One-Wayness in Quantum Cryptography

Tomoyuki Morimae[1] and Takashi Yamakawa[2,3,1]

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

Abstract

The existence of one-way functions is one of the most fundamental assumptions in classical cryptography. In the quantum world, on the other hand, there are evidences that some cryptographic primitives can exist even if one-way functions do not exist [Kretschmer, TQC 2021; Morimae and Yamakawa, CRYPTO 2022; Ananth, Qian, and Yuen, CRYPTO 2022]. We therefore have the following important open problem in quantum cryptography: What is the most fundamental assumption in quantum cryptography? In this direction, [Brakerski, Canetti, and Qian, ITCS 2023] recently defined a notion called EFI pairs, which are pairs of efficiently generatable states that are statistically distinguishable but computationally indistinguishable, and showed its equivalence with some cryptographic primitives including commitments, oblivious transfer, and general multi-party computations. However, their work focuses on decision-type primitives and does not cover search-type primitives like quantum money and digital signatures. In this paper, we study properties of one-way state generators (OWSGs), which are a quantum analogue of one-way functions proposed by Morimae and Yamakawa. We first revisit the definition of OWSGs and generalize it by allowing mixed output states. Then we show the following results.

1. We define a weaker version of OWSGs, which we call weak OWSGs, and show that they are equivalent to OWSGs. It is a quantum analogue of the amplification theorem for classical weak one-way functions.
2. (Bounded-time-secure) quantum digital signatures with quantum public keys are equivalent to OWSGs.
3. Private-key quantum money schemes (with pure money states) imply OWSGs.
4. Quantum pseudo one-time pad schemes imply both OWSGs and EFI pairs. For EFI pairs, single-copy security suffices.
5. We introduce an incomparable variant of OWSGs, which we call secretly-verifiable and statistically-invertible OWSGs, and show that they are equivalent to EFI pairs.

Contents

1 Introduction

One-way functions (OWFs) are functions that are easy to compute but hard to invert. The existence of OWFs is one of the most fundamental assumptions in classical cryptography. OWFs are equivalent to many cryptographic primitives, such as commitments, digital signatures, pseudorandom generators (PRGs), symmetric-key encryption (SKE), and zero-knowledge, etc. Moreover, almost all other cryptographic primitives, such as collision-resistant hashes, public-key encryption (PKE), oblivious transfer (OT), multi-party computations (MPCs), etc., imply OWFs. In the quantum world, on the other hand, it seems that OWFs are not necessarily the most fundamental element. In fact, recently, several quantum cryptographic primitives, such as commitments, (one-time secure) digital signatures, quantum pseudo one-time pad (QPOTP)[1], and MPCs are constructed from pseudorandom states generators (PRSGs) [MY22, AQY22]. A PRSG [JLS18], which is a quantum analogue of a PRG, is a QPT algorithm that outputs a quantum state whose polynomially-many copies are computationally indistinguishable from the same number of copies of Haar random states. Kretschmer [Kre21] showed that PRSGs exist even if $\mathbf{BQP} = \mathbf{QMA}$ (relative to a quantum oracle), which means that PRSGs (and all the above primitives that can be constructed from PRSGs) could exist even if all quantum-secure (classical) cryptographic primitives including OWFs are broken.[2] Kretschmer, Qian, Sinha, and Tal [KQST23] also showed that 1-PRSGs (which are variants of PRSGs secure against adversaries that get only a single copy of the state) exist even if $\mathbf{NP} = \mathbf{P}$. We therefore have the following important open problem in quantum cryptography:

Question 1: *What is the most fundamental assumption in quantum cryptography?*

In classical cryptography, a pair of PPT algorithms whose output probability distributions are statistically distinguishable but computationally indistinguishable is known to be fundamental. Goldreich [Gol90] showed the equivalence of such a pair to PRGs, which also means the equivalence of such a pair to all cryptographic primitives in Minicrypt [Imp95]. It is natural to consider its quantum analogue: a pair of QPT algorithms whose output quantum states are statistically distinguishable but computationally indistinguishable. In fact, such a pair was implicitly studied in quantum commitments [Yan22]. In the canonical form of quantum commitments [YWLQ15], computationally hiding and statistically binding quantum commitments are equivalent to such pairs. The importance of such a pair as an independent quantum cryptograpic primitive was pointed out in [Yan22, BCQ22]. In particular, the authors of [BCQ22] explicitly defined it as *EFI pairs*,[3] and showed that EFI pairs are implied by several quantum cryptographic primitives such as (semi-honest) quantum OT, (semi-honest) quantum MPCs, and (honest-verifier) quantum computational zero-knowledge proofs. It is therefore natural to ask the following question.

Question 2: *Which other quantum cryptographic primitives imply EFI pairs?*

PRSGs and EFI pairs are "decision type" primitives, which correspond to PRGs in classical cryptography. An example of the other type of primitives, namely, "search type" one in classical cryptography, is OWFs. Recently, a quantum analogue of OWFs, so called one-way states generators (OWSGs), are introduced [MY22]. A OWSG is a QPT algorithm that, on input a classical bit string (key) k, outputs a quantum state $|\phi_k\rangle$. As the security, we require that it is hard to find k' such that $|\langle\phi_k|\phi_{k'}\rangle|^2$ is non-negligible given polynomially many copies of $|\phi_k\rangle$. The authors showed that OWSGs are implied by PRSGs, and that OWSGs imply (one-time

[1] QPOTP schemes are a one-time-secure SKE with quantum ciphertexts where the key length is shorter than the massage length. (For the definition, see Definition 6.1.)

[2] If $\mathbf{QMA} = \mathbf{BQP}$, then $\mathbf{NP} \subseteq \mathbf{BQP}$. Because all quantum-secure classical cryptographic primitives are in $\mathbf{NP}$, it means that they are broken by QPT algorithms.

[3] It stands for efficiently samplable, statistically far but computationally indistinguishable pairs of distributions.

secure) quantum digital signatures with quantum public keys. In classical cryptography, OWFs are connected to many cryptographic primitives. We are therefore interested in the following question.

Question 3: *Which quantum cryptographic primitives are related to OWSGs?*

In classical cryptography, PRGs (i.e., a decision-type primitive) and OWFs (i.e., a search-type primitive) are equivalent. In quantum cryptography, on the other hand, we do not know whether OWSGs and EFI pairs (or PRSGs) are equivalent or not. We therefore have the following open problem.

Question 4: *Are OWSGs and EFI pairs (or PRSGs) equivalent?*

1.1 Our Results

The study of quantum cryptography with complexity assumptions has became active only very recently, and therefore we do not yet have enough knowledge to answer **Question 1**. However, as an important initial step towards the ultimate goal, we give some answers to other questions above. Our results are summarized as follows. (See also Fig. 1.)

1. We first revisit the definition of OWSGs. In the original definition in [MY22], output states of OWSGs are assumed to be pure states. Moreover, the verification is done as follows: a bit string k' from the adversary is accepted if and only if the state $|\phi_k\rangle\langle\phi_k|$ is measured in the basis $\{|\phi_{k'}\rangle\langle\phi_{k'}|, I-|\phi_{k'}\rangle\langle\phi_{k'}|\}$, and the first result is obtained. (Note that in classical OWFs, the verification is implicit because it is trivial: just computing $f(x')$ for x' given by the adversary, and check whether it is equal to $f(x)$ or not. However, in the quantum case, we have to explicitly define the verification.) In this paper, to capture more general settings, we generalize the definition of OWSGs by allowing outputs to be mixed states. A non-trivial issue that arises from this modification is that there is no canonical way to verify input-output pairs of OWSGs. To deal with this issue, we include such a verification algorithm as a part of syntax of OWSGs. See Definition 3.1 for the formal definition.

2. We show an "amplification theorem" for OWSGs. That is, we define weak OWSGs (wOWSGs), which only requires the adversary's advantage to be $1-1/\mathrm{poly}(\lambda)$ instead of $\mathsf{negl}(\lambda)$, and show that a parallel repetition of wOWSGs gives OWSGs (Section 3.2). This is an analogue of the equivalence of weak one-way functions and (strong) one-way functions in classical cryptography [Yao82].

3. We show that one-time-secure quantum digital signatures (QDSs) with quantum public keys are equivalent to OWSGs (Section 4.3).[4] Moreover, we can generically upgrade one-time-secure QDSs into bounded-time-secure one (Section 4.2).[5]

4. We show that private-key quantum money schemes (with pure money states or with verification algorithms that satisfy some symmetry) imply OWSGs (Section 5.2 and Section 5.3).

5. We show that QPOTP schemes imply OWSGs (Section 6.2). This in particular means that IND-CPA secure quantum SKE or quantum PKE implies OWSGs (Appendix E).

6. We show that single-copy-secure QPOTP schemes imply EFI pairs (Section 6.3). Single-copy-security means that the adversary receives only a single copy of the quantum ciphertext. This in particular means that IND-CPA secure quantum SKE or quantum PKE implies EFI pairs (Appendix E).

[4] A construction of QDSs from OWSGs was already shown in [MY22], but in this paper, we generalize the definition of OWSGs, and we give the proof in the new definition.

[5] We thank Or Sattath for asking if we can get (stateless) bounded-time QDSs.

7. We introduce an incomparable variant of OWSGs, which we call secretly-verifiable and statistically-invertible OWSGs (SV-SI-OWSGs) (Section 7.1), and show that SV-SI-OWSGs are equivalent to EFI pairs (Section 7.2).

We remark that we consider the generalized definition of OWSGs with mixed state outputs by default. However, all the relationships between OWSGs and other primitives naturally extend to the pure state version if we consider the corresponding pure state variants of the primitives.

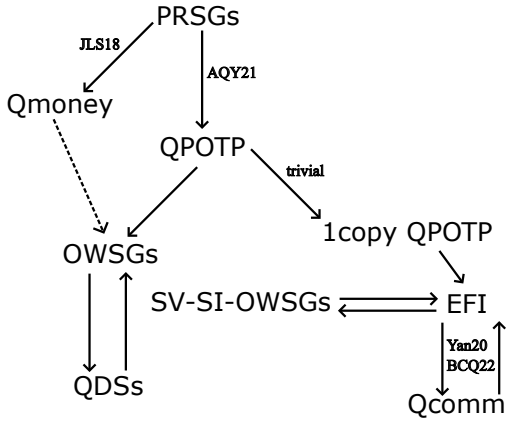

Figure 1: Summary of results. The dotted line means some restrictions: OWSGs are implied by quantum money schemes with *pure* money states or with *symmetric* verification algorithms.

1.2 Concurrent Work

A concurrent work by Cao and Xue [CX22] also studies OWSGs. In particular, they also show the equivalence between weak OWSGs and OWSGs. Though they consider OWSGs with pure state outputs as defined in [MY22], it is likely that their proof extends to the mixed state version as well. Interestingly, the proof strategies are different in our and their works. Their proof is based on the classical proof of the equivalence between the weak one-way functions and one-way functions [Yao82, Gol01]. On the other hand, our proof is based on the amplification theorem for weakly verifiable puzzles [CHS05]. Though their proof is simpler, an advantage of our approach is that it captures more general settings. For example, our proof also works for secretly-verifiable OWSGs (Definition 7.1). Their approach seems to rely on public verifiability in an essential manner and not applicable to secretly-verifiable OWSGs.

Besides the equivalence between weak OWSGs and OWSGs, there is no other overlap between our and their results.

1.3 Open Problems

The study of "quantum cryptography without one-way functions" has just started, and our understanding is very limited. There are many open problems in this emerging field, but we believe the following open problems are important, and some of them seem to be highly challenging.

1. What is the most fundamental assumption in quantum cryptography? It should be implied by many primitives, and should imply many primitives. It should also be simple. Or, there is no such thing in quantum cryptography?

2. OWSGs=EFI? Or at least can we show OWSGs→EFI[6] or EFI→OWSGs? If they are incomparable, is there any more fundamental primitive that is implied by both OWSGs and EFI pairs?

3. Do OWSGs imply private-key quantum money schemes? An adversary who can clone a quantum money state ψ_k would not necessarily be able to find the secret key k.

4. Do EFI pairs imply single-copy-secure PRSGs?

5. Can we construct unbounded-poly many-time secure digital signatures without one-way functions?

6. Which other primitives can be constructed without one-way functions? For example, how about PKE, NIZK, or proofs of quantumness?

7. $\mathbf{PP} \neq \mathbf{BQP}$ is necessary for the existence of PRSGs [Kre21]. What are the classical complexity assumptions necessary for the existences of other primitives, such as OWSGs, private-key quantum money schemes, and EFI pairs?[7]

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. negl is a negligible function, and poly is a polynomial. PPT stands for (classical) probabilistic polynomial-time and QPT stands for quantum polynomial-time. If we say that an adversary is QPT, it implicitly means non-uniform QPT. A QPT unitary is a unitary operator that can be implemented in a QPT quantum circuit.

For an algorithm A, $y \leftarrow A(x)$ means that the algorithm A outputs y on input x. In particular, if x and y are quantum states and A is a quantum algorithm, $y \leftarrow A(x)$ means the following: a unitary U is applied on $x \otimes |0...0\rangle\langle 0...0|$, and some qubits are traced out. Then, the state of remaining qubits is y. This, importantly, means that the state y is *uniquely decided* by the state x. If A is a QPT algorithm, the unitary U is QPT and the number of ancilla qubits $|0...0\rangle$ is $\mathrm{poly}(\lambda)$. If x is a classical bit string, y is a quantum state, and A is a quantum algorithm, $y \leftarrow A(x)$ sometimes means the following: a unitary U_x that depends on x is applied on $|0...0\rangle$, and some qubits are traced out. The state of the remaining qubits is y. This picture is the same as the most general one where x is given as input, but we sometime choose this picture if it is more convenient.

$\|X\|_1 := \mathrm{Tr}\sqrt{X^\dagger X}$ is the trace norm. $\mathrm{Tr}_{\mathbf{A}}(\rho_{\mathbf{A},\mathbf{B}})$ means that the subsystem (register) $\mathbf{A}$ of the state $\rho_{\mathbf{A},\mathbf{B}}$ on two subsystems (registers) $\mathbf{A}$ and $\mathbf{B}$ is traced out. For simplicity, we sometimes write $\mathrm{Tr}_{\mathbf{A},\mathbf{B}}(|\psi\rangle_{\mathbf{A},\mathbf{B}})$ to mean $\mathrm{Tr}_{\mathbf{A},\mathbf{B}}(|\psi\rangle\langle\psi|_{\mathbf{A},\mathbf{B}})$. I is the two-dimensional identity operator. For simplicity, we sometimes write $I^{\otimes n}$ as I if the dimension is clear from the context. For the notational simplicity, we sometimes write $|0...0\rangle$ just as $|0\rangle$, when the number of zeros is clear from the context. For two pure states $|\psi\rangle$ and $|\phi\rangle$, we sometimes write $\||\psi\rangle\langle\psi| - |\phi\rangle\langle\phi|\|_1$ as $\||\psi\rangle - |\phi\rangle\|_1$ to simplify the notation. $F(\rho, \sigma) := \|\sqrt{\rho}\sqrt{\sigma}\|_1^2$ is the fidelity between ρ and σ. We often use the well-known relation between the trace distance and the fidelity: $1 - \sqrt{F(\rho,\sigma)} \leq \frac{1}{2}\|\rho - \sigma\|_1 \leq \sqrt{1 - F(\rho,\sigma)}$.

[6] OWSGs→EFI has recently been shown in [KT23] for pure-output OWSGs.

[7] Recently, there have been some progresses regarding this open problem. First, it was shown that $\mathbf{PP} \neq \mathbf{BQP}$ is necessary for the existence of pure OWSGs [CGG+23]. Second, an evidence is given that single-copy PRSGs (and hence EFI pairs) could exist even if $\mathbf{P} = \mathbf{ALL}$ [LMW23].

2.2 EFI Pairs

The concept of EFI pairs was implicitly studied in [Yan22], and explicitly defined in [BCQ22].

Definition 2.1 (EFI pairs [BCQ22]). *An EFI pair is an algorithm* $\mathsf{StateGen}(b, 1^\lambda) \to \rho_b$ *that, on input* $b \in \{0,1\}$ *and the security parameter* λ*, outputs a quantum state* ρ_b *such that all of the following three conditions are satisfied.*

- *It is a uniform QPT algorithm.*
- ρ_0 *and* ρ_1 *are computationally indistinguishable. In other words, for any QPT adversary* $\mathcal{A}$*,* $|\Pr[1 \leftarrow \mathcal{A}(1^\lambda, \rho_0)] - \Pr[1 \leftarrow \mathcal{A}(1^\lambda, \rho_1)]| \leq \mathsf{negl}(\lambda)$.
- ρ_0 *and* ρ_1 *are statistically distinguishable, i.e.,* $\frac{1}{2}\|\rho_0 - \rho_1\|_1 \geq \frac{1}{\mathrm{poly}(\lambda)}$.

Remark 2.2. Note that in the above definition, the statistical distinguishability is defined with only $\geq 1/\mathrm{poly}(\lambda)$ advantage. However, if EFI pairs with the above definition exist, EFI pairs with $\geq 1 - \mathsf{negl}(\lambda)$ statistical distinguishability exist as well. In fact, we have only to define a new $\mathsf{StateGen}'$ that runs $\mathsf{StateGen}$ n times with sufficiently large $n = \mathrm{poly}(\lambda)$, and outputs $\rho_b^{\otimes n}$. The $\geq 1 - \mathsf{negl}(\lambda)$ statistical distinguishability for $\mathsf{StateGen}'$ is shown from the inequality [BCQ22],

$$\frac{1}{2}\|\rho^{\otimes n} - \sigma^{\otimes n}\|_1 \geq 1 - \exp(-n\|\rho - \sigma\|_1/4).$$

The computational indistinguishability for $\mathsf{StateGen}'$ is shown by the standard hybrid argument.

2.3 Quantum Commitments

We define canonical quantum bit commitments [Yan22] as follows.

Definition 2.3 (Canonical quantum bit commitments [Yan22]). *A canonical quantum bit commitment scheme is a family* $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ *of QPT unitaries on two registers* $\mathbf{C}$ *(called the* commitment *register) and* $\mathbf{R}$ *(called the* reveal *register). For simplicity, we often omit* λ *and simply write* $\{Q_0, Q_1\}$ *to mean* $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$.

Remark 2.4. Canonical quantum bit commitments are used as follows. In the commit phase, to commit to a bit $b \in \{0,1\}$, the sender generates a state $Q_b|0\rangle_{\mathbf{C},\mathbf{R}}$ and sends $\mathbf{C}$ to the receiver while keeping $\mathbf{R}$. In the reveal phase, the sender sends b and $\mathbf{R}$ to the receiver. The receiver projects the state on $(\mathbf{C}, \mathbf{R})$ onto $Q_b|0\rangle_{\mathbf{C},\mathbf{R}}$, and accepts if it succeeds and otherwise rejects. (In other words, the receiver applies the unitary $Q_b^\dagger$ on the registers $\mathbf{C}$ and $\mathbf{R}$, and measure all qubits in the computational basis. If all result are zero, accept. Otherwise, reject.)

