^*$ relative to the oracles (O, R^O) that works as follows: Receiving the auxiliary input σ and the first-round message m_1 from the external verifier, $\mathcal{P}^*$ generates the above quantum circuit C^O, queries (C^O, m_1) to the oracle Q^O to receive the response m_2, and send m_2 to the external verifier. Clearly, $\mathcal{P}^*$ passes the verification with the same probability as $\mathcal{P}$ does. Therefore, Π cannot satisfy c-soundness for any $c < s$. This means that there is no AI-IV-PoQ relative to (O, Q^O).

On the other hand, if f is not a quantumly-secure implementation of $\mathtt{P}$ relative to (O, R^O), then there is a QPT oracle-aided machine M such that $(f, M^{O,Q^O}) \in R_{\mathtt{P}}$. Again, since Q^O is simulatable in QPT with oracle access to O, there is a QPT oracle-aided machine $\widetilde{M}$ such that M^{O,Q^O} and $\widetilde{M}^O$ induce the same output distributions. Since $\mathtt{P}$ is semantical, we have $(f, \widetilde{M}^O) \in R_{\mathtt{P}}$. This contradicts the assumption that f is a quantumly-secure implementation of $\mathtt{P}$ relative to O. Therefore, f is a quantumly-secure implementation of $\mathtt{P}$ relative to (O, R^O).

□

8 Variants of PoQ from QE-OWFs

Definition 8.1 (QE-OWFs). *A function* $f : \{0,1\}^* \to \{0,1\}^*$ *is a (classically-secure) quantum-evaluation OWF (QE-OWF) if the following two properties are satisfied.*

- *There exists a QPT algorithm* QEval *such that* $\Pr[f(x) \leftarrow \mathsf{QEval}(x)] \geq 1 - 2^{-|x|}$ *for all* $x \in \{0,1\}^*$*.* [23]

[23] Actually the threshold can be any value larger than 1/2, because the amplification is possible.

- *For any PPT adversary $\mathcal{A}$, there exists a negligible function* negl *such that for any λ,*

$$\Pr[f(x') = f(x) : x' \leftarrow \mathcal{A}(1^\lambda, f(x)), x \leftarrow \{0,1\}^\lambda] \leq \mathsf{negl}(\lambda). \tag{142}$$

Remark 8.2. It is usually useless to consider OWFs whose evaluation algorithm is QPT but the security is against PPT adversaries. However, for our applications, classical security is enough. We therefore consider the classically-secure QE-OWFs, because it only makes our result stronger.

Before explaining our construction of variants of PoQ from QE-OWFs, we point out that QE-OWFs seems to be weaker than classically-secure and classical-evaluation OWFs. Let g be a classically-secure and classical-evaluation OWF. Let L be any language in $\mathbf{BQP}$. From them, we construct the function f as follows: $f(x,y) := L(x)\|g(y)$, where $L(x) = 1$ if $x \in L$ and $L(x) = 0$ if $x \notin L$. Then we have the following lemma.

Lemma 8.3. *f is QE-OWFs. Moreover, if $\mathbf{BQP} \neq \mathbf{BPP}$, f cannot be evaluated in classical polynomial-time.*

Proof. First, it is clear that there exists a QPT algorithm QEval such that for any x, y

$$\Pr[f(x,y) \leftarrow \mathsf{QEval}(x,y)] \geq 1 - 2^{-|x\|y|}. \tag{143}$$

Second, let us show the one-wayness of f. Assume that it is not one-way. Then, there exists a PPT adversary $\mathcal{A}$ and a polynomial p such that

$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}\frac{1}{2^m}\sum_{y\in\{0,1\}^m}\Pr[L(x) = L(x') \wedge g(y) = g(y') : (x',y') \leftarrow \mathcal{A}(L(x)\|g(y))] \geq \frac{1}{p}. \tag{144}$$

From this $\mathcal{A}$, we can construct a PPT adversary $\mathcal{B}$ that breaks the one-wayness of g as follows.

1. On input $g(y)$, sample $x \leftarrow \{0,1\}^n$ and $b \leftarrow \{0,1\}$.
2. Run $(x', y') \leftarrow \mathcal{A}(b\|g(y))$.
3. Output y'.

The probability that $\mathcal{B}$ breaks the one-wayness of g is

$$\frac{1}{2^m}\sum_{y\in\{0,1\}^m}\frac{1}{2^n}\sum_{x\in\{0,1\}^n}\frac{1}{2}\sum_{b\in\{0,1\}}\sum_{x',y'}\Pr[(x',y') \leftarrow \mathcal{A}(b\|g(y))]\delta_{g(y),g(y')} \tag{145}$$

$$\geq \frac{1}{2}\frac{1}{2^n}\sum_{x\in\{0,1\}^n}\frac{1}{2^m}\sum_{y\in\{0,1\}^m}\sum_{x',y'}\Pr[(x',y') \leftarrow \mathcal{A}(L(x)\|g(y))]\delta_{g(y),g(y')} \tag{146}$$

$$\geq \frac{1}{2}\frac{1}{2^n}\sum_{x\in\{0,1\}^n}\frac{1}{2^m}\sum_{y\in\{0,1\}^m}\sum_{x',y'}\Pr[(x',y') \leftarrow \mathcal{A}(L(x)\|g(y))]\delta_{g(y),g(y')}\delta_{L(x),L(x')} \tag{147}$$

$$\geq \frac{1}{2p}, \tag{148}$$

which is non-negligible.