Definition 2.5 (Hiding). *We say that a canonical quantum bit commitment scheme* $\{Q_0, Q_1\}$ *is computationally (rep. statistically)* hiding *if* $\mathrm{Tr}_{\mathbf{R}}(Q_0|0\rangle_{\mathbf{C},\mathbf{R}})$ *is computationally (resp. statistically) indistinguishable from* $\mathrm{Tr}_{\mathbf{R}}(Q_1|0\rangle_{\mathbf{C},\mathbf{R}})$*. We say that it is perfectly hiding if they are identical states.*

Definition 2.6 (Binding). *We say that a canonical quantum bit commitment scheme* $\{Q_0, Q_1\}$ *is computationally (rep. statistically)* binding *if for any QPT (resp. unbounded-time) unitary* U *over* $\mathbf{R}$ *and an additional register* $\mathbf{Z}$ *and any polynomial-size state* $|\tau\rangle_{\mathbf{Z}}$*, it holds that*

$$\left\|(\langle 0|Q_1^\dagger)_{\mathbf{C},\mathbf{R}}(I_{\mathbf{C}} \otimes U_{\mathbf{R},\mathbf{Z}})((Q_0|0\rangle)_{\mathbf{C},\mathbf{R}}|\tau\rangle_{\mathbf{Z}})\right\| = \mathsf{negl}(\lambda). \tag{1}$$

We say that it is perfectly hiding if the LHS is 0 *for all unbounded-time unitary* U. [8]

Remark 2.7. One may think that honest-binding defined above is too weak because it only considers honestly generated commitments. However, somewhat surprisingly, [Yan22] proved that it is equivalent to another binding notion called the *sum-binding* [DMS00].[9] The sum-binding property requires that the sum of probabilities that any (quantum polynomial-time, in the case of computational binding) *malicious* sender can open a commitment to 0 and 1 is at most $1 + \mathsf{negl}(\lambda)$. In addition, it has been shown that the honest-binding property is sufficient for cryptographic applications including zero-knowledge proofs/arguments (of knowledge), oblivious transfers, and multi-party computation [YWLQ15, FUYZ20, MY22, Yan21]. In this paper, we refer to honest-binding if we simply write binding.

In this paper, we use the following result.

Theorem 2.8 (Converting flavors [Yan22, HMY23]). *Let* $\{Q_0, Q_1\}$ *be a canonical quantum bit commitment scheme. Then there exists a canonical quantum bit commitment scheme* $\{Q'_0, Q'_1\}$*, and the following hold for* $\mathtt{X}, \mathtt{Y} \in \{$*computationally,statistically,perfectly*$\}$*:*

- *If* $\{Q_0, Q_1\}$ *is* $\mathtt{X}$ *hiding, then* $\{Q'_0, Q'_1\}$ *is* $\mathtt{X}$ *binding.*
- *If* $\{Q_0, Q_1\}$ *is* $\mathtt{Y}$ *binding, then* $\{Q'_0, Q'_1\}$ *is* $\mathtt{Y}$ *hiding.*

2.4 PRSGs

Although in this paper we do not use PRSGs, we provide its definition in Appendix A for the convenience of readers.

3 OWSGs

In this section, we first define OWSGs (Section 3.1). We then define weak OWSGs and show that weak OWSGs are equivalent to OWSGs (Section 3.2).

3.1 Definition of OWSGs

In this subsection, we define OWSGs. Note that the definition below is a generalization of the one given in [MY22] in the following three points. First, in [MY22], the generated states are pure, but here they can be mixed. Second, in [MY22], the secret key k is uniformly sampled at random, but now it is sampled by a QPT algorithm. Third, in [MY22], the verification algorithm is the specific algorithm that accepts the alleged key k' with probability $|\langle\phi_k|\phi_{k'}\rangle|^2$, while here we consider a general verification algorithm. We think the definition below is more general (and therefore more fundamental) than that in [MY22]. Hence hereafter we choose the definition below as the definition of OWSGs.

Definition 3.1 (One-way states generators (OWSGs)). *A one-way states generator (OWSG) is a set of algorithms* $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$ *such that*

[8] The above definition is asymmetric for 0 and 1, but it is easy to show that Equation (1) implies

$$\left\|(\langle 0|\, Q_0^\dagger)_{\mathbf{C},\mathbf{R}}(I_{\mathbf{C}} \otimes U_{\mathbf{R},\mathbf{Z}})((Q_1\,|0\rangle)_{\mathbf{C},\mathbf{R}}\,|\tau\rangle_{\mathbf{Z}})\right\| = \mathsf{negl}(\lambda)$$

for any U and $|\tau\rangle$.

[9] The term "sum-binding" is taken from [Unr16].

- $\mathsf{KeyGen}(1^\lambda) \to k$: *It is a QPT algorithm that, on input the security parameter λ, outputs a classical key $k \in \{0,1\}^\kappa$.*
- $\mathsf{StateGen}(k) \to \phi_k$: *It is a QPT algorithm that, on input k, outputs an m-qubit quantum state ϕ_k.*
- $\mathsf{Ver}(k', \phi_k) \to \top/\bot$: *It is a QPT algorithm that, on input ϕ_k and a bit string k', outputs $\top$ or $\bot$.*

We require the following correctness and security.

Correctness:

$$\Pr[\top \leftarrow \mathsf{Ver}(k, \phi_k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k)] \geq 1 - \mathsf{negl}(\lambda).$$

Security: *For any QPT adversary $\mathcal{A}$ and any polynomial t* [10]*,*

$$\Pr[\top \leftarrow \mathsf{Ver}(k', \phi_k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k), k' \leftarrow \mathcal{A}(1^\lambda, \phi_k^{\otimes t})] \leq \mathsf{negl}(\lambda).$$

Remark 3.2. In Appendix B, we show that if all ϕ_k are pure and $\Pr[\top \leftarrow \mathsf{Ver}(k, \phi_k)] \geq 1 - \mathsf{negl}(\lambda)$ for all k, restricting Ver to the following specific algorithm (used in [MY22]) does not lose the generality: On input k' and ϕ_k, measure ϕ_k with the basis $\{|\phi_{k'}\rangle\langle\phi_{k'}|, I - |\phi_{k'}\rangle\langle\phi_{k'}|\}$. If the first result is obtained output $\top$. Otherwise, output $\bot$.

Remark 3.3. If ϕ_k is pure, StateGen runs as follows. Apply a QPT unitary U on $|k\rangle|0...0\rangle$ to generate $|\phi_k\rangle \otimes |\eta_k\rangle$, and output $|\phi_k\rangle$. In this case, the existence of the "junk state" $|\eta_k\rangle$ is essential, because otherwise it is not secure against a QPT adversary who does the application of $U^\dagger$ and the computational-basis measurement.

Remark 3.4. OWSGs are constructed from PRSGs [MY22]. In [MY22], OWSGs are constructed from PRSGs with $m \geq c\kappa$ for $c > 1$. (Here, κ is the key-length (i.e., the input length of StateGen), and m is the output length of StateGen (i.e., the number of qubits of ϕ_k).) It can be improved to the construction of OWSGs from PRSGs with $m \geq \log \kappa$.[11] (For a proof, see Appendix C.)

Remark 3.5. Note that statistically-secure OWSGs do not exist. In other words, there exists an unbounded algorithm $\mathcal{A}$ that can break the security of OWSGs as follows:

1. Given $\phi_k^{\otimes t}$ with a certain polynomial t as input, run the shadow tomography algorithm [Aar19] to find k' such that $\Pr[\mathsf{Ver}(k', \phi_k) \to \top] \geq 1 - \frac{1}{\mathrm{poly}(\lambda)}$. If there exists such k', such k' can be found with only a certain polynomial t. If there is no such k', choose k' uniformly at ramdom.
2. Output k'.

3.2 Hardness Amplification for OWSGs

In this subsection, we define a weaker variant called weak one-way states generators (wOWSGs), and show that they are equivalent to OWSGs.

wOWSGs are defined as follows.

Definition 3.6 (Weak one-way states generators (wOWSGs)). *A weak one-way states generator (wOWSG) is a tuple of algorithms* $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$ *defined similarly to OWSGs except that the security is replaced with the following weak security.*

[10] StateGen is actually run t times to generate t copies of ϕ_k, but for simplicity, we just write $\phi_k \leftarrow \mathsf{StateGen}(k)$ only once. This simplification will often be used in this paper.

[11] We thank Luowen Qian for pointing out the fact to us.

Weak Security: *There exists a polynomial p such that for any QPT adversary $\mathcal{A}$ and any polynomial t,*

$$\Pr[\top \leftarrow \mathsf{Ver}(k', \phi_k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k), k' \leftarrow \mathcal{A}(1^\lambda, \phi_k^{\otimes t})] \leq 1 - \frac{1}{p}.$$

We prove that the existence of wOWSGs imply the existence of OWSGs. This is an analogue of Yao's amplification theorem for OWFs in the classical setting [Yao82, Gol01].

Theorem 3.7. *OWSGs exist if and only if wOWSGs exist.*

We observe that the classical proof for OWFs does not extend to OWSGs at least in a straightforward manner. The reason is that the classical proof uses the (obvious) fact that one can deterministically check if a given pair (x, y) satisfies $f(x) = y$ for a OWF f but we do not have such a deterministic verification algorithm for OWSGs.[12] On the other hand, we observe that the proof of a more general hardness amplification theorem for *weakly verifiable puzzles* shown by Canetti, Halevi, and Steiner [CHS05] extends to the quantum setting with minor tweak. Thus we choose to show the quantum analogue of [CHS05], which is more general than [Yao82]. We note that Radian and Sattath [RS19] observed that the proof of [CHS05] extends to the post-quantum setting where the adversary is quantum with essentially the same proof, but what we show is stronger than that since we consider quantum puzzles and answers.

Remark 3.8. We can also consider an even weaker variant of OWSGs where the correctness bound can be much smaller than 1 and we only require there is an inverse polynomial gap between the correctness and security bounds. Unfortunately, we do not know how to prove the equivalence between such a further weaker variant of weak OWSGs and standard OWSGs. To prove this, we will need a quantum analogue of the results of [IJK09, HS11], which are generalization of [CHS05].

First, we define a quantum analogue of weakly verifiable puzzles.

Definition 3.9 (Weakly verifiable quantum puzzles). *A weakly verifiable quantum puzzle is a tuple of algorithms* $(\mathsf{CheckGen}, \mathsf{PuzzleGen}, \mathsf{Ver})$ *as follows.*

- $\mathsf{CheckGen}(1^\lambda) \to k$: *It is a QPT algorithm that, on input the security parameter λ, outputs a classical string k, which we call a "check key".*
- $\mathsf{PuzzleGen}(k) \to \mathsf{puz}$: *It is a QPT algorithm that, on input k, outputs a quantum state* puz*, which we call a "puzzle".*
- $\mathsf{Ver}(\mathsf{ans}, k) \to \top/\bot$: *It is a QPT algorithm that, on input k and a quantum state* ans*, which we call an "answer", outputs $\top$ or $\bot$.*

Remark 3.10. Besides that puz and ans are quantum, another difference from the classical version [CHS05] is that the generation algorithm is divided into CheckGen that generates k and PuzzleGen that generates puz. We define it in this way because we want to consider poly-copy hardness, i.e., the hardness to find a valid answer even given polynomially many copies of the puzzle. If we use a single generation algorithm Gen that generates k and puz simultaneously like [CHS05], it is unclear how to define such a poly-copy hardness.

Remark 3.11. Later, we will see that OWSGs can be seen as weakly verifiable quantum puzzles: CheckGen corresponds to KeyGen of OWSGs, PuzzleGen corresponds to StateGen of OWSGs, and Ver of weakly verifiable quantum puzzles corresponds to Ver of OWSGs. Moreover, ans corresponds to k' of OWSGs,

[12] We remark that the concurrent work by Cao and Xue [CX22] avoids this issue by appropriately modifying the proof strategy.

which is the output of the adversary, i.e., $k' \leftarrow \mathcal{A}(\phi_k^{\otimes t})$. Because k' is a classical bit string, it is enough for our purpose to consider only classical ans, but here we assume ans is quantum to give a more general result. Ver of weakly verifiable quantum puzzles takes k, while Ver of OWSGs takes ϕ_k. This discrepancy can be easily solved by considering that Ver of weakly verifiable quantum puzzles first generates ϕ_k from k and then runs Ver of OWSGs.

Definition 3.12. *We say that an adversary $\mathcal{A}$ (t,ϵ)-solves a weakly verifiable quantum puzzle* $(\mathsf{CheckGen}, \mathsf{PuzzleGen}, \mathsf{Ver})$ *if*

$$\Pr[\top \leftarrow \mathsf{Ver}(\mathsf{ans}, k) : k \leftarrow \mathsf{CheckGen}(1^\lambda), \mathsf{puz} \leftarrow \mathsf{PuzzleGen}(k), \mathsf{ans} \leftarrow \mathcal{A}(\mathsf{puz}^{\otimes t})] \geq \epsilon.$$

Definition 3.13 (Parallel repetition). *For a weakly verifiable quantum puzzle* $(\mathsf{CheckGen}, \mathsf{PuzzleGen}, \mathsf{Ver})$ *and a positive integer n, we define its n-repetition* $(\mathsf{CheckGen}^n, \mathsf{PuzzleGen}^n, \mathsf{Ver}^n)$ *as follows.*

- $\mathsf{CheckGen}^n(1^\lambda) \to (k_1, \dots, k_n)$: *Run* $k_i \leftarrow \mathsf{CheckGen}(1^\lambda)$ *for* $i \in [n]$ *and output* $(k_1, \dots, k_n)$.
- $\mathsf{PuzzleGen}^n(k_1, \dots, k_n) \to (\mathsf{puz}_1, \dots, \mathsf{puz}_n)$: *Run* $\mathsf{puz}_i \leftarrow \mathsf{PuzzleGen}(k_i)$ *for* $i \in [n]$ *and output* $(\mathsf{puz}_1, \dots, \mathsf{puz}_n)$.
- $\mathsf{Ver}^n((\mathsf{ans}_1, \dots, \mathsf{ans}_n), (k_1 \dots, k_n)) \to \top/\bot$: *Run* $\mathsf{Ver}(\mathsf{ans}_i, k_i)$ *for* $i \in [n]$ *and output* $\top$ *if and only if all the execution of* Ver *outputs* $\top$.

Theorem 3.14. *Let n, q, t be polynomials and $\delta \in (0,1)$ be an inverse polynomial in λ. If there exists a QPT adversary $\mathcal{A}$ that (t, δ^n)-solves* $(\mathsf{CheckGen}^n, \mathsf{PuzzleGen}^n, \mathsf{Ver}^n)$*, then there exists a polynomial t' and a QPT adversary $\mathcal{A}'$ that $(t', \delta(1-\frac{1}{q}))$-solves* $(\mathsf{CheckGen}, \mathsf{PuzzleGen}, \mathsf{Ver})$. [13]

Proof of Theorem 3.14 (sketch). Since the proof is similar to that in the classical case ([CHS05, Lemma 1]), we only explain how to modify it. See Appendix F for the full proof.

First, let us recall the proof overview in the classical case where puz and ans are classical. The construction of $\mathcal{A}'$ can be divided into the "preprocessing phase" and "online phase". In the preprocessing phase, $\mathcal{A}'$ finds a "good" prefix $\mathsf{prefix}_{v-1} = (\mathsf{puz}_1, \dots, \mathsf{puz}_{v-1})$ for some $v \in [n]$. (The actual definition of goodness does not matter in this proof sketch.[14]) In the online phase, given a puzzle puz as an instance, $\mathcal{A}'$ repeats the following:

- $\mathcal{A}'$ runs $(\mathsf{ans}_1, \dots, \mathsf{ans}_n) \leftarrow \mathcal{A}(\mathsf{puz}_1, \dots, \mathsf{puz}_{v-1}, \mathsf{puz}, \mathsf{puz}_{v+1}, \dots, \mathsf{puz}_n)$ where $\mathsf{prefix}_{v-1} = (\mathsf{puz}_1, \dots, \mathsf{puz}_{v-1})$ is the "good" prefix found in the preprocessing phase, puz is the given instance, and $(\mathsf{puz}_{v+1}, \dots, \mathsf{puz}_n)$ is generated by $\mathcal{A}'$ itself along with the corresponding check key $(k_{v+1}, \dots, k_n)$.
- $\mathcal{A}'$ runs $\mathsf{Ver}(\mathsf{ans}_i, k_i)$ for $i \in \{v+1, \dots, n\}$ and outputs ans_v if $\mathsf{Ver}(\mathsf{ans}_i, k_i)$ outputs $\top$ for all $i \in \{v+1, \dots, n\}$. (Note that this is possible because $\mathcal{A}'$ generates k_i for $i \in \{v+1, \dots, n\}$ by itself.) Otherwise, it continues running the loop.

If $\mathcal{A}'$ does not halt after running the loop sufficiently many times, it aborts. They show that

- The probability that $\mathcal{A}'$ fails to find a good prefix is at most $\frac{\delta}{6q}$, and

[13] q is a parameter chosen freely that affects the time complexity and the success probability of $\mathcal{A}'$, but here we do not write the time complexity of $\mathcal{A}'$ explicitly, because we are not interested in it as long as it is QPT.

[14] For the readers who want to find the correspondence with the proof in [CHS05], we say that prefix_{v-1} is good if it satisfies Equations (1) and (2) of [CHS05].

- For any "good" prefix, $\mathcal{A}'$ solves the puzzle with probability at least $\delta(1-\frac{5}{6q})$.

Combining the above, we can conclude that the overall probability that $\mathcal{A}'$ solves the puzzle is at least $\delta(1-\frac{1}{q})$.

When we generalize this proof to the quantum setting where puz and ans are quantum, there is an issue that $\mathcal{A}'$ cannot reuse the "good" prefix in the online phase since it consists of *quantum* puzzles, which cannot be copied.[15] To resolve this problem, we observe that we can regard k as a classical description of $\mathsf{puz} \leftarrow \mathsf{PuzzleGen}(k)$. Then our idea is to define a prefix to be a sequence of check keys k instead of puzzles puz.[16] Since k is classical and in particular can be reused many times, the above issue is resolved. Another issue is that the online phase of $\mathcal{A}'$ has to run $\mathcal{A}$ many times where it embeds the problem instance puz into the input of $\mathcal{A}$, but puz is quantum and thus cannot be copied. To resolve this issue, we simply provide sufficiently many copies of puz to $\mathcal{A}'$.[17] Due to this modification, Theorem 3.14 does not preserve the number of copies, i.e., t' should be much larger than t, but we still have $t' = t \cdot \mathrm{poly}(\lambda)$ since $\mathcal{A}'$ runs $\mathcal{A}$ only polynoimally many times.

With the above differences in mind, it is straightforward to extend the proof of [CHS05, Lemma 1] to prove Theorem 3.14. □

Remark 3.15. We remark that the above proof does not work if k is quantum and potentially entangled with puz. (In such a setting, we can only consider single-copy hardness.) A generalization to such a setting would resolve the open problem about the amplification for computational binding of canonical quantum bit commitments asked by Yan [Yan20, Section 12, Problem 5]. In the above proof, it is crucial that we have a classical description of puz as k, and we do not know how to extend it to the case where k is quantum.

We are ready to prove Theorem 3.7.

Proof of Theorem 3.7. The "only if" direction is trivial since any OWSG is also wOWSG. In the following, we show the "if" direction, i.e., OWSGs exist if wOWSGs exist. Let $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$ be a wOWSG. Then there exists a polynomial p such that for any QPT adversary $\mathcal{A}'$ and any polynomial t',

$$\Pr[\top \leftarrow \mathsf{Ver}(k', \phi_k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k), k' \leftarrow \mathcal{A}'(1^\lambda, \phi_k^{\otimes t'})] \leq 1 - \frac{1}{p}.$$

Then we construct a OWSG $(\mathsf{KeyGen}^n, \mathsf{StateGen}^n, \mathsf{Ver}^n)$ as follows, where $n = \mathrm{poly}(\lambda)$ is chosen in such a way that $(1-\frac{1}{2p})^n = 2^{-\Omega(\lambda)}$. (For example, we can set $n = p\lambda$.)

- $\mathsf{KeyGen}^n(1^\lambda) \to (k_1, \dots, k_n)$: Run $k_i \leftarrow \mathsf{KeyGen}^n(1^\lambda)$ for $i \in [n]$ and output $(k_1, \dots, k_n)$.
- $\mathsf{StateGen}^n((k_1, \dots, k_n)) \to (\phi_{k_1}, \dots, \phi_{k_n})$: Run $\phi_{k_i} \leftarrow \mathsf{StateGen}(k_i)$ for $i \in [n]$ and output $(\phi_{k_1}, \dots, \phi_{k_n})$.
- $\mathsf{Ver}^n((k'_1, \dots, k'_n), (\phi_{k_1}, \dots, \phi_{k_n})) \to \top/\bot$: Run $\mathsf{Ver}(k'_i, \phi_{k_i})$ for $i \in [n]$ and output $\top$ if and only if $\mathsf{Ver}(k'_i, \phi_{k_i})$ outputs $\top$ for all $i \in [n]$.

[15] Actually, a similar problem already occurs when searching for a "good" prefix in the preprocessing phase.

[16] We remark that this does not work in the original formulation of (classical) weakly verifiable puzzle in [CHS05] because k and puz are generated simultaneously by a single generation algorithm. For this idea to work, it is important that we divide the generation algorithm in to $\mathsf{CheckGen}$ that generates k and $\mathsf{PuzzleGen}$ that generates puz from k as noted in Remark 3.10.

[17] If $\mathcal{A}$ is a non-uniform adversary with quantum advice, $\mathcal{A}'$ also needs to take sufficiently many copies of the advice of $\mathcal{A}$ as its own advice.

It is clear that $(\mathsf{KeyGen}^n, \mathsf{StateGen}^n, \mathsf{Ver}^n)$ satisfies correctness. Let us next show the security. Toward contradiction, suppose that $(\mathsf{KeyGen}^n, \mathsf{StateGen}^n, \mathsf{Ver}^n)$ is not secure. Then there exists a QPT adversary $\mathcal{A}$ and a polynomial t such that

$$\Pr\left[\top \leftarrow \mathsf{Ver}^n((k'_1,\dots,k'_n),(\phi_{k_1},\dots,\phi_{k_n})) : \begin{array}{l}(k_1,\dots,k_n)\leftarrow \mathsf{KeyGen}^n(1^\lambda),\\ (\phi_{k_1},\dots,\phi_{k_n})\leftarrow \mathsf{StateGen}^n(k_1,\dots,k_n),\\ (k'_1,\dots,k'_n)\leftarrow \mathcal{A}(1^\lambda,\phi_{k_1}^{\otimes t},\dots,\phi_{k_n}^{\otimes t})\end{array}\right]$$

is non-negligible, which, in particular, is larger than $(1-\frac{1}{2p})^n = 2^{-\Omega(\lambda)}$ for infinitely many λ. Since OWSGs can be seen as a weakly verifiable quantum puzzle as noted in Remark 3.11, by setting $q := 2p$ and $\delta := 1-\frac{1}{2p}$ in Theorem 3.14, there is a QPT algorithm $\mathcal{A}'$ and a polynomial t' such that

$$\begin{aligned}&\Pr[\top \leftarrow \mathsf{Ver}(k',\phi_k) : k\leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k), k'\leftarrow \mathcal{A}'(1^\lambda,\phi_k^{\otimes t'})]\\ &\geq \delta\left(1-\frac{1}{q}\right) = \left(1-\frac{1}{2p}\right)\left(1-\frac{1}{2p}\right) > 1-\frac{1}{p}\end{aligned}$$

for infinitely many λ. This contradicts the assumption. Therefore $(\mathsf{KeyGen}^n, \mathsf{StateGen}^n, \mathsf{Ver}^n)$ is a OWSG. □

4 QDSs

In this section, we first define QDSs (Section 4.1), and show that one-time-secure QDSs can be extended to q-time-secure ones (Section 4.2). We then show that one-time-secure QDSs are equivalent to OWSGs (Section 4.3).

4.1 Definition of QDSs

Quantum digital signatures are defined as follows.

Definition 4.1 (Quantum digital signatures (QDSs) [MY22]). *A quantum digital signature (QDS) scheme is a set of algorithms* $(\mathsf{SKGen}, \mathsf{PKGen}, \mathsf{Sign}, \mathsf{Ver})$ *such that*

- $\mathsf{SKGen}(1^\lambda) \to \mathsf{sk}$: *It is a QPT algorithm that, on input the security parameter* λ*, outputs a classical secret key* sk.
- $\mathsf{PKGen}(\mathsf{sk}) \to \mathsf{pk}$: *It is a QPT algorithm that, on input* sk*, outputs a quantum public key* pk.
- $\mathsf{Sign}(\mathsf{sk}, m) \to \sigma$: *It is a QPT algorithm that, on input* sk *and a message* m*, outputs a classical signature* σ.
- $\mathsf{Ver}(\mathsf{pk}, m, \sigma) \to \top/\bot$: *It is a QPT algorithm that, on input* pk*,* m*, and* σ*, outputs* $\top/\bot$.

We require the correctness and the security as follows.

Correctness: *For any* m*,*

$$\Pr\left[\top \leftarrow \mathsf{Ver}(\mathsf{pk}, m, \sigma) : \begin{array}{l}\mathsf{sk}\leftarrow \mathsf{SKGen}(1^\lambda),\\ \mathsf{pk}\leftarrow \mathsf{PKGen}(\mathsf{sk}),\\ \sigma\leftarrow \mathsf{Sign}(\mathsf{sk}, m)\end{array}\right] \geq 1-\mathsf{negl}(\lambda).$$

q-time security: *Let us consider the following security game,* Exp*, between a challenger $\mathcal{C}$ and a QPT adversary $\mathcal{A}$:*

1. *$\mathcal{C}$ runs* $\mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda)$.
2. *$\mathcal{C}$ runs* $\mathsf{pk} \leftarrow \mathsf{PKGen}(\mathsf{sk})$ *t times, and sends* $\mathsf{pk}^{\otimes t}$ *to $\mathcal{A}$.*
3. *For $i = 1$ to q, do:*
   (a) *$\mathcal{A}$ sends a message $m^{(i)}$ to $\mathcal{C}$.*
   (b) *$\mathcal{C}$ runs* $\sigma^{(i)} \leftarrow \mathsf{Sign}(\mathsf{sk}, m^{(i)})$*, and sends $\sigma^{(i)}$ to $\mathcal{A}$.*
4. *$\mathcal{A}$ sends σ' and m' to $\mathcal{C}$.*
5. *$\mathcal{C}$ runs* $\mathsf{pk} \leftarrow \mathsf{PKGen}(\mathsf{sk})$ *and* $v \leftarrow \mathsf{Ver}(\mathsf{pk}, m', \sigma')$*. If $m' \notin \{m^{(1)}, \dots, m^{(q)}\}$ and $v = \top$, the output of the game is 1. Otherwise, the output of the game is 0.*

For any QPT adversary $\mathcal{A}$ and any polynomial t, $\Pr[\mathsf{Exp} = 1] \leq \mathsf{negl}(\lambda)$.

Remark 4.2. By using the shadow tomography, we can show that statistically-secure QDSs do not exist. For a proof, see Appendix D.

4.2 Extension to q-time Security

Theorem 4.3. *If one-time-secure QDSs exist, then q-time-secure QDSs exist for any polynomial q.*

Proof of Theorem 4.3. The proof is similar to the one-time to q-time conversion for attribute-based encryption in [ISV+17].[18]

Let $(\mathsf{1DS.SKGen}, \mathsf{1DS.PKGen}, \mathsf{1DS.Sign}, \mathsf{1DS.Ver})$ be a one-time-secure QDS scheme. For a polynomial q, we construct a q-time-secure QDS scheme $(\mathsf{qDS.SKGen}, \mathsf{qDS.PKGen}, \mathsf{qDS.Sign}, \mathsf{qDS.Ver})$ as follows.

- $\mathsf{qDS.SKGen}(1^\lambda) \to \mathsf{sk}$: Run $\mathsf{sk}_{a,b} \leftarrow \mathsf{1DS.SKGen}(1^\lambda)$ for $a \in [\lambda]$ and $b \in [q^2]$ and output $\mathsf{sk} := (\mathsf{sk}_{a,b})_{a\in[\lambda],b\in[q^2]}$.
- $\mathsf{qDS.PKGen}(\mathsf{sk}) \to \mathsf{pk}$: Parse $\mathsf{sk} = (\mathsf{sk}_{a,b})_{a\in[\lambda],b\in[q^2]}$, run $\mathsf{pk}_{a,b} \leftarrow \mathsf{1DS.PKGen}(\mathsf{sk}_{a,b})$ for $a \in [\lambda]$ and $b \in [q^2]$ and output $\mathsf{pk} := \bigotimes_{a\in[\lambda],b\in[q^2]} \mathsf{pk}_{a,b}$.
- $\mathsf{qDS.Sign}(\mathsf{sk}, m) \to \sigma$: Parse $\mathsf{sk} = (\mathsf{sk}_{a,b})_{a\in[\lambda],b\in[q^2]}$, choose $b_a \leftarrow [q^2]$ for $a \in [\lambda]$, run $\sigma_a \leftarrow \mathsf{1DS.Sign}(\mathsf{sk}_{a,b_a}, m)$ for $a \in [\lambda]$, and output $\sigma = (b_a, \sigma_a)_{a\in[\lambda]}$.
- $\mathsf{qDS.Ver}(\mathsf{pk}, m, \sigma) \to \top/\bot$: Parse $\mathsf{pk} = \bigotimes_{a\in[\lambda],b\in[q^2]} \mathsf{pk}_{a,b}$ and $\sigma = (b_a, \sigma_a)_{a\in[\lambda]}$, run $\mathsf{1DS.Ver}(\mathsf{pk}_{a,b_a}, m, \sigma_a)$ for $a \in [\lambda]$, and output $\top$ if and only if all execution of $\mathsf{1DS.Ver}$ output $\top$.

We show that the above scheme satisfies q-time security. Let $\mathcal{A}$ be an adversary against q-time security of the above scheme. In the q-time security Exp as defined in Definition 4.1, let $\sigma^{(i)} = (b_a^{(i)}, \sigma_a^{(i)})_{a\in[\lambda]}$ be the i-th signature generated by $\mathcal{C}$ and $\sigma' = (b'_a, \sigma'_a)_{a\in[\lambda]}$ be $\mathcal{A}$'s final output. Let Good be the event that there is $a \in [\lambda]$ such that $b_a^{(i)}$ is distinct for all $i \in [q]$. Then we can show that $\Pr[\mathsf{Good}] = 1 - \mathsf{negl}(\lambda)$ similarly to

[18] We also note that the proof works also for the classical setting.

[ISV+17, Lemma 1]. Specifically, this is shown as follows. For each $a \in [\lambda]$, let Bad_a be the event that $b_a^{(i)}$ is *not* distinct for $i \in [q]$. Then by a simple combinatorial argument, we can see that

$$\begin{aligned}\Pr[\mathsf{Bad}_a] &= 1 - \left(\frac{q^2(q^2-1)\cdots(q^2-q+1)}{(q^2)^q}\right) \\ &\leq 1 - \left(1 - \frac{q-1}{q^2}\right)^q.\end{aligned}$$

for each $a \in [\lambda]$. Thus, we have

$$\begin{aligned}\Pr[\mathsf{Good}] &= 1 - \prod_{a\in[\lambda]} \Pr[\mathsf{Bad}_a] \\ &\geq 1 - \left(1 - \left(1 - \frac{q-1}{q^2}\right)^q\right)^\lambda \\ &= 1 - \left(1 - \left(1 - \frac{1}{q} + \frac{1}{q^2}\right)^q\right)^\lambda \\ &\geq 1 - (1-e^{-1})^\lambda \\ &= 1 - \mathsf{negl}(\lambda).\end{aligned}$$

Conditioned on Good occurs and $\mathcal{A}$ wins (i.e., Exp returns 1), if we uniformly choose $a^* \leftarrow [\lambda]$ and $b^* \leftarrow [q^2]$, then the probability that $b_{a^*}^{(i)}$ is distinct for all $i \in [q]$ and $b^* = b'_{a^*}$ is at least $\frac{1}{q^2\lambda}$. Thus, by randomly guessing $a^* \leftarrow [\lambda]$ and $b^* \leftarrow [q^2]$ and embedding a problem instance of $\mathsf{1QDS}$ into the corresponding position, we can reduce the q-time security of qQDS to the one-time security of $\mathsf{1QDS}$. That is, we can construct an adversary $\mathcal{B}$ against the one-time security of $\mathsf{1QDS}$ as follows:

$\mathcal{B}(\mathsf{pk}^{\otimes t})$**:** Choose $a^* \leftarrow [\lambda]$ and $b^* \leftarrow [q^2]$, generate $\mathsf{sk}_{a,b} \leftarrow \mathsf{1DS.SKGen}(1^\lambda)$ and run $\mathsf{pk}_{a,b} \leftarrow \mathsf{1DS.PKGen}(\mathsf{sk}_{a,b})$ times to get $\mathsf{pk}_{a,b}^{\otimes t}$ for $(a,b) \in ([\lambda] \times [q^2]) \setminus \{(a^*,b^*)\}$, set $\mathsf{pk}_{a^*,b^*}^{\otimes t} := \mathsf{pk}^{\otimes t}$, and sends $\mathsf{pk}^{\otimes t} := \bigotimes_{a\in[\lambda],b\in[q^2]} \mathsf{pk}_{a,b}^{\otimes t}$ to $\mathcal{A}$. When $\mathcal{A}$ sends $m^{(i)}$ for $i \in [q]$, choose $b_a^{(i)}$ for $a \in [\lambda]$, generates $\sigma_a \leftarrow \mathsf{1DS.Sign}(\mathsf{sk}_{a,b_a^{(i)}}, m)$ for $a \in [q] \setminus \{a^*\}$ and σ_{a^*} in one of the following way:

- If $b_{a^*}^{(i)} = b^*$ and $\mathcal{B}$ has never sent m to the external challenger, send m to the external challenger and let $\sigma_{a^*}^{(i)}$ be the returned signature;
- If $b_{a^*}^{(i)} = b^*$ and $\mathcal{B}$ has sent m to the external challenger before, immediately abort;
- If $b_{a^*}^{(i)} \neq b^*$, generate $\sigma_{a^*} \leftarrow \mathsf{1DS.Sign}(\mathsf{sk}_{a^*,b_{a^*}^{(i)}}, m)$. Note that this is possible without the help of the external challenger since $\mathcal{B}$ generated $\mathsf{sk}_{a,b}^{(i)}$ by itself for $(a,b) \neq (a^*,b^*)$.

Return $\sigma^{(i)} := (b_a^{(i)}, \sigma_a^{(i)})_{a\in[\lambda]}$ to $\mathcal{A}$. When $\mathcal{A}$ sends $\sigma' = (b'_a, \sigma'_a)_{a\in[\lambda]}$, send σ'_{a^*} to the external challenger if $b'_{a^*} = b^*$ and otherwise abort.

As already discussed, Good occurs with probability $1 - \mathsf{negl}(\lambda)$ and conditioned on that Good occurs, the probability that $\mathcal{B}$ wins is at least $\frac{1}{q^2\lambda}$ times the probability that $\mathcal{A}$ wins. Since q is polynomial, the reduction loss is polynomial and thus qDS is q-time secure if qDS is one-time secure. □

4.3 Equivalence of OWSGs and QDSs

Theorem 4.4. *OWSGs exist if and only if one-time-secure QDSs exist.*

Remark 4.5. By using the equivalence between OWSGs and wOWSGs (Theorem 3.7), the result that one-time-secure QDSs imply OWSGs can be improved to a stronger result (with a similar proof) that one-time-secure QDSs with weak security imply OWSGs. Here, the weak security of QDSs means that there exists a polynomial p such that for any QPT adversary $\mathcal{A}$ and any polynomial t, $\Pr[\mathsf{Exp}=1] \leq 1-\frac{1}{p}$.

Proof of Theorem 4.4. Let us first show that one-time-secure QDSs imply OWSGs. Let $(\mathsf{DS.SKGen}, \mathsf{DS.PKGen}, \mathsf{DS.Sign}, \mathsf{DS.Ver})$ be a one-time-secure QDS scheme for a single-bit message. From it, we construct a OWSG as follows.

- $\mathsf{KeyGen}(1^\lambda) \to k$: Run $\mathsf{sk} \leftarrow \mathsf{DS.SKGen}(1^\lambda)$. Output $k := \mathsf{sk}$.
- $\mathsf{StateGen}(k) \to \phi_k$: Parse $k = \mathsf{sk}$. Run $\mathsf{pk} \leftarrow \mathsf{DS.PKGen}(\mathsf{sk})$. Output $\phi_k := \mathsf{pk}$.
- $\mathsf{Ver}(k', \phi_k) \to \top/\bot$: Parse $\phi_k = \mathsf{pk}$. Run $\sigma \leftarrow \mathsf{DS.Sign}(k', 1)$. Run $v \leftarrow \mathsf{DS.Ver}(\mathsf{pk}, 1, \sigma)$. Output v.

Correctness is clear. Let us show the security. Assume that it is not secure. It means that there exists a QPT adversary $\mathcal{A}$ and a polynomial t such that

$$\Pr\left[\top \leftarrow \mathsf{DS.Ver}(\mathsf{pk}, 1, \sigma) : \begin{array}{c} \mathsf{sk} \leftarrow \mathsf{DS.SKGen}(1^\lambda) \\ \mathsf{pk} \leftarrow \mathsf{DS.PKGen}(\mathsf{sk}) \\ \mathsf{sk}' \leftarrow \mathcal{A}(\mathsf{pk}^{\otimes t}) \\ \sigma \leftarrow \mathsf{DS.Sign}(\mathsf{sk}', 1) \end{array}\right] \geq \frac{1}{\mathrm{poly}(\lambda)}.$$

From such $\mathcal{A}$, let us construct a QPT adversary $\mathcal{B}$ that breaks the security of the QDS as follows: On input $\mathsf{pk}^{\otimes t}$, it runs $\mathsf{sk}' \leftarrow \mathcal{A}(\mathsf{pk}^{\otimes t})$. It then runs $\sigma \leftarrow \mathsf{DS.Sign}(\mathsf{sk}', 1)$, and outputs $(1, \sigma)$. It is clear that the probability that σ is accepted as a valid signature of the message 1 is equal to the left-hand-side of the above inequality. Hence the security of the QDS is broken by $\mathcal{B}$.

Let us next give a proof that OWSGs imply QDSs. The construction is a "quantum version" of the Lamport signatures [Lam79], and is similar to that given in [MY22]. Let $(\mathsf{OWSG.KeyGen}, \mathsf{OWSG.StateGen}, \mathsf{OWSG.Ver})$ be a OWSG. From it, we construct a one-time-secure QDS for a single-bit message as follows.

- $\mathsf{SKGen}(1^\lambda) \to \mathsf{sk}$: Run $k_0 \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)$. Run $k_1 \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)$. Output $\mathsf{sk} := (\mathsf{sk}_0, \mathsf{sk}_1)$, where $\mathsf{sk}_b := k_b$ for $b \in \{0,1\}$.
- $\mathsf{PKGen}(\mathsf{sk}) \to \mathsf{pk}$: Parse $\mathsf{sk} = (\mathsf{sk}_0, \mathsf{sk}_1)$, where $\mathsf{sk}_b = k_b$ for $b \in \{0,1\}$. Run $\phi_{k_b} \leftarrow \mathsf{OWSG.StateGen}(k_b)$ for each $b \in \{0,1\}$. Output $\mathsf{pk} := (\mathsf{pk}_0, \mathsf{pk}_1)$, where $\mathsf{pk}_b := \phi_{k_b}$ for $b \in \{0,1\}$.
- $\mathsf{Sign}(\mathsf{sk}, m) \to \sigma$: Parse $\mathsf{sk} = (\mathsf{sk}_0, \mathsf{sk}_1)$, where $\mathsf{sk}_b = k_b$ for $b \in \{0,1\}$. Output $\sigma := \mathsf{sk}_m$.
- $\mathsf{Ver}(\mathsf{pk}, m, \sigma) \to \top/\bot$: Parse $\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1)$, where $\mathsf{pk}_b = \phi_{k_b}$ for $b \in \{0,1\}$. Run $v \leftarrow \mathsf{OWSG.Ver}(\sigma, \phi_{k_m})$. Output v.