Finally, it is clear that if there exists a PPT algorithm that computes $f(x,y)$ for any x, y with probability at least $1 - 2^{-|x\|y|}$, then the algorithm can solve L, which contradicts $\mathbf{BQP} \neq \mathbf{BPP}$. $\square$

Now we show the main result of this section.

Theorem 8.4. *If (classically-secure) QE-OWFs exist, then QV-PoQ exist or (classically-secure and classical-evaluation) infinitely-often OWFs exist.*

Proof. Let f be a classically-secure QE-OWF. From the f, we construct a QV-PoQ $(\mathcal{P}, \mathcal{V})$ as follows.

1. The verifier $\mathcal{V}$ chooses $x \leftarrow \{0,1\}^\lambda$, and sends it to the prover $\mathcal{P}$.

2. $\mathcal{P}$ runs $y \leftarrow \mathsf{QEval}(x)$, and sends y to $\mathcal{V}$.

3. $\mathcal{V}$ runs $y' \leftarrow \mathsf{QEval}(x)$. If $y = y'$, $\mathcal{V}$ outputs $\top$. Otherwise, $\mathcal{V}$ outputs $\bot$.

The 1-completeness is shown as follows. The probability that $\mathcal{V}$ accepts with the honest prover is

$$\frac{1}{2^\lambda}\sum_x\sum_y \Pr[y \leftarrow \mathsf{QEval}(x)]^2 \geq \frac{1}{2^\lambda}\sum_x \Pr[f(x) \leftarrow \mathsf{QEval}(x)]^2 \tag{149}$$

$$\geq \frac{1}{2^\lambda}\sum_x (1-2^{-\lambda})^2 \tag{150}$$

$$\geq 1 - \mathsf{negl}(\lambda). \tag{151}$$

If the soundness is also satisfied, then we have a QV-PoQ.

Assume that the soundness is not satisfied. Then there exists a PPT algorithm P^* such that for any polynomial poly such that

$$\frac{1}{2^\lambda}\sum_x\sum_y \Pr[y \leftarrow P^*(x)]\Pr[y \leftarrow \mathsf{QEval}(x)] \geq 1 - \frac{1}{\mathrm{poly}(\lambda)} \tag{152}$$

for infinitely many λ. Then we have for any polynomial poly

$$1 - \frac{1}{\mathrm{poly}(\lambda)} \leq \frac{1}{2^\lambda}\sum_x\sum_y \Pr[y \leftarrow P^*(x)]\Pr[y \leftarrow \mathsf{QEval}(x)] \tag{153}$$

$$= \frac{1}{2^\lambda}\sum_x \Pr[f(x) \leftarrow P^*(x)]\Pr[f(x) \leftarrow \mathsf{QEval}(x)] \tag{154}$$

$$+ \frac{1}{2^\lambda}\sum_x\sum_{y\neq f(x)} \Pr[y \leftarrow P^*(x)]\Pr[y \leftarrow \mathsf{QEval}(x)] \tag{155}$$

$$\leq \frac{1}{2^\lambda}\sum_x \Pr[f(x) \leftarrow P^*(x)] + \frac{1}{2^\lambda}\sum_x\sum_{y\neq f(x)} \Pr[y \leftarrow \mathsf{QEval}(x)] \tag{156}$$

$$\leq \frac{1}{2^\lambda}\sum_x \Pr[f(x) \leftarrow P^*(x)] + \frac{1}{2^\lambda}\sum_x 2^{-\lambda} \tag{157}$$

for infinitely many λ, which gives that for any polynomial poly

$$\frac{1}{2^\lambda}\sum_x \Pr[f(x) \leftarrow P^*(x)] \geq 1 - \frac{1}{\mathrm{poly}(\lambda)} \tag{158}$$

for infinitely many λ. If we write the random seed for P^* explicitly, it means that for any polynomial poly

$$\frac{1}{2^{\lambda+p(\lambda)}}\sum_{x\in\{0,1\}^\lambda}\sum_{r\in\{0,1\}^{p(\lambda)}} \delta_{f(x),P^*(x;r)} \geq 1 - \frac{1}{\mathrm{poly}(\lambda)} \tag{159}$$

for infinitely many λ, where $p(\lambda)$ is the length of the seed, and $\delta_{\alpha,\beta} = 1$ if $\alpha = \beta$ and it is 0 otherwise. Define the set

$$G := \{(x,r) \in \{0,1\}^\lambda \times \{0,1\}^p : f(x) = P^*(x;r)\}. \tag{160}$$

Then, from Equation (159), we have for any polynomial poly

$$\frac{2^{\lambda+p} - |G|}{2^{\lambda+p}} \leq \frac{1}{\mathrm{poly}(\lambda)} \tag{161}$$

for infinitely many λ.