It is clear that this construction satisfies correctness. Let us next show the security. Let us consider the following security game between the challenger $\mathcal{C}$ and a QPT adversary $\mathcal{A}$:

1. $\mathcal{C}$ runs $k_0 \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)$. $\mathcal{C}$ runs $k_1 \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)$.
2. $\mathcal{C}$ runs $\phi_{k_b} \leftarrow \mathsf{OWSG.StateGen}(k_b)$ $t+1$ times for $b \in \{0,1\}$. $\mathcal{C}$ sends $(\phi_{k_0}^{\otimes t}, \phi_{k_1}^{\otimes t})$ to $\mathcal{A}$.
3. $\mathcal{A}$ sends $m \in \{0,1\}$ to $\mathcal{C}$.
4. $\mathcal{C}$ sends k_m to $\mathcal{A}$.
5. $\mathcal{A}$ sends σ to $\mathcal{C}$.
6. $\mathcal{C}$ runs $v \leftarrow \mathsf{OWSG.Ver}(\sigma, \phi_{k_{m\oplus 1}})$. If the result is $\top$, $\mathcal{C}$ outputs 1. Otherwise, $\mathcal{C}$ outputs 0.

Assume that our construction is not one-time secure, which means that $\Pr[\mathcal{C} \to 1]$ is non-negligible for a QPT $\mathcal{A}$ and a polynomial t. Then, we can construct a QPT adversary $\mathcal{B}$ that breaks the security of the OWSG as follows. (Here, $\mathcal{C}'$ is the challenger of the security game of the OWSG.)

1. $\mathcal{C}'$ runs $k \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)$. $\mathcal{C}'$ runs $\phi_k \leftarrow \mathsf{OWSG.StateGen}(k)$ $t+1$ times. $\mathcal{C}'$ sends $\phi_k^{\otimes t}$ to $\mathcal{B}$.
2. $\mathcal{B}$ chooses $r \leftarrow \{0,1\}$. $\mathcal{B}$ runs $k' \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)$. $\mathcal{B}$ runs $\phi_{k'} \leftarrow \mathsf{OWSG.StateGen}(k')$ t times. If $r=0$, $\mathcal{B}$ sends $(\phi_k^{\otimes t}, \phi_{k'}^{\otimes t})$ to $\mathcal{A}$. If $r=1$, $\mathcal{B}$ sends $(\phi_{k'}^{\otimes t}, \phi_k^{\otimes t})$ to $\mathcal{A}$.
3. $\mathcal{A}$ sends $m \in \{0,1\}$ to $\mathcal{B}$.
4. If $r=m$, $\mathcal{B}$ aborts. If $r \neq m$, $\mathcal{B}$ sends k' to $\mathcal{A}$.
5. $\mathcal{A}$ sends σ to $\mathcal{B}$.
6. $\mathcal{B}$ sends σ to $\mathcal{C}'$.
7. $\mathcal{C}'$ runs $v \leftarrow \mathsf{OWSG.Ver}(\sigma, \phi_k)$. If $v=\top$, $\mathcal{C}'$ outputs 1. Otherwise, $\mathcal{C}'$ outputs 0.

By a straightforward calculation, which is given below,

$$\Pr[\mathcal{C}' \to 1] = \frac{1}{2}\Pr[\mathcal{C} \to 1]. \tag{2}$$

Therefore, if $\Pr[\mathcal{C} \to 1]$ is non-negligible, $\Pr[\mathcal{C}' \to 1]$ is also non-negligible, which means that $\mathcal{B}$ breaks the security of the OWSG.

Let us show Eq. (2). For simplicity, we write $\Pr[k]$ to mean $\Pr[k \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)]$. Then,

$$\begin{aligned}
&\Pr[\mathcal{C}' \to 1] \\
=& \sum_{k,k',\sigma} \Pr[k]\Pr[k']\frac{1}{2}\Pr[1 \leftarrow \mathcal{A}(\phi_k^{\otimes t}, \phi_{k'}^{\otimes t})]\Pr[\sigma \leftarrow \mathcal{A}(k')]\Pr[\top \leftarrow \mathsf{OWSG.Ver}(\sigma, \phi_k)] \\
&+ \sum_{k,k',\sigma} \Pr[k]\Pr[k']\frac{1}{2}\Pr[0 \leftarrow \mathcal{A}(\phi_{k'}^{\otimes t}, \phi_k^{\otimes t})]\Pr[\sigma \leftarrow \mathcal{A}(k')]\Pr[\top \leftarrow \mathsf{OWSG.Ver}(\sigma, \phi_k)] \\
=& \sum_{k,k',\sigma} \Pr[k]\Pr[k']\frac{1}{2}\Pr[1 \leftarrow \mathcal{A}(\phi_k^{\otimes t}, \phi_{k'}^{\otimes t})]\Pr[\sigma \leftarrow \mathcal{A}(k')]\Pr[\top \leftarrow \mathsf{OWSG.Ver}(\sigma, \phi_k)] \\
&+ \sum_{k,k',\sigma} \Pr[k]\Pr[k']\frac{1}{2}\Pr[0 \leftarrow \mathcal{A}(\phi_k^{\otimes t}, \phi_{k'}^{\otimes t})]\Pr[\sigma \leftarrow \mathcal{A}(k)]\Pr[\top \leftarrow \mathsf{OWSG.Ver}(\sigma, \phi_{k'})] \\
=& \frac{1}{2}\Pr[\mathcal{C} \to 1].
\end{aligned}$$

□

5 Quantum Money

In this section, we first define private-key quantum money schemes (Section 5.1). We then construct OWSGs from quantum money schemes with pure money states (Section 5.2). We also show that OWSGs can be constructed from quantum money schemes where the verification algorithms satisfy a certain symmetric property (Section 5.3).

5.1 Definition of Private-key Quantum Money

Private-key quantum money schemes are defined as follows.

Definition 5.1 (Private-key quantum money [JLS18, AC12]). *A private-key quantum money scheme is a set of algorithms* $(\mathsf{KeyGen}, \mathsf{Mint}, \mathsf{Ver})$ *such that*

- $\mathsf{KeyGen}(1^\lambda) \to k$: *It is a QPT algorithm that, on input the security parameter λ, outputs a classical secret key k.*
- $\mathsf{Mint}(k) \to \$_k$: *It is a QPT algorithm that, on input k, outputs an m-qubit quantum state $\$_k$.*
- $\mathsf{Ver}(k, \rho) \to \top/\bot$: *It is a QPT algorithm that, on input k and a quantum state ρ, outputs $\top/\bot$.*

We require the following correctness and security.

Correctness:

$$\Pr[\top \leftarrow \mathsf{Ver}(k, \$_k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \$_k \leftarrow \mathsf{Mint}(k)] \geq 1 - \mathsf{negl}(\lambda).$$

Security: *For any QPT adversary $\mathcal{A}$ and any polynomial t,*

$$\Pr[\mathsf{Count}(k, \xi) \geq t + 1 : k \leftarrow \mathsf{KeyGen}(1^\lambda), \$_k \leftarrow \mathsf{Mint}(k), \xi \leftarrow \mathcal{A}(1^\lambda, \$_k^{\otimes t})] \leq \mathsf{negl}(\lambda),$$

where ξ is a quantum state on ℓ registers, $R_1, ..., R_\ell$, each of which is of m qubits, and Count *is the following QPT algorithm: on input ξ, it runs $\top/\bot \leftarrow \mathsf{Ver}(k, \xi_j)$ for each $j \in [1, 2, ..., \ell]$, where $\xi_j := Tr_{R_1,...,R_{j-1},R_{j+1},...,R_\ell}(\xi)$, and outputs the total number of $\top$.*

Remark 5.2. Private-key quantum money schemes are constructed from PRSGs [JLS18].

Remark 5.3. As is shown in [Aar19], private-key quantum money schemes are broken by an unbounded adversary, and therefore statistically-secure private-key quantum money schemes do not exist. (The idea is as follows: the unbounded adversary first finds all $\{k_i\}_i$ such that $\mathsf{Ver}(k_i, \$_k)$ is large with the shadow tomography, and then searches a state ρ by the brute-force such that $\mathsf{Ver}(k_i, \rho)$ is close to $\mathsf{Ver}(k_i, \$_k)$ FOR ALL i. Finally, the adversary outputs many copies of ρ.)

5.2 OWSGs from Quantum Money with Pure Money States

Theorem 5.4. *If private-key quantum money schemes with pure quantum money states exist, then OWSGs exist.*

Remark 5.5. For example, the private-key quantum money scheme of [JLS18] has pure quantum money states.

Remark 5.6. By using the equivalence between OWSGs and wOWSGs (Theorem 3.7), this result can be improved to a stronger result (with a similar proof) that private-key quantum money schemes with pure quantum money states and with weak security imply OWSGs. Here, the weak security means that there exists a polynomial p such that for any QPT adversary $\mathcal{A}$ and any polynomial t,

$$\Pr[\mathsf{Count}(k,\xi) \geq t+1 : k \leftarrow \mathsf{KeyGen}(1^\lambda), \$_k \leftarrow \mathsf{Mint}(k), \xi \leftarrow \mathcal{A}(1^\lambda, \$_k^{\otimes t})] \leq 1 - \frac{1}{p}.$$

Proof of Theorem 5.4. Let $(\mathsf{QM.KeyGen}, \mathsf{QM.Mint}, \mathsf{QM.Ver})$ be a private-key quantum money scheme with pure money states. From it, we construct a OWSG as follows.

- $\mathsf{KeyGen}(1^\lambda) \to k$: Run $k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)$. Output k.
- $\mathsf{StateGen}(k) \to \phi_k$: Run $|\$_k\rangle \leftarrow \mathsf{QM.Mint}(k)$. Output $\phi_k := |\$_k\rangle\langle\$_k|$.
- $\mathsf{Ver}(k', \phi_k) \to \top/\bot$: Parse $\phi_k = |\$_k\rangle\langle\$_k|$. Measure $|\$_k\rangle$ with the basis $\{|\$_{k'}\rangle\langle\$_{k'}|, I - |\$_{k'}\rangle\langle\$_{k'}|\}$, and output $\top$ if the first result is obtained. Output $\bot$ if the second result is obtained. (This measurement is done in the following way: generate $U(|k'\rangle|0...0\rangle) = |\$_{k'}\rangle|\eta_{k'}\rangle$, and discard the first register. Then apply $U^\dagger$ on $|\$_k\rangle|\eta_{k'}\rangle$, and measure all qubits in the computationl basis. If the result is $k'0...0$, accept. Otherwise, reject.)

The correctness is clear. Let us show the security. Assume that it is not secure. Then, there exists a QPT adversary $\mathcal{A}$, a polynomial t, and a polynomial p such that

$$\sum_{k,k'} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] |\langle\$_k|\$_{k'}\rangle|^2 \geq \frac{1}{p}.$$

Define the set

$$S := \left\{(k,k') \;\middle|\; |\langle\$_k|\$_{k'}\rangle|^2 \geq \frac{1}{2p}\right\}.$$

Then, we have

$$\sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] > \frac{1}{2p}.$$

This is shown as follows.

$$\begin{aligned}
\frac{1}{p} &\leq \sum_{k,k'} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] |\langle\$_k|\$_{k'}\rangle|^2 \\
&= \sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] |\langle\$_k|\$_{k'}\rangle|^2 \\
&\quad + \sum_{(k,k')\notin S} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] |\langle\$_k|\$_{k'}\rangle|^2 \\
&< \sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] + \frac{1}{2p}.
\end{aligned}$$

Let us also define

$$T := \left\{k \;\middle|\; \Pr[\top \leftarrow \mathsf{QM.Ver}(k, |\$_k\rangle)] \geq 1 - \frac{1}{8p}\right\}.$$

Then,

$$\sum_{k\in T}\Pr[k\leftarrow \mathsf{QM.KeyGen}(1^\lambda)] > 1-\mathsf{negl}(\lambda).$$

This is shown as follows.

$$\begin{aligned}
1-\mathsf{negl}(\lambda) &\leq \sum_{k}\Pr[k\leftarrow \mathsf{QM.KeyGen}(1^\lambda)]\Pr[\top\leftarrow \mathsf{QM.Ver}(k,|\$_k\rangle)]\\
&= \sum_{k\in T}\Pr[k\leftarrow \mathsf{QM.KeyGen}(1^\lambda)]\Pr[\top\leftarrow \mathsf{QM.Ver}(k,|\$_k\rangle)]\\
&\quad+\sum_{k\notin T}\Pr[k\leftarrow \mathsf{QM.KeyGen}(1^\lambda)]\Pr[\top\leftarrow \mathsf{QM.Ver}(k,|\$_k\rangle)]\\
&< \sum_{k\in T}\Pr[k\leftarrow \mathsf{QM.KeyGen}(1^\lambda)]\\
&\quad+\Big(1-\frac{1}{8p}\Big)\Big(1-\sum_{k\in T}\Pr[k\leftarrow \mathsf{QM.KeyGen}(1^\lambda)]\Big).
\end{aligned}$$

Here, the first inequality is from the correctness of the quantum money scheme.

Let us fix (k,k') such that $(k,k')\in S$ and $k\in T$. The probability of having such (k,k') is, from the union bound,

$$\begin{aligned}
\sum_{(k,k')\in S\wedge k\in T}\Pr[k\leftarrow \mathsf{QM.KeyGen}(1^\lambda)]\Pr[k'\leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] &> \frac{1}{2p}+1-\mathsf{negl}(\lambda)-1\\
&= \frac{1}{2p}-\mathsf{negl}(\lambda).
\end{aligned}$$

From the $\mathcal{A}$, we construct a QPT adversary $\mathcal{B}$ that breaks the security of the private-key quantum money scheme as follows: On input $|\$_k\rangle^{\otimes t}$, it runs $k'\leftarrow\mathcal{A}(|\$_k\rangle^{\otimes t})$. It then runs $|\$_{k'}\rangle\leftarrow\mathsf{QM.Mint}(k')$ ℓ times, where ℓ is a polynomial specified later, and outputs $\xi:=|\$_{k'}\rangle^{\otimes\ell}$. Let us show that thus defined $\mathcal{B}$ breaks the security of the private-key quantum money scheme. Let v_j be the bit that is 1 if the output of $\mathsf{QM.Ver}(k,\xi_j)$ is $\top$, and is 0 otherwise. Then, for any (k,k') such that $(k,k')\in S$ and $k\in T$,

$$\begin{aligned}
\Pr[v_j=1] &= \Pr[\top\leftarrow \mathsf{QM.Ver}(k,\xi_j)]\\
&= \Pr[\top\leftarrow \mathsf{QM.Ver}(k,|\$_{k'}\rangle)]\\
&\geq \Pr[\top\leftarrow \mathsf{QM.Ver}(k,|\$_k\rangle)]-\sqrt{1-\frac{1}{2p}}\\
&\geq 1-\frac{1}{8p}-\sqrt{1-\frac{1}{2p}}\\
&\geq \frac{1}{8p}
\end{aligned}$$

for each $j\in[1,2,...,\ell]$. Here, in the first inequality, we have used the fact that $\Pr[1\leftarrow\mathcal{D}(|\$_k\rangle)]-\Pr[1\leftarrow\mathcal{D}(|\$_{k'}\rangle)]\leq\sqrt{1-\frac{1}{2p}}$ for any algorithm $\mathcal{D}$. This is because $|\langle\$_k|\$_{k'}\rangle|^2\geq\frac{1}{2p}$ for any $(k,k')\in S$.[19] Moreover,

[19] Due to the relation between the fidelity and the trace distance, we have $\frac{1}{2}\||\$_k\rangle\langle\$_k|-|\$_{k'}\rangle\langle\$_{k'}|\|_1\leq\sqrt{1-|\langle\$_k|\$_{k'}\rangle|^2}$, which means that $\langle\$_k|\Pi|\$_k\rangle-\langle\$_{k'}|\Pi|\$_{k'}\rangle\leq\sqrt{1-|\langle\$_k|\$_{k'}\rangle|^2}$ for any POVM element Π.

in the second inequality, we have used the fact that $\Pr[\top \leftarrow \mathsf{QM.Ver}(k, |\$_k\rangle)] \geq 1 - \frac{1}{8p}$ for any $k \in T$. Finally, in the last inequality, we have used the Bernoulli's inequality.[20]

Let us take $\ell \geq \max(16p(t+1), 16^2p^3)$. Then, for any (k, k') such that $(k, k') \in S$ and $k \in T$,

$$\begin{aligned}
\Pr[\mathsf{Count}(k, |\$_{k'}\rangle^{\otimes \ell}) \geq t+1] &= \Pr\Big[\sum_{j=1}^{\ell} v_j \geq t+1\Big] \\
&\geq \Pr\Big[\sum_{j=1}^{\ell} v_j \geq \frac{\ell}{16p}\Big] \\
&= \Pr\Big[\sum_{j=1}^{\ell} v_j \geq \frac{\ell}{8p} - \frac{\ell}{16p}\Big] \\
&\geq \Pr\Big[\sum_{j=1}^{\ell} v_j \geq \mathbb{E}(\sum_{j=1}^{\ell} v_j) - \frac{\ell}{16p}\Big] \\
&\geq 1 - 2\exp\Big[-\frac{2\ell}{16^2p^2}\Big] \\
&\geq 1 - 2e^{-2p}.
\end{aligned}$$

Here, in the third inequality, we have used Hoeffding's inequality. The probability that $\mathcal{B}$ breaks the security of the quantum money scheme is therefore

$$\begin{aligned}
&\sum_{k,k'} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] \Pr[\mathsf{Count}(k, |\$_{k'}\rangle^{\otimes \ell}) \geq t+1] \\
\geq &\sum_{(k,k') \in S \wedge k \in T} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] \Pr[\mathsf{Count}(k, |\$_{k'}\rangle^{\otimes \ell}) \geq t+1] \\
\geq &\ (1 - 2e^{-2p}) \sum_{(k,k') \in S \wedge k \in T} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\$_k\rangle^{\otimes t})] \\
\geq &\ (1 - 2e^{-2p}) \Big(\frac{1}{2p} - \mathsf{negl}(\lambda)\Big),
\end{aligned}$$

which is non-negligible. The $\mathcal{B}$ therefore breaks the security of the private-key quantum money scheme. □

5.3 OWSGs from Quantum Money with Symmetric Verifiability

We consider the following restriction for quantum money.

Definition 5.7 (Symmetric-verifiability). *We say that a private-key quantum money scheme satisfies the symmetric-verifiability if* $\Pr[\top \leftarrow \mathsf{Ver}(k, \$_{k'})] = \Pr[\top \leftarrow \mathsf{Ver}(k', \$_k)]$ *for all* $k \neq k'$.

Remark 5.8. For example, if all money states are pure, and $\mathsf{Ver}(\alpha, \rho)$ is the following algorithm, the symmetric-verifiability is satisfied: Measure ρ with the basis $\{|\$_\alpha\rangle\langle\$_\alpha|, I - |\$_\alpha\rangle\langle\$_\alpha|\}$. If the first result is obtained, output $\top$. Otherwise, output $\bot$.

Theorem 5.9. *If private-key quantum money schemes with symmetric-verifiability exist, then OWSGs exist.*

[20] $(1+x)^r \leq 1 + rx$ for any real r and x such that $0 \leq r \leq 1$ and $x \geq -1$.

Remark 5.10. By using the equivalence between OWSGs and wOWSGs (Theorem 3.7), this result can be improved to a stronger result (with a similar proof) that private-key quantum money schemes with symmetric-verifiability and with weak security imply OWSGs. Here, the weak security means that there exists a polynomial p such that for any QPT adversary $\mathcal{A}$ and any polynomial t,

$$\Pr[\mathsf{Count}(k,\xi) \geq t+1 : k \leftarrow \mathsf{KeyGen}(1^\lambda), \$_k \leftarrow \mathsf{Mint}(k), \xi \leftarrow \mathcal{A}(1^\lambda, \$_k^{\otimes t})] \leq 1 - \frac{1}{p}.$$

Proof of Theorem 5.9. Let $(\mathsf{QM.KeyGen}, \mathsf{QM.Mint}, \mathsf{QM.Ver})$ be a private-key quantum money scheme with the symmetric verifiability. From it, we construct a OWSG as follows.

- $\mathsf{KeyGen}(1^\lambda) \to k$: Run $k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)$. Output k.
- $\mathsf{StateGen}(k) \to \phi_k$: Run $\$_k \leftarrow \mathsf{QM.Mint}(k)$. Output $\phi_k := \$_k$.
- $\mathsf{Ver}(k', \phi_k) \to \top/\bot$: Parse $\phi_k = \$_k$. Run $v \leftarrow \mathsf{QM.Ver}(k', \$_k)$. Output v.