Define the function $g:(x,r)\to P^*(x;r)$. We show that it is a classically-secure and classical-evaluation infinitely-often distributionally OWF. (For the definition of distributionally OWFs, see Appendix D.) It is enough because distributionally OWFs imply OWFs (Lemma D.2). To show it, assume that it is not. Then, for any polynomial poly there exists a PPT algorithm $\mathcal{A}$ such that

$$\left\|\frac{1}{2^{\lambda+p}}\sum_{x,r}(x,r)\otimes g(x,r)-\frac{1}{2^{\lambda+p}}\sum_{x,r}\mathcal{A}(g(x,r))\otimes g(x,r)\right\|_1\le\frac{1}{\mathrm{poly}(\lambda)} \tag{162}$$

for infinitely many λ. Here, we have used quantum notations although everything is classical, because it is simpler. Moreover, for the notational simplicity, we omit bras and kets: (x,r) means $|(x,r)\rangle\langle(x,r)|$, $g(x,r)$ means $|g(x,r)\rangle\langle g(x,r)|$, and $\mathcal{A}(g(x,r))$ is the (diagonal) density matrix that represents the classical output distribution of the algorithm $\mathcal{A}$ on input $g(x,r)$.

From the algorithm $\mathcal{A}$, we construct a PPT adversary $\mathcal{B}$ that breaks the distributional one-wayness of f as follows:

1. On input $f(x)$, sample $r\leftarrow\{0,1\}^p$.
2. Run $(x',r')\leftarrow\mathcal{A}(f(x))$.
3. Output x'.

Then for any polynomial poly

$$\left\|\frac{1}{2^\lambda}\sum_x x\otimes f(x)-\frac{1}{2^\lambda}\sum_x \mathcal{B}(f(x))\otimes f(x)\right\|_1 \tag{163}$$

$$=\left\|\frac{1}{2^\lambda}\sum_x x\otimes f(x)-\frac{1}{2^\lambda}\sum_x \frac{1}{2^p}\sum_r \mathrm{Tr}_R[\mathcal{A}(f(x))]\otimes f(x)\right\|_1 \tag{164}$$

$$=\left\|\frac{1}{2^\lambda}\sum_x\sum_r \frac{1}{2^p}\mathrm{Tr}(r)x\otimes f(x)-\frac{1}{2^\lambda}\sum_x \frac{1}{2^p}\sum_r \mathrm{Tr}_R[\mathcal{A}(f(x))]\otimes f(x)\right\|_1 \tag{165}$$

$$\leq\left\|\frac{1}{2^\lambda}\sum_x\sum_r \frac{1}{2^p}x\otimes r\otimes f(x)-\frac{1}{2^\lambda}\sum_x \frac{1}{2^p}\sum_r \mathcal{A}(f(x))\otimes f(x)\right\|_1 \tag{166}$$

$$=\left\|\frac{1}{2^{\lambda+p}}\sum_{x,r}[x\otimes r\otimes f(x)-\mathcal{A}(f(x))\otimes f(x)]\right\|_1 \tag{167}$$

$$=\Bigg\|\frac{1}{2^{\lambda+p}}\sum_{(x,r)\in G}[x\otimes r\otimes f(x)-\mathcal{A}(f(x))\otimes f(x)] \tag{168}$$

$$+\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}[x\otimes r\otimes f(x)-\mathcal{A}(f(x))\otimes f(x)] \tag{169}$$

$$+\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}[x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)] \tag{170}$$

$$-\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}[x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)]\Bigg\|_1 \tag{171}$$

$$=\Bigg\|\frac{1}{2^{\lambda+p}}\sum_{(x,r)\in G}[x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)] \tag{172}$$

$$+\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}[x\otimes r\otimes f(x)-\mathcal{A}(f(x))\otimes f(x)] \tag{173}$$

$$+\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}[x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)] \tag{174}$$

$$-\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}[x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)]\Bigg\|_1 \tag{175}$$

$$=\Bigg\|\frac{1}{2^{\lambda+p}}\sum_{x,r}[x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)] \tag{176}$$

$$+\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}[x\otimes r\otimes f(x)-\mathcal{A}(f(x))\otimes f(x)] \tag{177}$$

$$-\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}[x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)]\Bigg\|_1 \tag{178}$$

$$\leq\left\|\frac{1}{2^{\lambda+p}}\sum_{x,r}[x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)]\right\|_1 \tag{179}$$

$$+\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}\left\|x\otimes r\otimes f(x)-\mathcal{A}(f(x))\otimes f(x)\right\|_1 \tag{180}$$

$$+\frac{1}{2^{\lambda+p}}\sum_{(x,r)\notin G}\left\|x\otimes r\otimes P^*(x;r)-\mathcal{A}(P^*(x;r))\otimes P^*(x;r)\right\|_1 \tag{181}$$

$$\leq\frac{1}{\mathrm{poly}(\lambda)} \tag{182}$$

for infinitely λ, which means that f is not distributional one-way. It contradicts the assumption that f is one-way, because one-wayness implies distributionally one-wayness. In Equation (164), R is the register of the state $\mathcal{A}(f(x))$ that contains "the output r" of the algorithm $\mathcal{A}$. The last inequality comes from Equation (162) and Equation (161). $\square$

Acknowledgements. We thank Mark Zhandry for pointing out Remark 7.6. TM is supported by JST Moonshot R&D JPMJMS2061-5-1-1, JST FOREST, MEXT QLEAP, the Grant-in-Aid for Scientific Research (B) No.JP19H04066, the Grant-in Aid for Transformative Research Areas (A) 21H05183, and the Grant-in-Aid for Scientific Research (A) No.22H00522.

References

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 333–342. ACM Press, June 2011. (Cited on page 3, 6.)

[AA14] Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. *Theory Comput.*, 10:133–166, 2014. (Cited on page 10.)

[AA15] Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 307–316. ACM Press, June 2015. (Cited on page 6, 10.)

[Aar08] Scott Aaronson. On perfect completeness for qma. *arXiv:0806.0450*, 2008. (Cited on page 7.)

[Aar10] Scott Aaronson. BQP and the polynomial hierarchy. In Leonard J. Schulman, editor, *42nd ACM STOC*, pages 141–150. ACM Press, June 2010. (Cited on page 3, 6, 10, 44.)

[Aar14] Scott Aaronson. The equivalence of sampling and searching. *Theory of Computing Systems*, 55:281–298, 2014. (Cited on page 3.)

[AC17] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. CCC'17: Proceedings of the 32nd Computational Complexity Conference, 2017. (Cited on page 3, 6, 11.)

[ACC+22] Atul Singh Arora, Andrea Coladangelo, Matthew Coudron, Alexandru Gheorghiu, Uttam Singh, and Hendrik Waldner. Quantum depth in the random oracle model. *arXiv:2210.06454*, 2022. (Cited on page 3, 6, 10.)

[AG19] Scott Aaronson and Sam Gunn. On the classical hardness of spoofing linear cross-entropy benchmarking. *arXiv:1910.12085*, 2019. (Cited on page 3, 6.)

[AGQY22] Prabhanjan Ananth, Aditya Gulati, Luowen Qian, and Henry Yuen. Pseudorandom (function-like) quantum state generators: New definitions and applications. TCC, 2022. (Cited on page 7.)

[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, Heidelberg, August 2022. (Cited on page 7.)

[AR16] Benny Applebaum and Pavel Raykov. On the relationship between statistical zero-knowledge and statistical randomized encodings. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 449–477. Springer, Heidelberg, August 2016. (Cited on page 5.)

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd FOCS*, pages 106–115. IEEE Computer Society Press, October 2001. (Cited on page 31.)

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. (Cited on page 33.)

[BBF13] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 296–315. Springer, Heidelberg, December 2013. (Cited on page 32.)

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Shafi Goldwasser, editor, *ITCS 2012*, pages 326–349. ACM, January 2012. (Cited on page 31.)

[BCM+21] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *Journal of the ACM*, 68(5):31:1–31:47, 2021. (Cited on page 3, 4, 6, 16.)

[BCQ23] Zvika Brakerski, Ran Canetti, and Luowen Qian. On the computational hardness needed for quantum cryptography. ITCS 2023: 14th Innovations in Theoretical Computer Science, 2023. (Cited on page 7.)

[BDRV18] Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 133–161. Springer, Heidelberg, April / May 2018. (Cited on page 4.)

[BFNV19] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15:159–163, 2019. (Cited on page 3, 6.)

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6:1–6:48, 2012. (Cited on page 31.)

[BGK18] Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362:308–311, 2018. (Cited on page 11.)

[BGKT20] Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16:1040–1045, 2020. (Cited on page 11.)

[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, Heidelberg, May 2019. (Cited on page 4, 13.)

[BJS11] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467:459–472, 2011. (Cited on page 3, 6.)

[BKVV20] Zvika Brakerski, Venkata Koppula, Umesh Vazirani, and Thomas Vidick. Simpler proofs of quantumness. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia*, volume 158 of *LIPIcs*, pages 8:1–8:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. (Cited on page 7, 10.)

[BMS16] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Physical Review Letters*, 117:080501, 2016. (Cited on page 3, 6.)

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 449–460. Springer, Heidelberg, July 2008. (Cited on page 31.)

[CD09] Ran Canetti and Ronny Ramzi Dakdouk. Towards a theory of extractable functions. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 595–613. Springer, Heidelberg, March 2009. (Cited on page 31.)

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004. (Cited on page 4.)

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 17–33. Springer, Heidelberg, February 2005. (Cited on page 6.)

[CX22] Shujiao Cao and Rui Xue. On constructing one-way quantum state generators, and more. Cryptology ePrint Archive, Report 2022/1323, 2022. `https://eprint.iacr.org/2022/1323`. (Cited on page 7.)

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, August 1992. (Cited on page 31.)

[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 4.)

[FKM+18] Keisuke Fujii, Hirotada Kobayashi, Tomoyuki Morimae, Harumichi Nishimura, Seiichiro Tani, and Shuhei Tamate. Impossibility of classically simulating one-clean-qubit model with multiplicative error. *Physical Review Letters*, 120:200502, 2018. (Cited on page 3, 6.)

[FR99] Lance Fortnow and John Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences*, 59:240–252, 1999. (Cited on page 43.)

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016. (Cited on page 31.)

[GL89] Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32. ACM, 1989. (Cited on page 29.)

[GS20] Daniel Grier and Luke Schaeffer. Interactive shallow Clifford circuits: quantum advantage against NC^1 and beyond. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 875–888. ACM Press, June 2020. (Cited on page 11.)

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011. (Cited on page 7, 31.)