The correctness is clear. Let us show the security. Assume that it is not secure. Then, there exists a QPT adversary $\mathcal{A}$, a polynomial t, and a polynomial p such that

$$\sum_{k,k'} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(\$_k^{\otimes t})] \Pr[\top \leftarrow \mathsf{QM.Ver}(k', \$_k)] \geq \frac{1}{p}.$$

Define the set

$$S := \left\{(k,k') \;\middle|\; \Pr[\top \leftarrow \mathsf{QM.Ver}(k', \$_k)] \geq \frac{1}{2p}\right\}.$$

Then, we have

$$\sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(\$_k^{\otimes t})] > \frac{1}{2p}.$$

This is shown as follows.

$$\begin{aligned}
\frac{1}{p} &\leq \sum_{k,k'} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(\$_k^{\otimes t})] \Pr[\top \leftarrow \mathsf{QM.Ver}(k', \$_k)] \\
&= \sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(\$_k^{\otimes t})] \Pr[\top \leftarrow \mathsf{QM.Ver}(k', \$_k)] \\
&\quad + \sum_{(k,k')\notin S} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(\$_k^{\otimes t})] \Pr[\top \leftarrow \mathsf{QM.Ver}(k', \$_k)] \\
&< \sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{QM.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(\$_k^{\otimes t})] + \frac{1}{2p}.
\end{aligned}$$

From the $\mathcal{A}$, we construct a QPT adversary $\mathcal{B}$ that breaks the security of the private-key quantum money scheme as follows: On input $\$_k^{\otimes t}$, it runs $k' \leftarrow \mathcal{A}(\$_k^{\otimes t})$. It then runs $\$_{k'} \leftarrow \mathsf{QM.Mint}(k')$ ℓ times, where ℓ is a polynomial specified later, and outputs $\xi := \$_{k'}^{\otimes \ell}$. Let us show that thus defined $\mathcal{B}$ breaks the security of the

private-key quantum money scheme. Let v_j be the bit that is 1 if the output of $\mathsf{QM.Ver}(k,\xi_j)$ is $\top$, and is 0 otherwise. Then, for any $(k,k')\in S$,

$$\begin{aligned}\Pr[v_j=1] &= \Pr[\top\leftarrow \mathsf{QM.Ver}(k,\xi_j)]\\ &= \Pr[\top\leftarrow \mathsf{QM.Ver}(k,\$_{k'})]\\ &= \Pr[\top\leftarrow \mathsf{QM.Ver}(k',\$_{k})]\\ &\geq \frac{1}{2p}\end{aligned}$$

for each $j\in[1,2,...,\ell]$. Let us take $\ell\geq \max(4p(t+1),16p^3)$. Then, for any $(k,k')\in S$,

$$\begin{aligned}\Pr[\mathsf{Count}(k,\$_{k'}^{\otimes\ell})\geq t+1] &= \Pr\Big[\sum_{j=1}^{\ell} v_j\geq t+1\Big]\\ &\geq \Pr\Big[\sum_{j=1}^{\ell} v_j\geq \frac{\ell}{4p}\Big]\\ &= \Pr\Big[\sum_{j=1}^{\ell} v_j\geq \frac{\ell}{2p}-\frac{\ell}{4p}\Big]\\ &\geq \Pr\Big[\sum_{j=1}^{\ell} v_j\geq \mathbb{E}(\sum_{j=1}^{\ell} v_j)-\frac{\ell}{4p}\Big]\\ &\geq 1-2\exp\Big[-\frac{2\ell}{16p^2}\Big]\\ &\geq 1-2e^{-2p}.\end{aligned}$$

The probability that the $\mathcal{B}$ breaks the security of the quantum money scheme is therefore

$$\begin{aligned}&\sum_{k,k'}\Pr[k\leftarrow\mathsf{QM.KeyGen}(1^\lambda)]\Pr[k'\leftarrow\mathcal{A}(\$_k^{\otimes t})]\Pr[\mathsf{Count}(k,\$_{k'}^{\otimes\ell})\geq t+1]\\ \geq\ &\sum_{(k,k')\in S}\Pr[k\leftarrow\mathsf{QM.KeyGen}(1^\lambda)]\Pr[k'\leftarrow\mathcal{A}(\$_k^{\otimes t})]\Pr[\mathsf{Count}(k,\$_{k'}^{\otimes\ell})\geq t+1]\\ \geq\ &(1-2e^{-2p})\sum_{(k,k')\in S}\Pr[k\leftarrow\mathsf{QM.KeyGen}(1^\lambda)]\Pr[k'\leftarrow\mathcal{A}(\$_k^{\otimes t})]\\ \geq\ &(1-2e^{-2p})\frac{1}{2p},\end{aligned}$$

which is non-negligible. The $\mathcal{B}$ therefore breaks the security of the private-key quantum money scheme. □

6 QPOTP

In this section, we first define (IND-based) QPOTP schemes (Section 6.1). We then show that QPOTP schemes imply OWSGs (Section 6.2), and that single-copy-secure QPOTP schemes imply EFI pairs (Section 6.3).

6.1 Definition of QPOTP

Quantum pseudo one-time pad schemes are defined as follows.

Definition 6.1 ((IND-based) quantum pseudo one-time pad (QPOTP)). *An (IND-based) quantum pseudo one-time pad (QPOTP) scheme with the key length κ and the plaintext length ℓ ($\ell > \kappa$) is a set of algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that*

- $\mathsf{KeyGen}(1^\lambda) \to \mathsf{sk}$: *It is a QPT algorithm that, on input the security parameter λ, outputs a classical secret key* $\mathsf{sk} \in \{0,1\}^\kappa$.
- $\mathsf{Enc}(\mathsf{sk}, x) \to \mathsf{ct}$: *It is a QPT algorithm that, on input* sk *and a classical plaintext message $x \in \{0,1\}^\ell$, outputs an ℓn-qubit quantum ciphertext* ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to x'$: *It is a QPT algorithm that, on input* sk *and* ct, *outputs $x' \in \{0,1\}^\ell$.*

We require the following correctness and security.

Correctness: *For any $x \in \{0,1\}^\ell$,*

$$\Pr[x \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) : \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda), \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, x)] \geq 1 - \mathsf{negl}(\lambda).$$

Security: *For any $x_0, x_1 \in \{0,1\}^\ell$, any QPT adversary $\mathcal{A}$, and any polynomial t,*

$$\begin{aligned}&|\Pr[1 \leftarrow \mathcal{A}(\mathsf{ct}_0^{\otimes t}) : \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda), \mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{sk}, x_0)]\\ &- \Pr[1 \leftarrow \mathcal{A}(\mathsf{ct}_1^{\otimes t}) : \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda), \mathsf{ct}_1 \leftarrow \mathsf{Enc}(\mathsf{sk}, x_1)]| \leq \mathsf{negl}(\lambda).\end{aligned}$$

Definition 6.2. *We say that a QPOTP scheme is single-copy-secure if the security holds only for $t = 1$.*

Remark 6.3. Note that the above definition of QPOTP is different from that of [AQY22] in the following two points. First, we consider a general secret key generation QPT algorithm, while they consider uniform sampling of the secret key. Second, we consider the IND-based version of the security, while the security definition of [AQY22] is as follows: For any $x \in \{0,1\}^\ell$, any QPT adversary $\mathcal{A}$, and any polynomial t,

$$\begin{aligned}&|\Pr[1 \leftarrow \mathcal{A}(\mathsf{ct}^{\otimes t}) : \mathsf{sk} \leftarrow \{0,1\}^\kappa, \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, x)]\\ &- \Pr[1 \leftarrow \mathcal{A}((|\psi_1\rangle \otimes ... \otimes |\psi_\ell\rangle)^{\otimes t}) : |\psi_1\rangle, ..., |\psi_\ell\rangle \leftarrow \mu_n]| \leq \mathsf{negl}(\lambda),\end{aligned}$$

where $|\psi\rangle \leftarrow \mu_n$ means the Haar random sampling of n-qubit states. It is clear that the security definition of [AQY22] implies our IND-based security, and therefore if QPOTP schemes of [AQY22] exist, those of Definition 6.1 exist. Since our results are constructions of OWSGs and EFI pairs from QPOTP, the above modification only makes our results stronger.

Remark 6.4. QPOTP is constructed from PRSGs [AQY22].

6.2 OWSGs from QPOTP

Theorem 6.5. *If QPOTP schemes with $\kappa < \ell$ exist, then OWSGs exist.*

Proof of Theorem 6.5. Let $(\mathsf{OTP.KeyGen}, \mathsf{OTP.Enc}, \mathsf{OTP.Dec})$ be a QPOTP scheme with $\kappa < \ell$. From it, we construct a wOWSG as follows.[21] (From Theorem 3.7, it is enough for the existence of OWSGs.)

- $\mathsf{KeyGen}(1^\lambda) \to k$: Run $\mathsf{sk} \leftarrow \mathsf{OTP.KeyGen}(1^\lambda)$. Choose $x \leftarrow \{0,1\}^\ell$. Output $k := (\mathsf{sk}, x)$.
- $\mathsf{StateGen}(k) \to \phi_k$: Parse $k = (\mathsf{sk}, x)$. Run $\mathsf{ct}_{\mathsf{sk},x} \leftarrow \mathsf{OTP.Enc}(\mathsf{sk}, x)$. Output $\phi_k := \mathsf{ct}_{\mathsf{sk},x} \otimes |x\rangle\langle x|$.
- $\mathsf{Ver}(k', \phi_k) \to \top/\bot$: Parse $k' = (\mathsf{sk}', x')$. Parse $\phi_k = \mathsf{ct}_{\mathsf{sk},x} \otimes |x\rangle\langle x|$. Run $x'' \leftarrow \mathsf{OTP.Dec}(\mathsf{sk}', \mathsf{ct}_{\mathsf{sk},x})$. If $x'' = x' = x$, output $\top$. Otherwise, output $\bot$.

The correctness is clear. Let us show the security. Assume that it is not secure. It means that for any polynomial p there exist a QPT adversary $\mathcal{A}$ and a polynomial t such that

$$\Pr\left[x' = x'' = x : \begin{array}{l} \mathsf{sk} \leftarrow \mathsf{OTP.KeyGen}(1^\lambda), \\ x \leftarrow \{0,1\}^\ell, \\ \mathsf{ct}_{\mathsf{sk},x} \leftarrow \mathsf{OTP.Enc}(\mathsf{sk}, x), \\ (\mathsf{sk}', x') \leftarrow \mathcal{A}(\mathsf{ct}_{\mathsf{sk},x}^{\otimes t} \otimes |x\rangle\langle x|^{\otimes t}) \\ x'' \leftarrow \mathsf{OTP.Dec}(\mathsf{sk}', \mathsf{ct}_{\mathsf{sk},x}) \end{array}\right] \geq 1 - \frac{1}{p}. \tag{3}$$

From this $\mathcal{A}$, we construct a QPT adversary $\mathcal{B}$ that breaks the security of the QPOTP scheme as follows. Let $b \in \{0,1\}$ be the parameter of the following security game.

1. $\mathcal{B}$ chooses $x_0, x_1 \leftarrow \{0,1\}^\ell$, and sends them to the challenger $\mathcal{C}$.
2. $\mathcal{C}$ runs $\mathsf{sk} \leftarrow \mathsf{OTP.KeyGen}(1^\lambda)$.
3. $\mathcal{C}$ runs $\mathsf{ct}_{\mathsf{sk},x_b} \leftarrow \mathsf{OTP.Enc}(\mathsf{sk}, x_b)$ $t+1$ times.
4. $\mathcal{C}$ sends $\mathsf{ct}_{\mathsf{sk},x_b}^{\otimes t+1}$ to $\mathcal{B}$.
5. $\mathcal{B}$ runs $(\mathsf{sk}', x') \leftarrow \mathcal{A}(\mathsf{ct}_{\mathsf{sk},x_b}^{\otimes t} \otimes |x_0\rangle\langle x_0|^{\otimes t})$.
6. $\mathcal{B}$ runs $x'' \leftarrow \mathsf{OTP.Dec}(\mathsf{sk}', \mathsf{ct}_{k,x_b})$. If $x' = x'' = x_0$, $\mathcal{B}$ outputs $b' = 0$. Otherwise, it outputs $b' = 1$.

It is clear that $\Pr[b' = 0 | b = 0]$ is equivalent to the left-hand-side of Eq. (3). On the other hand,

$$\begin{aligned}
& \Pr[b' = 0 | b = 1] \\
= & \frac{1}{2^{2\ell}} \sum_{x_0, x_1, \mathsf{sk}, \mathsf{sk}'} \Pr[\mathsf{sk} \leftarrow \mathsf{OTP.KeyGen}(1^\lambda)] \Pr[\mathsf{sk}' \leftarrow \mathcal{A}(\mathsf{ct}_{\mathsf{sk},x_1}^{\otimes t} \otimes |x_0\rangle\langle x_0|^{\otimes t})] \\
& \times \Pr[x_0 \leftarrow \mathsf{OTP.Dec}(\mathsf{sk}', \mathsf{ct}_{\mathsf{sk},x_1})] \\
\leq & \frac{1}{2^{2\ell}} \sum_{x_0, x_1, \mathsf{sk}, \mathsf{sk}'} \Pr[\mathsf{sk} \leftarrow \mathsf{OTP.KeyGen}(1^\lambda)] \Pr[x_0 \leftarrow \mathsf{OTP.Dec}(\mathsf{sk}', \mathsf{ct}_{\mathsf{sk},x_1})] \\
= & \frac{1}{2^{2\ell}} \sum_{x_1, \mathsf{sk}, \mathsf{sk}'} \Pr[\mathsf{sk} \leftarrow \mathsf{OTP.KeyGen}(1^\lambda)] \sum_{x_0} \Pr[x_0 \leftarrow \mathsf{OTP.Dec}(\mathsf{sk}', \mathsf{ct}_{\mathsf{sk},x_1})] \\
= & \frac{1}{2^{2\ell}} \sum_{x_1, \mathsf{sk}, \mathsf{sk}'} \Pr[\mathsf{sk} \leftarrow \mathsf{OTP.KeyGen}(1^\lambda)] \\
= & \frac{2^\kappa}{2^\ell} \leq \frac{1}{2}.
\end{aligned}$$

[21] A similar proof idea was given in Lemma 4.6 of [GGKT05]. However, the direct application of the proof will not work, because ciphertexts (and therefore output states of OWSGs) are quantum and the verification of "preimages" is done by the additional verification algorithm.

Therefore $|\Pr[b'=0|b=0]-\Pr[b'=0|b=1]|$ is non-negligible, which means that the $\mathcal{B}$ breaks the security of the QPOTP. □

6.3 EFI Pairs from Single-Copy-Secure QPOTP

We will use the following result, which is (implicitly) shown in [LC19].[22]

Theorem 6.6 ([LC19]). *Let $\mathsf{KeyGen}(1^\lambda)\to k$ be an algorithm that, on input the security parameter λ, outputs a classical secret key $k\in\{0,1\}^\kappa$. Let $\{\mathsf{Enc}^k,\mathsf{Dec}^k\}_k$ be quantum operations. Assume that the following is satisfied: For any $\rho_{\mathbf{A},\mathbf{B}}$,*

$$\frac{1}{2}\Big\|\sum_k \Pr[k](\mathsf{Dec}^k_{\mathbf{A}}\otimes I_{\mathbf{B}})(\mathsf{Enc}^k_{\mathbf{A}}\otimes I_{\mathbf{B}})(\rho_{\mathbf{A},\mathbf{B}})-\rho_{\mathbf{A},\mathbf{B}}\Big\|_1\le \mathsf{negl}(\lambda), \tag{4}$$

where $\Pr[k]:=\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]$. Let us define

$$\begin{aligned}\rho_0&:=\sum_k \Pr[k](\mathsf{Enc}^k_{\mathbf{A}}\otimes I_{\mathbf{B}})|\Psi\rangle\langle\Psi|_{\mathbf{A},\mathbf{B}},\\ \rho_1&:=\sum_k \Pr[k](\mathsf{Enc}^k_{\mathbf{A}}\otimes I_{\mathbf{B}})\Big(|0^n\rangle\langle 0^n|_{\mathbf{A}}\otimes \frac{I^{\otimes n}}{2^n}_{\mathbf{B}}\Big),\end{aligned}$$

where $|\Psi\rangle_{\mathbf{A},\mathbf{B}}$ is the maximally entangled state over n-qubit registers $\mathbf{A}$ and $\mathbf{B}$. If $\frac{1}{2}\|\rho_0-\rho_1\|_1\le\mathsf{negl}(\lambda)$, then $\kappa\ge 2n+\log(1-\mathsf{negl}(\lambda))$.

Theorem 6.7. *If single-copy-secure QPOTP schemes with $\kappa<\ell$ exist then EFI pairs exist.*

Proof of Theorem 6.7. Let $(\mathsf{OTP.KeyGen},\mathsf{OTP.Enc},\mathsf{OTP.Dec})$ be a single-copy-secure QPOTP scheme with $\kappa<\ell$. From it, we construct an EFI pair $\mathsf{StateGen}(1^\lambda,b)\to\rho_b$ as a QPT algorithm that outputs

$$\begin{aligned}\rho_0&:=\frac{1}{2^{2n}}\sum_{\mathsf{sk},x,z}\Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(x,z)}\otimes\Big[(X^xZ^z\otimes I)|\Psi\rangle\langle\Psi|(X^xZ^z\otimes I)\Big],\\ \rho_1&:=\frac{1}{2^{2n}}\sum_{\mathsf{sk},x,z}\Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(x,z)}\otimes\Big[(X^xZ^z\otimes I)\Big(|0^n\rangle\langle 0^n|\otimes\frac{I^{\otimes n}}{2^n}\Big)(X^xZ^z\otimes I)\Big],\end{aligned}$$

where $\Pr[\mathsf{sk}]:=\Pr[\mathsf{sk}\leftarrow\mathsf{OTP.KeyGen}(1^\lambda)]$, $\mathsf{ct}_{\mathsf{sk},(x,z)}\leftarrow\mathsf{OTP.Enc}(\mathsf{sk},(x,z))$, $x,z\in\{0,1\}^n$, $X^x:=\bigotimes_{i=1}^n X_i^{x_i}$, $Z^z:=\bigotimes_{i=1}^n Z_i^{z_i}$, $2n=\ell$ and $|\Psi\rangle$ is the maximally-entangled state on two n-qubit registers. It is clear that ρ_0 and ρ_1 can be generated in QPT in a natural way.

First let us show the computational indistinguishability of ρ_0 and ρ_1. Let us define

$$\begin{aligned}\rho'_0&:=\frac{1}{2^{2n}}\sum_{\mathsf{sk},x,z}\Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(0^n,0^n)}\otimes\Big[(X^xZ^z\otimes I)|\Psi\rangle\langle\Psi|(X^xZ^z\otimes I)\Big],\\ \rho'_1&:=\frac{1}{2^{2n}}\sum_{\mathsf{sk},x,z}\Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(0^n,0^n)}\otimes\Big[(X^xZ^z\otimes I)\Big(|0^n\rangle\langle 0^n|\otimes\frac{I^{\otimes n}}{2^n}\Big)(X^xZ^z\otimes I)\Big].\end{aligned}$$

[22] See Theorem 4 of [LC19]. ρ_0 and ρ_1 correspond to ρ_{MC} and σ_{MC}, respectively. Moreover, Take $\epsilon=\gamma=\mathsf{negl}(\lambda)$.