[HKEG19] Dominik Hangleiter, Martin Kliesch, Jens Eisert, and Christian Gogolin. Sample complexity of device-independently certified "quantum supremacy". *Physical Review Letters*, 122:21050, 2019. (Cited on page 3.)

[HM96] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 201–215. Springer, Heidelberg, August 1996. (Cited on page 5, 13, 14.)

[HNO+09] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Comput.*, 39(3):1153–1218, 2009. (Cited on page 4, 5, 8, 13, 14.)

[HT98] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 408–423. Springer, Heidelberg, August 1998. (Cited on page 31.)

[HY20] Akinori Hosoyamada and Takashi Yamakawa. Finding collisions in a quantum world: Quantum black-box separation of collision-resistance and one-wayness. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 3–32. Springer, Heidelberg, December 2020. (Cited on page 33.)

[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 45.)

[Jer16] Emil Jerábek. Integer factoring and modular square roots. *J. Comput. Syst. Sci.*, 82(2):380–394, 2016. (Cited on page 43.)

[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, Heidelberg, August 2018. (Cited on page 7.)

[KLVY22] Yael Tauman Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. Cryptology ePrint Archive, Paper 2022/400, 2022. https://eprint.iacr.org/2022/400. (Cited on page 4, 6.)

[KMCVY22] Gregory D. Kahanamoku-Meyer, Soonwon Choi, Umesh V. Vazirani, and Norman Y. Yao. Classically verifiable quantum advantage from a computational bell test. *Nature Physics*, 2022. (Cited on page 4, 6, 8, 9, 10, 25, 28, 45.)

[KNY18] Ilan Komargodski, Moni Naor, and Eylon Yogev. Collision resistant hashing for paranoids: Dealing with multiple collisions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 162–194. Springer, Heidelberg, April / May 2018. (Cited on page 4.)

[KQST22] William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in algorithmica. *arXiv:2212.00879*, 2022. (Cited on page 7.)

[Kre21] W. Kretschmer. Quantum pseudorandomness and classical complexity. *TQC 2021*, 2021. (Cited on page 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, Heidelberg, August 2018. (Cited on page 4, 13.)

[LLQ22] Jiahui Liu, Qipeng Liu, and Luowen Qian. Beating classical impossibility of position verification. ITCS 2022: 13rd Innovations in Theoretical Computer Science, 2022. (Cited on page 11.)

[Mah18] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 332–338. IEEE Computer Society Press, October 2018. (Cited on page 4.)

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000. (Cited on page 31.)

[Mor17] Tomoyuki Morimae. Hardness of classically sampling the one-clean-qubit model with constant total variation distance error. *Physical Review A*, 96:040302(R), 2017. (Cited on page 3, 6.)

[MY22a] Tomoyuki Morimae and Takashi Yamakawa. One-wayness in quantum cryptography. Cryptology ePrint Archive, Report 2022/1336, 2022. https://eprint.iacr.org/2022/1336. (Cited on page 7.)

[MY22b] 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, Heidelberg, August 2022. (Cited on page 7.)

[MY23] Tomoyuki Morimae and Takashi Yamakawa. Proofs of quantumness from trapdoor permutations. ITCS 2023: 14th Innovations in Theoretical Computer Science (ITCS)., 2023. (Cited on page 4, 6, 8, 9, 11, 28.)

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003. (Cited on page 7.)

[NOVY93] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for NP can be based on general complexity assumptions (extended abstract). In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 196–214. Springer, Heidelberg, August 1993. (Cited on page 9, 11.)

[OV08] Shien Jin Ong and Salil P. Vadhan. An equivalence between zero knowledge and commitments. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 482–500. Springer, Heidelberg, March 2008. (Cited on page 5, 14, 44.)

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *Second Israel Symposium on Theory of Computing Systems, ISTCS 1993, Natanya, Israel, June 7-9, 1993, Proceedings*, pages 3–17. IEEE Computer Society, 1993. (Cited on page 5, 12.)

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 1–20. Springer, Heidelberg, February 2004. (Cited on page 32, 33.)

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994. (Cited on page 3, 6.)

[Sta22] StackExchange. https://math.stackexchange.com/questions/4393348/pairwise-independent-balls-in-bins-probability-of-an-empty-bin. 2022. (Cited on page 15.)

[SV03] Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003. (Cited on page 44.)

[TD04] B. M. Terhal and D. P. DiVincenzo. Adaptive quantum computation, constant-depth circuits and arthur-merlin games. *Quant. Inf. Comput.*, 4(2):134–145, 2004. (Cited on page 3, 6.)

[Vad06] Salil Vadhan. An unconditional study of computational zero knowledge. *SIAM J. Comput.*, 36(4):1160–1214, 2006. (Cited on page 12.)

[WKST19] Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 515–526. ACM Press, June 2019. (Cited on page 11.)

[YZ22] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. FOCS 2022: 63rd IEEE Symposium on Foundations of Computer Science, 2022. (Cited on page 4, 6.)

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Inf. Comput.*, 15(7&8):557–567, 2015. (Cited on page 33.)

A Necessity of Assumptions for (AI-/IO-)IV-PoQ

In this appendix, we prove that the existence of AI-IV-PoQ implies $\mathbf{PP} \neq \mathbf{BPP}$. (Remember that IV-PoQ implies IO-IV-PoQ, and that IO-IV-PoQ implies AI-IV-PoQ.)