Then, for any QPT adversary $\mathcal{A}$, from the security of the QPOTP scheme,

$$\begin{aligned}
& |\Pr[1 \leftarrow \mathcal{A}(\rho_0)] - \Pr[1 \leftarrow \mathcal{A}(\rho'_0)]| \\
\leq\ & \frac{1}{2^{2n}} \sum_{x,z} \Big| \Pr\Big[1 \leftarrow \mathcal{A}\Big(\sum_{\mathsf{sk}} \Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(x,z)} \otimes \Big[(X^x Z^z \otimes I)|\Psi\rangle\langle\Psi|(X^x Z^z \otimes I)\Big]\Big)\Big] \\
& - \Pr\Big[1 \leftarrow \mathcal{A}\Big(\sum_{\mathsf{sk}} \Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(0^n,0^n)} \otimes \Big[(X^x Z^z \otimes I)|\Psi\rangle\langle\Psi|(X^x Z^z \otimes I)\Big]\Big)\Big]\Big| \\
\leq\ & \mathsf{negl}(\lambda),
\end{aligned}$$

and

$$|\Pr[1 \leftarrow \mathcal{A}(\rho_1)] - \Pr[1 \leftarrow \mathcal{A}(\rho'_1)]| \leq \mathsf{negl}(\lambda)$$

in a similar way. Moreover, $\rho'_0 = \rho'_1$, because

$$\begin{aligned}
\rho'_0 &= \sum_{\mathsf{sk}} \Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(0^n,0^n)} \otimes \frac{1}{2^{2n}} \sum_{x,z} \Big[(X^x Z^z \otimes I)|\Psi\rangle\langle\Psi|(X^x Z^z \otimes I)\Big], \\
&= \sum_{\mathsf{sk}} \Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(0^n,0^n)} \otimes \frac{I^{\otimes n}}{2^n} \otimes \frac{I^{\otimes n}}{2^n}, \\
\rho'_1 &= \sum_{\mathsf{sk}} \Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(0^n,0^n)} \otimes \frac{1}{2^{2n}} \sum_{x,z} \Big[(X^x Z^z \otimes I)\Big(|0^n\rangle\langle 0^n| \otimes \frac{I^{\otimes n}}{2^n}\Big)(X^x Z^z \otimes I)\Big] \\
&= \sum_{\mathsf{sk}} \Pr[\mathsf{sk}]\mathsf{ct}_{\mathsf{sk},(0^n,0^n)} \otimes \frac{I^{\otimes n}}{2^n} \otimes \frac{I^{\otimes n}}{2^n}.
\end{aligned}$$

Hence we have

$$|\Pr[1 \leftarrow \mathcal{A}(\rho_0)] - \Pr[1 \leftarrow \mathcal{A}(\rho_1)]| \leq \mathsf{negl}(\lambda)$$

for any QPT adversary $\mathcal{A}$.

Next let us show the statistical distinguishability of ρ_0 and ρ_1. From the QPOTP, construct KeyGen and Enc^k, Dec^k of Theorem 6.6 as follows.

- $\mathsf{KeyGen}(1^\lambda) \to k$: Run $\mathsf{sk} \leftarrow \mathsf{OTP.KeyGen}(1^\lambda)$. Output $k := \mathsf{sk}$.
- $\mathsf{Enc}^k(\rho_{\mathbf{A},\mathbf{B}}) = \rho'_{\mathbf{A}',\mathbf{B}}$: Parse $k = \mathsf{sk}$. Choose $x, z \leftarrow \{0,1\}^n$. Run $\mathsf{ct}_{\mathsf{sk},(x,z)} \leftarrow \mathsf{OTP.Enc}(\mathsf{sk},(x,z))$. Output
$$\rho'_{\mathbf{A}',\mathbf{B}} := \frac{1}{2^{2n}} \sum_{x,z} \mathsf{ct}_{\mathsf{sk},(x,z)} \otimes \Big([(X^x Z^z)_{\mathbf{A}} \otimes I_{\mathbf{B}}]\rho_{\mathbf{A},\mathbf{B}}[(X^x Z^z)_{\mathbf{A}} \otimes I_{\mathbf{B}}]\Big).$$
- $\mathsf{Dec}^k(\rho'_{\mathbf{A}',\mathbf{B}}) = \rho_{\mathbf{A},\mathbf{B}}$: Parse $k = \mathsf{sk}$. Parse
$$\rho'_{\mathbf{A}',\mathbf{B}} = \frac{1}{2^{2n}} \sum_{x,z} \mathsf{ct}_{\mathsf{sk},(x,z)} \otimes \Big([(X^x Z^z)_{\mathbf{A}} \otimes I_{\mathbf{B}}]\rho_{\mathbf{A},\mathbf{B}}[(X^x Z^z)_{\mathbf{A}} \otimes I_{\mathbf{B}}]\Big).$$
Run $(x,z) \leftarrow \mathsf{OTP.Dec}(\mathsf{sk}, \mathsf{ct}_{\mathsf{sk},(x,z)})$. Obtain $\rho_{\mathbf{A},\mathbf{B}}$ by correcting the Pauli one-time pad. Output $\rho_{\mathbf{A},\mathbf{B}}$.

It satisfies Eq. (4). Assume that $\frac{1}{2}\|\rho_0 - \rho_1\|_1 \leq \mathsf{negl}(\lambda)$. Then, from Theorem 6.6, $\kappa \geq 2n + \log(1 - \mathsf{negl}(\lambda))$, which means $\kappa \geq \ell + \log(1 - \mathsf{negl}(\lambda))$, but it contradicts the assumption of $\kappa < \ell$. Therefore, the statistical distinguishability of ρ_0 and ρ_1 is shown. □

7 SV-SI-OWSGs

In this section, we define SV-SI-OWSGs (Section 7.1), and show that SV-SI-OWSGs are equivalent to EFI pairs (Section 7.2). In Section 7.1, before defining SV-SI-OWSGs, we first define SV-OWSGs for a didactic purpose. We will point out that SV-OWSGs seem to need a more constraint so that they become equivalent to EFI. We then define SV-SI-OWSGs.

7.1 Definition of SV-SI-OWSGs

We first define secretly-verifiable OWSGs (SV-OWSGs) as follows.

Definition 7.1 (Secretly-verifiable OWSGs (SV-OWSGs)). *A secretly-verifiable OWSG (SV-OWSG) is a set of algorithms* $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$ *as follows.*

- $\mathsf{KeyGen}(1^\lambda) \to k$: *It is a QPT algorithm that, on input the security parameter λ, outputs a key $k \in \{0,1\}^\kappa$.*
- $\mathsf{StateGen}(k) \to \phi_k$: *It is a QPT algorithm that, on input k, outputs an m-qubit state ϕ_k.*
- $\mathsf{Ver}(k', k) \to \top/\bot$: *It is a QPT algorithm that, on input k and k', outputs $\top/\bot$.*

We require the following two properties.

Correctness:

$$\Pr[\top \leftarrow \mathsf{Ver}(k, k) : k \leftarrow \mathsf{KeyGen}(1^\lambda)] \geq 1 - \mathsf{negl}(\lambda).$$

Security: *For any QPT adversary $\mathcal{A}$ and any polynomial t,*

$$\Pr[\top \leftarrow \mathsf{Ver}(k', k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k), k' \leftarrow \mathcal{A}(\phi_k^{\otimes t})] \leq \mathsf{negl}(\lambda).$$

The following lemma shows that, without loss of generality, Ver can be replaced with the algorithm of just checking whether $k = k'$ or not.

Lemma 7.2. *Let* $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$ *be a SV-OWSG. Then, the following SV-OWSG* $(\mathsf{KeyGen}', \mathsf{StateGen}', \mathsf{Ver}')$ *exists.*

- KeyGen' *and* $\mathsf{StateGen}'$ *are the same as* KeyGen *and* $\mathsf{StateGen}$*, respectively.*
- $\mathsf{Ver}'(k', k) \to \top/\bot$: *On input k and k', output $\top$ if $k = k'$. Otherwise, output $\bot$.*

Proof. The correctness of $(\mathsf{KeyGen}', \mathsf{StateGen}', \mathsf{Ver}')$ is clear. Let us show the security. Assume that it is not secure. Then there exist a QPT adversary $\mathcal{A}$, a polynomial t, and a polynomial p such that

$$\sum_k \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes t})] \geq \frac{1}{p}.$$

Define the set

$$S := \left\{ k \;\middle|\; \Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes t})] \geq \frac{1}{2p} \right\}.$$

Then, we have

$$\sum_{k\in S}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]>\frac{1}{2p}.$$

This is because

$$\begin{aligned}\frac{1}{p}&\leq\sum_{k}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]\Pr[k\leftarrow\mathcal{A}(\phi_k^{\otimes t})]\\&=\sum_{k\in S}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]\Pr[k\leftarrow\mathcal{A}(\phi_k^{\otimes t})]\\&\quad+\sum_{k\notin S}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]\Pr[k\leftarrow\mathcal{A}(\phi_k^{\otimes t})]\\&<\sum_{k\in S}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]+\frac{1}{2p}.\end{aligned}$$

Also define the set

$$T:=\Big\{k\ \Big|\ \Pr[\top\leftarrow\mathsf{Ver}(k,k)]\geq 1-\frac{1}{p}\Big\}.$$

Then, we have

$$\sum_{k\in T}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]>1-\mathsf{negl}(\lambda).$$

This is because

$$\begin{aligned}1-\mathsf{negl}(\lambda)&\leq\sum_{k}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]\Pr[\top\leftarrow\mathsf{Ver}(k,k)]\\&=\sum_{k\in T}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]\Pr[\top\leftarrow\mathsf{Ver}(k,k)]\\&\quad+\sum_{k\notin T}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]\Pr[\top\leftarrow\mathsf{Ver}(k,k)]\\&<\sum_{k\in T}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]\\&\quad+\Big(1-\frac{1}{p}\Big)\Big(1-\sum_{k\in T}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]\Big).\end{aligned}$$

Here, the first inequality is from the correctness of $(\mathsf{KeyGen},\mathsf{StateGen},\mathsf{Ver})$. From the union bound, we have

$$\sum_{k\in S\cap T}\Pr[k\leftarrow\mathsf{KeyGen}(1^\lambda)]>\frac{1}{2p}-\mathsf{negl}(\lambda).$$

From the $\mathcal{A}$, we construct a QPT adversary that breaks the security of $(\mathsf{KeyGen},\mathsf{StateGen},\mathsf{Ver})$ as follows:

On input $\phi_k^{\otimes t}$, run $k' \leftarrow \mathcal{A}(\phi_k^{\otimes t})$. Output k'. Then, the probability that $\mathcal{B}$ breaks the security is

$$\begin{aligned}
&\sum_{k,k'} \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(\phi_k^{\otimes t})] \Pr[\top \leftarrow \mathsf{Ver}(k', k)] \\
\geq &\sum_{k} \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes t})] \Pr[\top \leftarrow \mathsf{Ver}(k, k)] \\
\geq &\sum_{k \in S \cap T} \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes t})] \Pr[\top \leftarrow \mathsf{Ver}(k, k)] \\
\geq &\frac{1}{2p}\Big(1 - \frac{1}{p}\Big)\Big(\frac{1}{2p} - \mathsf{negl}(\lambda)\Big),
\end{aligned}$$

which is non-negligible. Therefore $\mathcal{B}$ breaks the security of $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$. □

Note that statistically-secure SV-OWSGs are easy to realize. For example, consider the following construction:

- $\mathsf{KeyGen}(1^\lambda)$: Sample $k \leftarrow \{0,1\}^\lambda$.
- $\mathsf{StateGen}(k)$: Output $\frac{I^{\otimes m}}{2^m}$.
- $\mathsf{Ver}(k', k)$: Output $\top$ if $k' = k$. Otherwise, output $\bot$.

We therefore need a constraint to have a meaningful primitive. We define secretly-verifiable and statistically-invertible OWSGs (SV-SI-OWSGs) as follows. Introducing the statistical invertibility allows us to avoid trivial constructions with the statistical security.

Definition 7.3 (Secretly-verifiable and statistically-invertible OWSGs (SV-SI-OWSGs)). *A secretly-verifiable and statistically-invertible OWSG (SV-SI-OWSG) is a set of algorithms* $(\mathsf{KeyGen}, \mathsf{StateGen})$ *as follows.*

- $\mathsf{KeyGen}(1^\lambda) \to k$: *It is a QPT algorithm that, on input the security parameter λ, outputs a key $k \in \{0,1\}^\kappa$.*
- $\mathsf{StateGen}(k) \to \phi_k$: *It is a QPT algorithm that, on input k, outputs an m-qubit state ϕ_k.*

We require the following two properties.

Statistical invertibility: *There exists a polynomial p such that, for any k and k' ($k \neq k'$), $\frac{1}{2}\|\phi_k - \phi_{k'}\|_1 \geq \frac{1}{p}$.*

Computational non-invertibility: *For any QPT adversary $\mathcal{A}$ and any polynomial t,*

$$\Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes t}) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k)] \leq \mathsf{negl}(\lambda).$$

The following lemma shows that the statistical invertibility with advantage $\frac{1}{\mathrm{poly}(\lambda)}$ can be amplified to $1 - 2^{-q}$ for any polynomial q.

Lemma 7.4. *If a SV-SI-OWSG exists then a SV-SI-OWSG with statistical invertibility larger than $1 - 2^{-q}$ with any polynomial q exists.*

Proof. Let $(\mathsf{KeyGen}, \mathsf{StateGen})$ be a SV-SI-OWSG with statistical invertibility larger than $\frac{1}{p}$, where p is a polynomial. From it, we construct a new SV-SI-OWSG $(\mathsf{KeyGen}', \mathsf{StateGen}')$ as follows:

- $\mathsf{KeyGen}'(1^\lambda) \to k$: Run $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, and output k.
- $\mathsf{StateGen}'(k) \to \phi'_k$: Run $\phi_k \leftarrow \mathsf{StateGen}(k)$ $2pq$ times, and output $\phi'_k := \phi_k^{\otimes 2pq}$.

First, for any k and k' ($k \neq k'$),

$$\begin{aligned}\frac{1}{2}\|\phi'_k - \phi'_{k'}\|_1 &= \frac{1}{2}\|\phi_k^{\otimes 2pq} - \phi_{k'}^{\otimes 2pq}\|_1 \\ &\geq 1 - \exp(-2qp\|\phi_k - \phi_{k'}\|_1/4) \\ &\geq 1 - \exp(-q) \\ &\geq 1 - 2^{-q},\end{aligned}$$

which shows the statistical invertibility of $(\mathsf{KeyGen}', \mathsf{StateGen}')$ with the advantage larger than $1 - 2^{-q}$. Second, from the computational non-invertibility of $(\mathsf{KeyGen}, \mathsf{StateGen})$,

$$\begin{aligned}&\Pr[k \leftarrow \mathcal{A}(\phi_k'^{\otimes t}) : k \leftarrow \mathsf{KeyGen}'(1^\lambda), \phi'_k \leftarrow \mathsf{StateGen}'(k)] \\ =\ &\Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes 2pqt}) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k)] \\ \leq\ &\mathsf{negl}(\lambda)\end{aligned}$$

for any QPT adversary $\mathcal{A}$ and any polynomial t, which shows the computational non-invertibility of $(\mathsf{KeyGen}', \mathsf{StateGen}')$. □

The following lemma shows that the statistical invertibility is equivalent to the existence of a (unbounded) adversary that can find the correct k given many copies of ϕ_k except for a negligible error.

Lemma 7.5. *The statistical invertibility is satisfied if and only if the following is satisfied: There exists a (not necessarily QPT) POVM measurement $\{\Pi_k\}_{k\in\{0,1\}^\kappa}$ and a polynomial t such that $Tr(\Pi_k\phi_k^{\otimes t}) \geq 1 - \mathsf{negl}(\lambda)$ and $Tr(\Pi_{k'}\phi_k^{\otimes t}) \leq \mathsf{negl}(\lambda)$ for all k and k' ($k \neq k'$).*

Proof. First, we show the if part. Assume that there exists a POVM measurement $\{\Pi_k\}_{k\in\{0,1\}^\kappa}$ and a polynomial t such that $\mathrm{Tr}(\Pi_k\phi_k^{\otimes t}) \geq 1 - \mathsf{negl}(\lambda)$ and $\mathrm{Tr}(\Pi_{k'}\phi_k^{\otimes t}) \leq \mathsf{negl}(\lambda)$ for all k and k' ($k \neq k'$). Then,

$$\begin{aligned}\frac{t}{2}\|\phi_k - \phi_{k'}\|_1 &\geq \frac{1}{2}\|\phi_k^{\otimes t} - \phi_{k'}^{\otimes t}\|_1 \\ &\geq \mathrm{Tr}(\Pi_k\phi_k^{\otimes t}) - \mathrm{Tr}(\Pi_k\phi_{k'}^{\otimes t}) \\ &\geq 1 - \mathsf{negl}(\lambda) - \mathsf{negl}(\lambda) \\ &= 1 - \mathsf{negl}(\lambda),\end{aligned}$$

which means

$$\frac{1}{2}\|\phi_k - \phi_{k'}\|_1 \geq \frac{1}{t} - \mathsf{negl}(\lambda) \geq \frac{1}{2t}.$$

Next, we show the only if part. Assume that the statistical invertibility is satisfied. Then, there exists a polynomial p such that $\frac{1}{2}\|\phi_k - \phi_{k'}\|_1 \geq \frac{1}{p}$ for all k and k' ($k \neq k'$). Let $t := 12p\kappa$. Then,

$$\frac{1}{2}\|\phi_k^{\otimes t} - \phi_{k'}^{\otimes t}\|_1 \geq 1 - e^{-t\frac{\|\phi_k - \phi_{k'}\|_1}{4}} \geq 1 - e^{-6\kappa} \geq 1 - 2^{-6\kappa},$$

which means $F(\phi_k^{\otimes t}, \phi_{k'}^{\otimes t}) \leq 2^{-6\kappa+1}$. From Theorem 7.6 below,

$$\max_k(1 - \mathrm{Tr}(\mu_k \phi_k^{\otimes t})) \leq \sum_{k \neq k'} \sqrt{F(\phi_k^{\otimes t}, \phi_{k'}^{\otimes t})} \leq 2^{-3\kappa+1}(2^{2\kappa} - 2^{\kappa}) \leq 2^{-\kappa+1},$$

which means $\mathrm{Tr}(\mu_k \phi_k^{\otimes t}) \geq 1 - 2^{-\kappa+1}$ and $\mathrm{Tr}(\mu_{k'} \phi_k^{\otimes t}) \leq 2^{-\kappa+1}$ for any k and k' ($k' \neq k$). □

Theorem 7.6 ([Mon19]). *Let $\{\rho_i\}_i$ be a set of states. Define the POVM measurement $\{\mu_i\}_i$ with $\mu_i := \Sigma^{-1/2}\rho_i\Sigma^{-1/2}$, where $\Sigma := \sum_i \rho_i$, and the inverse is taken on the support of Σ. Then, $\max_i(1 - Tr(\mu_i\rho_i)) \leq \sum_{i \neq j} \sqrt{F(\rho_i, \rho_j)}$.*

7.2 Equivalence of SV-SI-OWSGs and EFI Pairs

Theorem 7.7. *SV-SI-OWSGs exist if and only if EFI pairs exist.*

This Theorem is shown by combining the following two theorems.

Theorem 7.8. *If EFI pairs exist then SV-SI-OWSGs exist.*

Theorem 7.9. *If SV-SI-OWSGs exist then EFI pairs exist.*

Proof of Theorem 7.8. We show that if EFI pairs exist then SV-SI-OWSGs exist. Let $\mathsf{EFI.StateGen}(1^\lambda, b) \to \rho_b$ be an EFI pair. As is explained in Remark 2.2, we can assume without loss of generality that $\frac{1}{2}\|\rho_0 - \rho_1\|_1 \geq 1 - \mathsf{negl}(\lambda)$, which means $F(\rho_0, \rho_1) \leq \mathsf{negl}(\lambda)$. From the EFI pair, we construct a SV-SI-OWSG as follows.

- $\mathsf{KeyGen}(1^\lambda) \to k$: Choose $k \leftarrow \{0,1\}^\kappa$, and output k.
- $\mathsf{StateGen}(k) \to \phi_k$: Run $\mathsf{EFI.StateGen}(1^\lambda, k_i) \to \rho_{k_i}$ for each $i \in [\kappa]$. Output $\phi_k := \bigotimes_{i=1}^\kappa \rho_{k_i}$.