Assume that there is an AI-IV-PoQ. From it, we can construct another AI-IV-PoQ $(\mathcal{P}, \mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2))$ where each prover's message is a single bit. Without loss of generality, we can assume that the first phase of the AI-IV-PoQ runs as follows: For $j = 1, 2, ..., N$, the prover $\mathcal{P}$ and the verifier $\mathcal{V}_1$ repeat the following.

1. $\mathcal{P}$ possesses a state $|\psi_{j-1}\rangle$. It applies a unitary $U_j(\alpha_1, \beta_1, ..., \alpha_{j-1}, \beta_{j-1})$ that depends on the transcript $(\alpha_1, \beta_1, ..., \alpha_{j-1}, \beta_{j-1})$ obtained so far on the state, and measures a qubit in the computational basis to obtain the measurement result $\alpha_j \in \{0,1\}$. $\mathcal{P}$ sends $\alpha_j \in \{0,1\}$ to $\mathcal{V}_1$. The post-measurement state is $|\psi_j\rangle$.

2. $\mathcal{V}_1$ does some classical computing and sends a bit string β_j to $\mathcal{P}$.

We claim that a PPT prover $\mathcal{P}^*$ that queries the **#P** oracle can exactly sample from $\mathcal{P}$'s jth output probability distribution $\Pr[\alpha_j|\alpha_{j-1}, ..., \alpha_1]$ for each $j \in [N]$. This is shown as follows.

1. $\mathcal{P}^*$ computes $\Pr[\alpha_1] = \|(|\alpha_1\rangle\langle\alpha_1| \otimes I)U_1|0...0\rangle\|^2$ for each $\alpha_1 \in \{0,1\}$, and samples α_1 according to $\Pr[\alpha_1]$. Let $\alpha_1^* \in \{0,1\}$ be the result of the sampling. $\mathcal{P}^*$ sends α_1^* to $\mathcal{V}_1$.

2. $\mathcal{V}_1$ sends β_1^* to $\mathcal{P}^*$.

3. $\mathcal{P}^*$ and $\mathcal{V}_1$ repeat the following for $j = 2, 3, ..., N$.

   (a) $\mathcal{P}^*$ computes $\Pr[\alpha_j|\alpha_{j-1}^*, ..., \alpha_1^*] = \frac{\Pr[\alpha_j, \alpha_{j-1}^*, ..., \alpha_1^*]}{\Pr[\alpha_{j-1}^*, ..., \alpha_1^*]}$ for each $\alpha_j \in \{0,1\}$, and samples α_j according to $\Pr[\alpha_j|\alpha_{j-1}^*, ..., \alpha_1^*]$. Here,

   $$\begin{aligned}&\Pr[\alpha_j, \alpha_{j-1}^*, ..., \alpha_1^*]\\ &= \|(|\alpha_1^*\rangle\langle\alpha_1^*| \otimes ... \otimes |\alpha_{j-1}^*\rangle\langle\alpha_{j-1}^*| \otimes |\alpha_j\rangle\langle\alpha_j| \otimes I)U_j(\alpha_1^*, \beta_1^*, ..., \alpha_{j-1}^*, \beta_{j-1}^*)...U_2(\alpha_1^*, \beta_1^*)U_1|0...0\rangle\|^2.\end{aligned} \tag{183}$$

   Let $\alpha_j^* \in \{0,1\}$ be the result of the sampling. $\mathcal{P}^*$ sends α_j^* to $\mathcal{V}_1$.

   (b) $\mathcal{V}_1$ sends β_j^* to $\mathcal{P}^*$.

It is known that for any QPT algorithm that outputs $z \in \{0,1\}^\ell$, the probability $\Pr[z]$ that it outputs z can be computed in classical deterministic polynomial-time by querying the **#P** oracle [FR99]. Therefore, $\mathcal{P}^*$ can compute $\Pr[\alpha_1]$ and $\Pr[\alpha_j|\alpha_{j-1}^*, ..., \alpha_1^*]$ for any $j = 2, 3, ..., N$. It is known that a **#P** function can be computed in classical deterministic polynomial-time by querying the $\mathbf{PP}$ oracle. Therefore, if $\mathbf{PP} = \mathbf{BPP}$, $\mathcal{P}^*$ is enough to be a PPT algorithm.

B Omitted Contents in Section 2

B.1 Auxiliary-Input Collision-Resistance and $\mathbf{PWPP} \nsubseteq \mathbf{FBPP}$

We prepare several definitions.

Definition B.1 (Auxiliary-input collision-resistant hash functions). *An auxiliary-input collision-resistant hash function is a polynomial-time computable auxiliary-input function ensemble* $\mathcal{H} := \{h_\sigma : \{0,1\}^{p(|\sigma|)} \to \{0,1\}^{q(|\sigma|)}\}_{\sigma \in \{0,1\}^*}$ *such that* $q(|\sigma|) < p(|\sigma|)$ *and for every uniform PPT adversary* $\mathcal{A}$ *and polynomial* poly*, there exists an infinite set* $\Lambda \subseteq \{0,1\}^*$ *such that,*

$$\Pr[h_\sigma(x') = h_\sigma(x) : (x, x') \leftarrow \mathcal{A}(\sigma)] \le \frac{1}{\mathrm{poly}(|\sigma|)} \tag{184}$$

for all $\sigma \in \Lambda$.

A search problem is specified by a relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ where one is given $x \in \{0,1\}^*$ and asked to find $y \in \{0,1\}^*$ such that $(x, y) \in R$. We say that a search problem R is many-one reducible to another search problem S if there are polynomial-time computable functions f and g such that $(f(x), y) \in S$ implies $(x, g(x, y)) \in R$.

A search problem Weak Pigeon is defined as follows: $(C, (x, x')) \in$ Weak Pigeon if and only if C represents a circuit from $\{0,1\}^m$ to $\{0,1\}^n$ for $m > n$, $x \neq x'$, and $C(x) = C(x')$.

Definition B.2 ([Jer16]). $\mathbf{PWPP}$ *is the class of all search problems that are many-one reducible to Weak Pigeon.*

Definition B.3 ([Aar10]). $\mathbf{FBPP}$ *is the class of all search problems R for which there exists a PPT algorithm $\mathcal{A}$ such that for any $x \in \{0,1\}^*$,*

$$\Pr[(x,y) \in R : y \leftarrow \mathcal{A}(x)] = 1 - o(1). \tag{185}$$

The following theorem almost directly follows from the definitions.

Theorem B.4. *There exist auxiliary-input collision-resistant hash functions if and only if $\mathbf{PWPP} \not\subseteq \mathbf{FBPP}$.*

Proof. We first show the "if" direction. Toward contradiction, we assume that there is no auxiliary-input collision-resistant hash function. We consider a polynomial-time computable auxiliary-input function ensemble $\mathcal{H} = \{h_\sigma\}$ defined as follows: if σ represents a circuit $C : \{0,1\}^n \to \{0,1\}^m$ such that $n > m$, then h_σ takes $x \in \{0,1\}^n$ and outputs $C(x) \in \{0,1\}^m$. Otherwise, h_σ is defined arbitrarily, (say, h_σ is a constant function that always outputs 0). By our assumption, this is not an auxiliary-input collision-resistant hash function, so there is a PPT algorithm $\mathcal{A}$ and a polynomial poly such that

$$\Pr[h_\sigma(x') = h_\sigma(x) : (x,x') \leftarrow \mathcal{A}(\sigma)] \geq 1/\mathrm{poly}(|\sigma|) \tag{186}$$

for all but finitely many σ. For those σ, we can amplify the success probability of $\mathcal{A}$ to $1 - o(1)$ by polynomially many times repetition. This gives a PPT algorithm that solves Weak Pigeon with probability $1 - o(1)$, which contradicts $\mathbf{PWPP} \not\subseteq \mathbf{FBPP}$. Therefore, the above $\mathcal{H}$ must be an auxiliary-input collision-resistant hash function.

Next, we prove the "only if" direction. Toward contradiction, we assume $\mathbf{PWPP} \subseteq \mathbf{FBPP}$. This means that there is a PPT algorithm that solves Weak Pigeon with probability $1 - o(1)$ on all instances. Let $\mathcal{H} = \{h_\sigma\}$ be an arbitrary polynomial-time computable auxiliary-input function ensemble. Then we can use the algorithm that solves Weak Pigeon to find a collision of h_σ for all auxiliary inputs σ with probability $1 - o(1)$. This means that $\mathcal{H} = \{h_\sigma\}$ is not an auxiliary-input collision-resistant hash function. This contradicts the existence of auxiliary-input collision-resistant hash functions. Thus, we have $\mathbf{PWPP} \not\subseteq \mathbf{FBPP}$. □

B.2 Auxiliary-Input Commitments from $\mathbf{SZK} \not\subseteq \mathbf{BPP}$

We prove Theorem 2.17, i.e., we construct constant-round auxiliary-input statistically-hiding and computationally-binding bit commitments assuming $\mathbf{SZK} \not\subseteq \mathbf{BPP}$.

First, we recall the definition of instance-dependent commitments.

Definition B.5 (Instance-dependent commitment). *Instance-dependent commitments for a promise problem $(L_{\mathsf{yes}}, L_{\mathsf{no}})$ is a family of commitment schemes $\{\Pi_\sigma\}_{\sigma\in\{0,1\}^*}$ such that*

- *Π_σ is statistically hiding if $\sigma \in L_{\mathsf{yes}}$,*
- *Π_σ is statistically binding if $\sigma \in L_{\mathsf{no}}$.*

Theorem B.6 ([OV08]). *For any promise problem $(L_{\mathsf{yes}}, L_{\mathsf{no}}) \in \mathbf{SZK}$, there is a constant-round instance-dependent commitments for $(L_{\mathsf{yes}}, L_{\mathsf{no}})$.*

Instance-dependent commitments for $\mathbf{SZK}$ directly give auxiliary-input statistically-hiding and computationally-binding commitments if $\mathbf{SZK} \not\subseteq \mathbf{BPP}$.