The statistical invertibility is easily shown as follows. If $k \neq k'$, there exists a $j \in [\kappa]$ such that $k_j \neq k'_j$. Then,

$$F(\phi_k, \phi_{k'}) = \prod_{i=1}^{\kappa} F(\rho_{k_i}, \rho_{k'_i}) \leq F(\rho_{k_j}, \rho_{k'_j}) \leq \mathsf{negl}(\lambda),$$

which means $\frac{1}{2}\|\phi_k - \phi_{k'}\|_1 \geq 1 - \mathsf{negl}(\lambda)$. This shows the statistical invertibility.

Let us next show the computational non-invertibility. From the standard hybrid argument, and the computational indistinguishability of ρ_0 and ρ_1, we have

$$\left|\frac{1}{2^\kappa}\sum_{k\in\{0,1\}^\kappa} \Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes t})] - \frac{1}{2^\kappa}\sum_{k\in\{0,1\}^\kappa} \Pr[k \leftarrow \mathcal{A}(\phi_{0^\kappa}^{\otimes t})]\right| \leq \mathsf{negl}(\lambda) \tag{5}$$

for any QPT adversary $\mathcal{A}$ and any polynomial t. (It will be shown later.) Hence

$$\begin{aligned}
& \Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes t}) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \phi_k \leftarrow \mathsf{StateGen}(k)] \\
= \; & \frac{1}{2^\kappa}\sum_{k\in\{0,1\}^\kappa} \Pr[k \leftarrow \mathcal{A}(\phi_k^{\otimes t})] \\
\leq \; & \frac{1}{2^\kappa}\sum_{k\in\{0,1\}^\kappa} \Pr[k \leftarrow \mathcal{A}(\phi_{0^\kappa}^{\otimes t})] + \mathsf{negl}(\lambda) \\
= \; & \frac{1}{2^\kappa} + \mathsf{negl}(\lambda),
\end{aligned}$$

which shows the computational non-invertibility.

Let us show Eq. (5). For each $z \in \{0,1\}^{\kappa t}$, define $\Phi_z := \bigotimes_{i=1}^{\kappa t} \rho_{z_i}$. Let $z, z' \in \{0,1\}^{\kappa t}$ be two bit strings such that, for a single $j \in [\kappa t]$, $z_j = 0$, $z'_j = 1$, and $z_i = z'_i$ for all $i \neq j$. (In other words, z and z' are the same except for the jth bit.) Then, we can show that

$$\Big|\frac{1}{2^\kappa}\sum_{k\in\{0,1\}^\kappa} \Pr[k \leftarrow \mathcal{A}(\Phi_z)] - \frac{1}{2^\kappa}\sum_{k\in\{0,1\}^\kappa} \Pr[k \leftarrow \mathcal{A}(\Phi_{z'})]\Big| \leq \mathsf{negl}(\lambda) \tag{6}$$

for any QPT adversary $\mathcal{A}$. In fact, assume that

$$\Big|\frac{1}{2^\kappa}\sum_{k} \Pr[k \leftarrow \mathcal{A}(\Phi_z)] - \frac{1}{2^\kappa}\sum_{k} \Pr[k \leftarrow \mathcal{A}(\Phi_{z'})]\Big| \geq \frac{1}{\mathrm{poly}(\lambda)}$$

for a QPT adversary $\mathcal{A}$. Then, from this $\mathcal{A}$, we can construct a QPT adversary $\mathcal{B}$ that breaks the security of the EFI pair as follows: On input ρ_b, choose $k \leftarrow \{0,1\}^\kappa$, and run $k' \leftarrow \mathcal{A}((\bigotimes_{i=1}^{j-1} \rho_{z_i}) \otimes \rho_b \otimes (\bigotimes_{i=j+1}^{\kappa t} \rho_{z_i}))$. If $k' = k$, output $b' = 1$. If $k' \neq k$, output $b' = 0$. Because

$$\begin{aligned}
\Pr[b' = 1 | b = 0] &= \frac{1}{2^\kappa}\sum_k \Pr[k \leftarrow \mathcal{A}(\Phi_z)],\\
\Pr[b' = 1 | b = 1] &= \frac{1}{2^\kappa}\sum_k \Pr[k \leftarrow \mathcal{A}(\Phi_{z'})],
\end{aligned}$$

we have

$$|\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]| \geq \frac{1}{\mathrm{poly}(\lambda)},$$

which means that the $\mathcal{B}$ breaks the security of the EFI pair. From the standard hybrid argument and Eq. (6), we have Eq. (5). □

Proof of Theorem 7.9. We show that if SV-SI-OWSGs exist then EFI pairs exist. Let $(\mathsf{OWSG.KeyGen}, \mathsf{OWSG.StateGen})$ be a SV-SI-OWSG. Without loss of generality, we can assume that $\mathsf{OWSG.KeyGen}$ is the following algorithm: first apply a QPT unitary U on $|0...0\rangle$ to generate

$$U|0...0\rangle = \sum_k \sqrt{\Pr[k \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)]}|k\rangle|\mu_k\rangle,$$

and trace out the second register, where $\{|\mu_k\rangle\}_k$ are some normalized states. Moreover, without loss of generality, we can also assume that $\mathsf{OWSG.StateGen}$ is the following algorithm: first apply a QPT unitary V_k that depends on k on $|0...0\rangle$ to generate $V_k|0...0\rangle = |\psi_k\rangle_{\mathbf{A},\mathbf{B}}$, and trace out the register $\mathbf{A}$.

From the SV-SI-OWSG, we want to construct an EFI pair. For that goal, we construct a statistically-hiding and computationally-binding canonical quantum bit commitment scheme from SV-SI-OWSG. Due to Theorem 2.8 (the equivalence between different flavors of commitments), we then have a statistically-binding and computationally-hiding canonical quantum bit commitment scheme, which is equivalent to an EFI pair. From the SV-SI-OWSG, we construct a statistically-hiding and computationally-binding canonical quantum bit commitment scheme $\{Q_0, Q_1\}$ as follows.

$$\begin{aligned}
Q_0|0\rangle_{\mathbf{C},\mathbf{R}} &:= \sum_k \sqrt{\Pr[k]}(|k\rangle|\mu_k\rangle)_{\mathbf{C}_1}|\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2}|0\rangle_{\mathbf{R}_3},\\
Q_1|0\rangle_{\mathbf{C},\mathbf{R}} &:= \sum_k \sqrt{\Pr[k]}(|k\rangle|\mu_k\rangle)_{\mathbf{C}_1}|\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2}|k\rangle_{\mathbf{R}_3},
\end{aligned}$$

where $\Pr[k] := \Pr[k \leftarrow \mathsf{OWSG.KeyGen}(1^\lambda)]$, $\mathbf{C}_2$ is the combination of all "$\mathbf{A}$ registers" of $|\psi_k\rangle$, $\mathbf{R}_2$ is the combination of all "$\mathbf{B}$ registers" of $|\psi_k\rangle$, $\mathbf{C} := (\mathbf{C}_1, \mathbf{C}_2)$ and $\mathbf{R} := (\mathbf{R}_2, \mathbf{R}_3)$. Moreover, t is a polynomial specified later. It is clear that such $\{Q_0, Q_1\}$ is implemented in QPT in a natural way.

Let us first show the computational binding of $\{Q_0, Q_1\}$. Assume that it is not computationally binding. Then, there exists a QPT unitary U, an ancilla state $|\tau\rangle$, and a polynomial p such that

$$\|(\langle 0|Q_1^\dagger)_{\mathbf{C},\mathbf{R}} U_{\mathbf{R},\mathbf{Z}} (Q_0|0\rangle_{\mathbf{C},\mathbf{R}} \otimes |\tau\rangle_{\mathbf{Z}})\| \geq \frac{1}{p}.$$

Then,

$$\begin{aligned}
\frac{1}{p^2} &\leq \|(\langle 0|Q_1^\dagger)_{\mathbf{C},\mathbf{R}} U_{\mathbf{R},\mathbf{Z}} (Q_0|0\rangle_{\mathbf{C},\mathbf{R}} \otimes |\tau\rangle_{\mathbf{Z}})\|^2 \\
&= \Big\| \Big(\sum_{k'} \sqrt{\Pr[k']} \langle k', \mu_{k'}|_{\mathbf{C}_1} \langle \psi_{k'}|^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} \langle k'|_{\mathbf{R}_3} \Big) \\
&\quad \times \Big(\sum_{k} \sqrt{\Pr[k]} |k, \mu_k\rangle_{\mathbf{C}_1} U_{\mathbf{R},\mathbf{Z}} |\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} |0\rangle_{\mathbf{R}_3} |\tau\rangle_{\mathbf{Z}} \Big) \Big\|^2 \\
&= \Big\| \sum_{k} \Pr[k] \langle \psi_k|^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} \langle k|_{\mathbf{R}_3} U_{\mathbf{R},\mathbf{Z}} |\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} |0\rangle_{\mathbf{R}_3} |\tau\rangle_{\mathbf{Z}} \Big\|^2 \\
&\leq \Big(\sum_{k} \Pr[k] \Big\| \langle \psi_k|^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} \langle k|_{\mathbf{R}_3} U_{\mathbf{R},\mathbf{Z}} |\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} |0\rangle_{\mathbf{R}_3} |\tau\rangle_{\mathbf{Z}} \Big\| \Big)^2 \\
&\leq \sum_{k} \Pr[k] \Big\| \langle \psi_k|^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} \langle k|_{\mathbf{R}_3} U_{\mathbf{R},\mathbf{Z}} |\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} |0\rangle_{\mathbf{R}_3} |\tau\rangle_{\mathbf{Z}} \Big\|^2 \\
&\leq \sum_{k} \Pr[k] \Big\| \langle k|_{\mathbf{R}_3} U_{\mathbf{R},\mathbf{Z}} |\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} |0\rangle_{\mathbf{R}_3} |\tau\rangle_{\mathbf{Z}} \Big\|^2. \qquad (7)
\end{aligned}$$

In the third inequality, we have used Jensen's inequality.[23] From this U, we construct a QPT adversary $\mathcal{B}$ that breaks the computational non-invertibility of the SV-SI-OWSG as follows: On input the $\mathbf{R}_2$ register of $|\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2}$, apply $U_{\mathbf{R},\mathbf{Z}}$ on $|\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2} |0\rangle_{\mathbf{R}_3} |\tau\rangle_{\mathbf{Z}}$, and measure the $\mathbf{R}_3$ register in the computational basis. Output the result. Then, the probability that $\mathcal{B}$ correctly outputs k is equal to Eq. (7). Therefore, $\mathcal{B}$ breaks the computational non-invertibility of the SV-SI-OWSG.

Let us next show the statistical hiding of $\{Q_0, Q_1\}$. In the following, we construct a (not necessarily QPT) unitary $W_{\mathbf{R},\mathbf{Z}}$ such that

$$\|W_{\mathbf{R},\mathbf{Z}} Q_0 |0\rangle_{\mathbf{C},\mathbf{R}} |0\rangle_{\mathbf{Z}} - Q_1 |0\rangle_{\mathbf{C},\mathbf{R}} |0\rangle_{\mathbf{Z}}\|_1 \leq \mathsf{negl}(\lambda). \qquad (8)$$

Then, we have

$$\begin{aligned}
& \|\mathrm{Tr}_{\mathbf{R}}(Q_0|0\rangle_{\mathbf{C},\mathbf{R}}) - \mathrm{Tr}_{\mathbf{R}}(Q_1|0\rangle_{\mathbf{C},\mathbf{R}})\|_1 \\
=\ & \|\mathrm{Tr}_{\mathbf{R},\mathbf{Z}}(Q_0|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}}) - \mathrm{Tr}_{\mathbf{R},\mathbf{Z}}(Q_1|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}})\|_1 \\
=\ & \|\mathrm{Tr}_{\mathbf{R},\mathbf{Z}}(W_{\mathbf{R},\mathbf{Z}} Q_0|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}}) - \mathrm{Tr}_{\mathbf{R},\mathbf{Z}}(Q_1|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}})\|_1 \\
\leq\ & \|W_{\mathbf{R},\mathbf{Z}} Q_0|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}} - Q_1|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}}\|_1 \\
\leq\ & \mathsf{negl}(\lambda),
\end{aligned}$$

[23] For a real convex function f, $f(\sum_i p_i x_i) \leq \sum_i p_i f(x_i)$.

which shows the statistical hiding of $\{Q_0, Q_1\}$.

Now we explain how to construct $W_{\mathbf{R},\mathbf{Z}}$. From Lemma 7.5, there exists a (not necessarily QPT) POVM measurement $\{\Pi_k\}_k$ and a polynomial t such that $\mathrm{Tr}(\Pi_k \phi_k^{\otimes t}) \geq 1 - \mathsf{negl}(\lambda)$ and $\mathrm{Tr}(\Pi_{k'} \phi_k^{\otimes t}) \leq \mathsf{negl}(\lambda)$ for all k and k' ($k \neq k'$). Let $U_{\mathbf{R}_2,\mathbf{Z}}$ be a unitary operator that implements the POVM measurement $\{\Pi_k\}_k$ in the following way

$$U_{\mathbf{R}_2,\mathbf{Z}}|\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2}|0...0\rangle_{\mathbf{Z}} = \sqrt{1-\epsilon_k}|k\rangle|junk_k\rangle + \sum_{k':k'\neq k}\sqrt{\epsilon_{k'}}|k'\rangle|junk_{k'}\rangle,$$

where $\mathbf{Z}$ is the ancilla register, $\{\epsilon_i\}_i$ are real numbers such that $1-\epsilon_k \geq 1-\mathsf{negl}(\lambda)$ and $\epsilon_{k'} \leq \mathsf{negl}(\lambda)$ for all $k' \neq k$, and $\{|junk_i\rangle\}_i$ are "junk" states that are normalized. Measuring the first register of the state realizes the POVM. Let $V_{\mathbf{R},\mathbf{Z}}$ be the following unitary:[24]

1. Apply $U_{\mathbf{R}_2,\mathbf{Z}}$ on $|\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2}|0...0\rangle_{\mathbf{Z}}|0\rangle_{\mathbf{R}_3}$:

$$U_{\mathbf{R}_2,\mathbf{Z}}|\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2}|0...0\rangle_{\mathbf{Z}}|0\rangle_{\mathbf{R}_3} = \Big[\sqrt{1-\epsilon_k}|k\rangle|junk_k\rangle + \sum_{k':k'\neq k}\sqrt{\epsilon_{k'}}|k'\rangle|junk_{k'}\rangle\Big]|0\rangle_{\mathbf{R}_3}.$$

2. Copy the content of the first register to the register $\mathbf{R}_3$:

$$\sqrt{1-\epsilon_k}|k\rangle|junk_k\rangle|k\rangle_{\mathbf{R}_3} + \sum_{k':k'\neq k}\sqrt{\epsilon_{k'}}|k'\rangle|junk_{k'}\rangle|k'\rangle_{\mathbf{R}_3}.$$

Define $W_{\mathbf{R},\mathbf{Z}} := U^\dagger_{\mathbf{R}_2,\mathbf{Z}} V_{\mathbf{R},\mathbf{Z}}$.

Let us show that thus constructed $W_{\mathbf{R},\mathbf{Z}}$ satisfies Eq. (8).

$$\begin{aligned}
&\Big((\langle 0|Q_1^\dagger)_{\mathbf{C},\mathbf{R}}\langle 0|_{\mathbf{Z}}\Big)\Big(W_{\mathbf{R},\mathbf{Z}}Q_0|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}}\Big)\\
=\ &\Big((\langle 0|Q_1^\dagger)_{\mathbf{C},\mathbf{R}}\langle 0|_{\mathbf{Z}}\Big)\Big(U^\dagger_{\mathbf{R}_2,\mathbf{Z}}V_{\mathbf{R},\mathbf{Z}}Q_0|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}}\Big)\\
=\ &\Big((\langle 0|Q_1^\dagger)_{\mathbf{C},\mathbf{R}}\langle 0|_{\mathbf{Z}}U^\dagger_{\mathbf{R}_2,\mathbf{Z}}\Big)\Big(V_{\mathbf{R},\mathbf{Z}}Q_0|0\rangle_{\mathbf{C},\mathbf{R}}|0\rangle_{\mathbf{Z}}\Big)\\
=\ &\Big(\sum_k \sqrt{\Pr[k]}(\langle k|\langle\mu_k|)_{\mathbf{C}_1}\Big[\sqrt{1-\epsilon_k}\langle k|\langle junk_k|\langle k|_{\mathbf{R}_3} + \sum_{k'\neq k}\sqrt{\epsilon_{k'}}\langle k'|\langle junk_{k'}|\langle k|_{\mathbf{R}_3}\Big]\Big)\\
&\times\Big(\sum_k \sqrt{\Pr[k]}(|k\rangle|\mu_k\rangle)_{\mathbf{C}_1}\Big[\sqrt{1-\epsilon_k}|k\rangle|junk_k\rangle|k\rangle_{\mathbf{R}_3} + \sum_{k'\neq k}\sqrt{\epsilon_{k'}}|k'\rangle|junk_{k'}\rangle|k'\rangle_{\mathbf{R}_3}\Big]\Big)\\
=\ &\sum_k \Pr[k](1-\epsilon_k)\\
\geq\ &1-\mathsf{negl}(\lambda).
\end{aligned}$$

□

Acknowledgements. TM is supported by JST CREST JPMJCR23I3, JST Moonshot R&D JPMJMS2061-5-1-1, JST FOREST, MEXT QLEAP, the Grant-in-Aid for 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.

[24] For simplicity, we define $V_{\mathbf{R},\mathbf{Z}}$ by explaining how it acts on $|\psi_k\rangle^{\otimes t}_{\mathbf{C}_2,\mathbf{R}_2}|0...0\rangle_{\mathbf{Z}}|0\rangle_{\mathbf{R}_3}$, but it is clear from the explanation how $V_{\mathbf{R},\mathbf{Z}}$ is defined.

References

[Aar19] Scott Aaronson. Shadow tomography of quantum states. *SIAM J. Comput.*, 49(5):STOC18–368, 2019. (Cited on page 10, 19.)

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 41–60. ACM Press, May 2012. (Cited on page 19.)

[AGM21] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Can you sign a quantum state? *Quantum*, 5:603, dec 2021. (Cited on page 46.)

[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 40.)

[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 4, 25.)

[BCQ22] Zvika Brakerski, Ran Canetti, and Luowen Qian. On the computational hardness needed for quantum cryptography. Cryptology ePrint Archive, Paper 2022/1181, 2022. (Cited on page 4, 8.)

[BS19] Zvika Brakerski and Omri Shmueli. (Pseudo) random quantum states with binary phase. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part I*, volume 11891 of *LNCS*, pages 229–250. Springer, Heidelberg, December 2019. (Cited on page 40.)

[BS20] Zvika Brakerski and Omri Shmueli. Scalable pseudorandom quantum states. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 417–440. Springer, Heidelberg, August 2020. (Cited on page 40.)

[CGG+23] Bruno Cavalar, Eli Goldin, Matthew Gray, Peter Hall, Yanyi Liu, and Angelos Pelecanos. On the computational hardness of quantum one-wayness. *arXiv:2312.08363*, 2023. (Cited on page 7, 42.)

[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, 11, 12, 13, 46, 47.)

[CX22] Shujiao Cao and Rui Xue. On constructing one-way quantum state generators, and more. Cryptology ePrint Archive, Paper 2022/1323, 2022. (Cited on page 6, 11.)

[DMS00] Paul Dumais, Dominic Mayers, and Louis Salvail. Perfectly concealing quantum bit commitment from any quantum one-way permutation. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 300–315. Springer, Heidelberg, May 2000. (Cited on page 9.)

[FUYZ20] Junbin Fang, Dominique Unruh, Jun Yan, and Dehua Zhou. How to base security on the perfect/statistical binding property of quantum bit commitment? Cryptology ePrint Archive, Report 2020/621, 2020. https://ia.cr/2020/621. (Cited on page 9.)

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. SIAM J. COMPUT. Vol. 35, No. 1, pp. 217–246, 2005. (Cited on page 26.)

[Gol90] Oded Goldreich. A note on computational indistinguishability. Information Processing Letters 34.6 (1990), pp.277–281., 1990. (Cited on page 4.)

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001. (Cited on page 6, 11.)