Proof of Theorem 2.17. Fix a $\mathbf{SZK}$-complete promise problem $(L_{\mathsf{yes}}, L_{\mathsf{no}})$. (For example, we can take the statistical difference problem [SV03].) We regard instance-dependent commitments for $(L_{\mathsf{yes}}, L_{\mathsf{no}})$ as auxiliary-input commitments where $\Sigma := L_{\mathsf{yes}}$. Then correctness and statistical hiding as auxiliary-input commitments directly follows from those of instance-dependent commitments noting that they are statistically hiding if $\sigma \in L_{\mathsf{yes}}$. Suppose that this scheme does not satisfy computational binding. Then there is a PPT malicious committer $\mathcal{S}^*$ that finds decommitments to both 0 and 1 with an inverse-polynomial probability for all but finitely many $\sigma \in L_{\mathsf{yes}}$. We can augment it to construct $\widetilde{\mathcal{S}}^*$ that works for all $\sigma \in L_{\mathsf{yes}}$ by letting it find decommitments by brute-force for σ for which $\mathcal{S}^*$ fails to find decommitments. Since there are only finitely many such σ, $\widetilde{\mathcal{S}}^*$ still runs in PPT. On the other hand, if $\sigma \in L_{\mathsf{no}}$, $\widetilde{\mathcal{S}}^*$ cannot find decommitments to 0 and 1 except for a negligible probability. Therefore, we can use $\widetilde{\mathcal{S}}^*$ to distinguish elements of L_{yes} and L_{no} with an inverse-polynomial distinguishing gap. Since $(L_{\mathsf{yes}}, L_{\mathsf{no}})$ is $\mathbf{SZK}$-complete, this means $\mathbf{SZK} \subseteq \mathbf{BPP}$. This is a contradiction, and thus the above commitment scheme satisfies computational binding. □

C Omitted Proofs for the Completeness

In this appendix, we show that if $\mathcal{P}$ starts Step 5 of Algorithm 4 with the state $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$, the probability that $\mathcal{V}_2$ outputs $\top$ is $\frac{1}{2} + \frac{1}{2}\cos^2\frac{\pi}{8}$. The proof is the same as that in [KMCVY22], but for the convenience of readers, we provide it here.

First, the probability that $\mathcal{V}_1$ chooses $v_1 = 0$ is 1/2, and in that case, the honest $\mathcal{P}$ sends correct x_0 or x_1, and therefore $\mathcal{V}_2$ outputs $\top$ with probability 1.

Second, the probability that $\mathcal{V}_1$ chooses $v_1 = 1$ is 1/2. In that case, Equation (99) is $|\xi \cdot x_0\rangle + (-1)^{d\cdot(x_0 \oplus x_1)}|1 \oplus (\xi \cdot x_1)\rangle$, which is one of the BB84 states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. By the straightforward calculations, we can show that

$$\Pr[\eta = 0 | \, |0\rangle, v_2 = 0] \tag{187}$$
$$\Pr[\eta = 1 | \, |1\rangle, v_2 = 0] \tag{188}$$
$$\Pr[\eta = 0 | \, |+\rangle, v_2 = 0] \tag{189}$$
$$\Pr[\eta = 1 | \, |-\rangle, v_2 = 0] \tag{190}$$
$$\Pr[\eta = 0 | \, |0\rangle, v_2 = 1] \tag{191}$$
$$\Pr[\eta = 1 | \, |1\rangle, v_2 = 1] \tag{192}$$
$$\Pr[\eta = 1 | \, |+\rangle, v_2 = 1] \tag{193}$$
$$\Pr[\eta = 0 | \, |-\rangle, v_2 = 1] \tag{194}$$

are all equal $\cos^2\frac{\pi}{8}$. Then, we can confirm that the probability that

$$(\xi \cdot x_0 \neq \xi \cdot x_1) \wedge (\eta = \xi \cdot x_0), \tag{195}$$

or

$$(\xi \cdot x_0 = \xi \cdot x_1) \wedge (\eta = v_2 \oplus d \cdot (x_0 \oplus x_1)) \tag{196}$$

occurs is $\cos^2\frac{\pi}{8}$. In fact, if $\xi \cdot x_0 \neq \xi \cdot x_1$, $\mathcal{P}$'s state is $|0\rangle$ or $|1\rangle$. In that case, the probability that $\eta = \xi \cdot x_0$ is $\cos^2\frac{\pi}{8}$ for any $v_2 \in \{0,1\}$. If $\xi \cdot x_0 = \xi \cdot x_1$, $\mathcal{P}$'s state is $|+\rangle$ or $|-\rangle$. In that case, the probability that $\eta = v_2 \oplus d \cdot (x_0 \oplus x_1)$ is $\cos^2\frac{\pi}{8}$.

D Distributionally OWFs

Definition D.1 (Distributionally OWFs [IL89]). *A polynomial-time-computable function $f : \{0,1\}^* \to \{0,1\}^*$ is distributionally one-way if there exists a polynomial p such that for any PPT algorithm $\mathcal{A}$ and all sufficiently large λ, the statistical difference between $\{(x, f(x))\}_{x\leftarrow\{0,1\}^\lambda}$ and $\{(\mathcal{A}(1^\lambda, f(x)), f(x))\}_{x\leftarrow\{0,1\}^\lambda}$ is greater than $1/p(\lambda)$.*

The following result is known.

Lemma D.2 ([IL89]). *If distributionally OWFs exist then OWFs exist.*