[HMY23] Minki Hhan, Tomoyuki Morimae, and Takashi Yamakawa. From the hardness of detecting superpositions to cryptography: Quantum public key encryption and commitments. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 639–667. Springer, Heidelberg, April 2023. (Cited on page 9, 46.)

[HS11] Thomas Holenstein and Grant Schoenebeck. General hardness amplification of predicates and puzzles (extended abstract). In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 19–36. Springer, Heidelberg, March 2011. (Cited on page 11.)

[IJK09] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Chernoff-type direct product theorems. *Journal of Cryptology*, 22(1):75–92, January 2009. (Cited on page 11.)

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 134–147. IEEE Computer Society, 1995. (Cited on page 4.)

[IR90] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 8–26. Springer, Heidelberg, August 1990. (Cited on page 46.)

[ISV+17] Gene Itkis, Emily Shen, Mayank Varia, David Wilson, and Arkady Yerukhimovich. Bounded-collusion attribute-based encryption from minimal assumptions. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 67–87. Springer, Heidelberg, March 2017. (Cited on page 15, 16.)

[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 4, 19, 40.)

[KKNY12] Akinori Kawachi, Takeshi Koshiba, Harumichi Nishimura, and Tomoyuki Yamakami. Computational indistinguishability between quantum states and its cryptographic application. *Journal of Cryptology*, 25(3):528–555, July 2012. (Cited on page 43, 44, 46.)

[KQST23] William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in algorithmica. STOC, 2023. (Cited on page 4.)

[Kre21] W. Kretschmer. Quantum pseudorandomness and classical complexity. *TQC 2021*, 2021. (Cited on page 4, 7, 40.)

[KT23] Dakshita Khurana and Kabir Tomer. Commitments from quantum one-wayness. Cryptology ePrint Archive, Paper 2023/1620, 2023. https://eprint.iacr.org/2023/1620. (Cited on page 7.)

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, 1979., 1979. (Cited on page 17.)

[LC19] Ching-Yi Lai and Kai-Min Chung. Quantum encryption and generalized quantum shannon impossibility. *Designs, Codes and Cryptography volume 87, pages 1961–1972 (2019)*, 2019. (Cited on page 27.)

[LMW23] Alex Lombardi, Fermi Ma, and John Wright. A one-query lower bound for unitary synthesis and breaking quantum cryptography. Cryptology ePrint Archive, Paper 2023/1602, 2023. https://eprint.iacr.org/2023/1602. (Cited on page 7.)

[Mon19] Ashley Montanaro. Pretty simple bounds on quantum state discrimination. *arXiv:1908.08312*, 2019. (Cited on page 33.)

[MY22] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2022. (Cited on page 4, 5, 6, 9, 10, 14, 17.)

[RS19] Roy Radian and Or Sattath. Semi-quantum money. *arXiv/1908.08889*, 2019. (Cited on page 11.)

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 497–527. Springer, Heidelberg, May 2016. (Cited on page 9.)

[Yan20] Jun Yan. General properties of quantum bit commitments. Cryptology ePrint Archive, Paper 2020/1488, 2020. (Cited on page 13.)

[Yan21] Jun Yan. Quantum computationally predicate-binding commitments with application in quantum zero-knowledge arguments for NP. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 575–605. Springer, Heidelberg, December 2021. (Cited on page 9.)

[Yan22] Jun Yan. General properties of quantum bit commitments (extended abstract). In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 628–657. Springer, Heidelberg, December 2022. (Cited on page 4, 8, 9.)

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982. (Cited on page 5, 6, 11.)

[YWLQ15] Jun Yan, Jian Weng, Dongdai Lin, and Yujuan Quan. Quantum bit commmitment with application in quantum zero-knowledge proof (extended abstract). In Khaled M. Elbassioni and Kazuhisa Makino, editors, *ISAAC 2015*, volume 9472 of *Lecture Notes in Computer Science*, pages 555–565. Springer, 2015. (Cited on page 4, 9.)

A PRSGs

PRSGs are defined as follows.

Definition A.1 (Pseudorandom quantum states generators (PRSGs) [JLS18]). *A pseudorandom quantum states generator (PRSG) is a set of algorithms* $(\mathsf{KeyGen}, \mathsf{StateGen})$ *as follows.*

- $\mathsf{KeyGen}(1^\lambda) \to k$: *It is a QPT algorithm that, on input the security parameter λ, outputs a classical key k.*
- $\mathsf{StateGen}(k) \to |\phi_k\rangle$: *It is a QPT algorithm that, on input k, outputs an m-qubit quantum state $|\phi_k\rangle$.*

We require the following security: For any polynomial t and any QPT adversary $\mathcal{A}$,

$$|\Pr_{k\leftarrow\mathsf{KeyGen}(1^\lambda)}[1\leftarrow\mathcal{A}(|\phi_k\rangle^{\otimes t})] - \Pr_{|\psi\rangle\leftarrow\mu_m}[1\leftarrow\mathcal{A}(|\psi\rangle^{\otimes t})]| \le \mathsf{negl}(\lambda),$$

where μ_m is the Haar measure on m-qubit states.

Remark A.2. In the original definition of PRSGs [JLS18], the classical key k is uniformly sampled at random. We use a more general definition where k is sampled by a QPT algorithm, which was introduced in [BS19].

Remark A.3. Note that in the above definition, the outputs of $\mathsf{StateGen}$ are assumed to be pure. It could be possible to consider mixed states, but anyway the states have to be negligibly close to pure states (except for a negligible fraction of k).[25] We therefore, for simplicity, assume that the outputs of $\mathsf{StateGen}$ are pure. In that case, $\mathsf{StateGen}$ runs as follows: apply a QPT unitary U on $|k\rangle|0...0\rangle$ to generate $U(|k\rangle|0...0\rangle) = |\phi_k\rangle\otimes|\eta_k\rangle$, and output $|\phi_k\rangle$. Note that the existance of the "junk" register $|\eta_k\rangle$ is essential, because otherwise it is not secure.[26] (The simplest example of the junk register $|\eta_k\rangle$ would be $|\eta_k\rangle = |k\rangle$, i.e., $U(|k\rangle|0...0\rangle) = |k\rangle\otimes|\phi_k\rangle$. In that case, it is often convenient to consider that $\mathsf{StateGen}$ applies a QPT unitary U_k that depends on k on the state $|0...0\rangle$, and outputs $|\phi_k\rangle := U_k|0...0\rangle$.)

Remark A.4. PRSGs can be constructed from (quantum-secure) one-way functions [JLS18, BS19, BS20].

Remark A.5. [BS20] showed that PRSGs with $m = c\log\lambda$ for some $0 < c < 1$ can be constructed with the statistical security. On the other hand, [AGQY22], showed that PRSGs with $m \ge \log\lambda$ require computational assumptions. [AGQY22] also pointed out that the result of [Kre21] can be refined to show that the existence of PRSGs with $m = (1+\epsilon)\log\lambda$ for all $\epsilon > 0$ implies $\mathbf{BQP} \neq \mathbf{PP}$.

[25] Consider the case $t = 2$, and consider a QPT adversary $\mathcal{A}$ that, given $\phi_k^{\otimes 2}$ or $|\psi\rangle^{\otimes 2}$, runs the SWAP test on them. Then,

$$\begin{aligned}\left|\frac{1}{2^\lambda}\sum_k \Pr[1\leftarrow\mathcal{A}(\phi_k^{\otimes 2})] - \mathbb{E}_{|\psi\rangle\leftarrow\mu_m}\Pr[1\leftarrow\mathcal{A}(|\psi\rangle^{\otimes 2})]\right| &= \left|\frac{1}{2^\lambda}\sum_k \frac{1+\mathrm{Tr}(\phi_k^2)}{2} - \frac{1+1}{2}\right| \\ &= \frac{1}{2}\left|\frac{1}{2^\lambda}\sum_k \mathrm{Tr}(\phi_k^2) - 1\right|,\end{aligned}$$

which has to be negligible. Therefore, $\mathrm{Tr}(\phi_k^2)$ is negligibly close to 1 (except for a negligible fraction of k).

[26] Consider a QPT adversary who applies $U^\dagger$ on each copy of the received states, and measures them in the computational basis. If the same k is obtained many times, the adversary concludes that the states are copies of pseudorandom states.

B Verification Algorithm for Special Case

Lemma B.1. *Let* $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$ *be a OWSG that satisfies the following.*

- *All* ϕ_k *are pure.*
- $\Pr[\top \leftarrow \mathsf{Ver}(k, \phi_k)] \geq 1 - \mathsf{negl}(\lambda)$ *for all* k*.*

Then, the following OWSG $(\mathsf{KeyGen}', \mathsf{StateGen}', \mathsf{Ver}')$ *exists.*

- $\mathsf{KeyGen}', \mathsf{StateGen}'$ *are the same as* KeyGen *and* $\mathsf{StateGen}$*, respectively.*
- $\mathsf{Ver}'(k', \phi_k)$ *is the following algorithm. Measure* ϕ_k *with the basis* $\{|\phi_{k'}\rangle\langle\phi_{k'}|, I - |\phi_{k'}\rangle\langle\phi_{k'}|\}$*, and output* $\top$ *if the first result is obtained. Otherwise, output* $\bot$*.*

Proof. The correctness of $(\mathsf{KeyGen}', \mathsf{StateGen}', \mathsf{Ver}')$ is clear. Let us show the security. Assume that it is not secure. Then, there exists a QPT adversary $\mathcal{A}$, a polynomial t, and a polynomial p, such that

$$\sum_{k,k'} \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\phi_k\rangle^{\otimes t})] |\langle\phi_k|\phi_{k'}\rangle|^2 \geq \frac{1}{p}.$$

If we define

$$S := \Big\{(k, k') \;\Big|\; |\langle\phi_k|\phi_{k'}\rangle|^2 \geq \frac{1}{2p}\Big\},$$

we have

$$\sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\phi_k\rangle^{\otimes t})] \geq \frac{1}{2p}.$$

From the $\mathcal{A}$, we construct a QPT adversary $\mathcal{B}$ that breaks the security of the original OWSG as follows: On input $|\phi_k\rangle^{\otimes t}$, run $k' \leftarrow \mathcal{A}(|\phi_k\rangle^{\otimes t})$, and output k'. Then the probability that $\mathcal{B}$ breaks the original OWSG is

$$\begin{aligned}
&\sum_{k,k'} \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\phi_k\rangle^{\otimes t})] \Pr[\top \leftarrow \mathsf{Ver}(k', |\phi_k\rangle)] \\
\geq &\sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\phi_k\rangle^{\otimes t})] \Pr[\top \leftarrow \mathsf{Ver}(k', |\phi_k\rangle)] \\
\geq &\sum_{(k,k')\in S} \Pr[k \leftarrow \mathsf{KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\phi_k\rangle^{\otimes t})] \Big(\Pr[\top \leftarrow \mathsf{Ver}(k', |\phi_{k'}\rangle)] - \sqrt{1 - \frac{1}{2p}}\Big) \\
\geq &\frac{1}{2p}\Big(1 - \mathsf{negl}(\lambda) - \Big(1 - \frac{1}{4p}\Big)\Big),
\end{aligned}$$

which is non-negligible. Therefore the $\mathcal{B}$ breaks the security of the original OWSG. Here, in the second inequality, we have used the fact that for any $(k, k') \in S$, $|\langle\phi_k|\phi_{k'}\rangle|^2 \geq \frac{1}{2p}$, which means that $\mathrm{Tr}(\Pi|\phi_k\rangle\langle\phi_k|) - \mathrm{Tr}(\Pi|\phi_{k'}\rangle\langle\phi_{k'}|) \leq \sqrt{1 - \frac{1}{2p}}$ for any POVM element Π. In the last inequality, we have used Bernoulli's inequality, $(1 + x)^r \leq 1 + rx$ for any $0 \leq r \leq 1$ and $x \geq -1$. □

C OWSGs from PRSGs with Improved Parameters

The following result was pointed out by Luowen Qian. Also, a concurrent paper [CGG+23] shows similar results.

Theorem C.1. *If PRSGs with $m \geq \log \kappa$ exist, then OWSGs exist.*

Proof of Theorem C.1. Let $\mathsf{PRSG.KeyGen}(1^\lambda)$ and $\mathsf{PRSG.StateGen}(k) \to |\xi_k\rangle$ be a PRSG, where $|\xi_k\rangle$ is an m-qubit state. From it, we construct a OWSG as follows.

- $\mathsf{KeyGen}(1^\lambda) \to k$: Run $k \leftarrow \mathsf{PRSG.KeyGen}(1^\lambda)$.
- $\mathsf{StateGen}(k) \to |\phi_k\rangle$: Run $|\xi_k\rangle \leftarrow \mathsf{PRSG.StateGen}(k)$ r times, where r is a polynomial specified later. Output $|\phi_k\rangle := |\xi_k\rangle^{\otimes r}$.
- $\mathsf{Ver}(k', |\phi_k\rangle) \to \top/\bot$: Parse $|\phi_k\rangle = |\xi_k\rangle^{\otimes r}$. Measure $|\phi_k\rangle$ with the basis $\{|\phi_{k'}\rangle\langle\phi_{k'}|, I - |\phi_{k'}\rangle\langle\phi_{k'}|\}$. If the first result is obtained, output $\top$. Otherwise, output $\bot$.

Its correctness is clear. Let us show the security. Assume that it is not secure. Then, there exists a QPT adversary $\mathcal{A}$, a polynomial t, and a polynomial p, such that

$$\sum_{k,k'} \Pr[k \leftarrow \mathsf{PRSG.KeyGen}(1^\lambda)] \Pr[k' \leftarrow \mathcal{A}(|\phi_k\rangle^{\otimes t})] |\langle\phi_k|\phi_{k'}\rangle|^2 \geq \frac{1}{p}. \tag{9}$$

From this $\mathcal{A}$, we construct a QPT adversary $\mathcal{B}$ that breaks the security of the PRSG as follows. Let $b \in \{0,1\}$ be the parameter of the following security game.

1. The challenger $\mathcal{C}$ of the security game of the PRSG sends $\rho^{\otimes rt+r}$ to $\mathcal{B}$. Here, ρ is chosen as follows. If $b=0$, $\mathcal{C}$ runs $k \leftarrow \mathsf{PRSG.KeyGen}(1^\lambda)$, runs $|\xi_k\rangle \leftarrow \mathsf{PRSG.StateGen}(k)$, and set $\rho = |\xi_k\rangle$. If $b=1$, ρ is an m-qubit Haar random state $|\psi\rangle$.
2. $\mathcal{B}$ runs $k' \leftarrow \mathcal{A}(\rho^{\otimes rt})$. $\mathcal{B}$ measures $\rho^{\otimes r}$ with the basis $\{|\xi_{k'}\rangle\langle\xi_{k'}|^{\otimes r}, I - |\xi_{k'}\rangle\langle\xi_{k'}|^{\otimes r}\}$. If the first result is obtained, $\mathcal{B}$ outputs $b'=0$. Otherwise, it outputs $b'=1$.

Then, $\Pr[b'=0 \mid b=0]$ is equivalent to the left-hand-side of Eq. (9), and therefore non-negligible. On the other hand,

$$\begin{aligned}
\Pr[b'=0 \mid b=1] &= \mathbb{E}_{|\psi\rangle\leftarrow\mu_m} \sum_{k'} \Pr[k' \leftarrow \mathcal{A}(|\psi\rangle^{\otimes rt})] |\langle\xi_{k'}|\psi\rangle|^{2r} \\
&\leq \mathbb{E}_{|\psi\rangle\leftarrow\mu_m} \sum_{k'} |\langle\xi_{k'}|\psi\rangle|^{2r} \\
&= \sum_{k'} \langle\xi_{k'}|^{\otimes r} \left(\mathbb{E}_{|\psi\rangle\leftarrow\mu_m} |\psi\rangle\langle\psi|^{\otimes r}\right) |\xi_{k'}\rangle^{\otimes r} \\
&= \sum_{k'} \langle\xi_{k'}|^{\otimes r} \frac{\Pi_{sym}^{2^m,r}}{\mathrm{Tr}(\Pi_{sym}^{2^m,r})} |\xi_{k'}\rangle^{\otimes r} \\
&\leq \sum_{k'} \frac{1}{\mathrm{Tr}(\Pi_{sym}^{2^m,r})} \\
&\leq 2^\kappa 2^{-rm} r! \\
&\leq \mathsf{negl}(\lambda),
\end{aligned}$$

where $\mathbb{E}_{|\psi\rangle\leftarrow\mu_m}$ is the expectation value over m-qubit Haar random, $\Pi_{sym}^{2^m,r}$ is the projector onto the symmetric subspace of $(\mathbb{C}^{2^m})^{\otimes r}$. In the third inequality, we have used the fact that $\mathrm{Tr}(\Pi_{sym}^{N,t}) \geq \frac{N^t}{t!}$. In the last inequality, we have taken $m \geq \log \kappa$ and $r = \kappa$, and used Stirling's approximation. The $\mathcal{B}$ therefore breaks the security of the PRSG. □

D Impossibility of Statistically-Secure QDSs

In this appendix, we show the following:

Theorem D.1. *Statistically-secure QDSs do not exist.*

Proof. Let us consider the following unbounded adversary $\mathcal{A}$:

1. Given $\mathsf{pk}^{\otimes t}$ with a certain polynomial t, run the shadow tomography algorithm to find σ such that $\Pr[\top \leftarrow \mathsf{Ver}(\mathsf{pk}, m, \sigma)] \geq 1 - \frac{1}{p(\lambda)}$, where m is any fixed bit string, and p is any fixed polynomial. If there is no such σ, choose a bit string σ uniformly at random.
2. Output m and σ.

We show that the probability that such $\mathcal{A}$ win the security game is non-negligible. First, define the set

$$G := \left\{ \mathsf{sk} : \sum_{\sigma} \Pr[\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)] \Pr[\top \leftarrow \mathsf{Ver}(\mathsf{pk}, m, \sigma)] \geq 1 - \frac{1}{p} \right\}. \tag{10}$$

Then, from the correctness,

$$\begin{aligned}
1 - \mathsf{negl}(\lambda) &\leq \sum_{\mathsf{sk}} \Pr[\mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda)] \sum_{\sigma} \Pr[\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)] \Pr[\top \leftarrow \mathsf{Ver}(\mathsf{pk}, m, \sigma)] && (11)\\
&= \sum_{\mathsf{sk}\in G} \Pr[\mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda)] \sum_{\sigma} \Pr[\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)] \Pr[\top \leftarrow \mathsf{Ver}(\mathsf{pk}, m, \sigma)] && (12)\\
&+ \sum_{\mathsf{sk}\notin G} \Pr[\mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda)] \sum_{\sigma} \Pr[\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)] \Pr[\top \leftarrow \mathsf{Ver}(\mathsf{pk}, m, \sigma)] && (13)\\
&\leq \sum_{\mathsf{sk}\in G} \Pr[\mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda)] + 1 - \frac{1}{p}, && (14)
\end{aligned}$$

which means

$$\sum_{\mathsf{sk}\in G} \Pr[\mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda)] \geq \frac{1}{p} - \mathsf{negl}(\lambda). \tag{15}$$

For each $\mathsf{sk} \in G$, there exists σ_{sk} such that $\Pr[\top \leftarrow \mathsf{Ver}(\mathsf{pk}, m, \sigma_{\mathsf{sk}})] \geq 1 - \frac{1}{p}$, because otherwise it contradicts Equation (10). Therefore, for each $\mathsf{sk} \in G$, $\mathcal{A}$ can find σ such that $\Pr[\top \leftarrow \mathsf{Ver}(\mathsf{pk}, m, \sigma)] \geq 1 - \frac{1}{p}$, and because of Equation (15), the probability that $\mathcal{A}$ wins is non-negligible.

□

E Quantum SKE and Quantum PKE

In this section, we define quantum SKE (with classical keys and quantum ciphertexts) and its IND-CPA security. Then we show that IND-CPA secure quantum SKE implies QPOTP. By Theorems 6.5 and 6.7, this means that it implies both OWSGs and EFI pairs. We also observe that IND-CPA secure quantum SKE as defined here is equivalent to quantum PKE defined in [KKNY12].

E.1 Quantum SKE

We define quantum SKE as SKE with classical keys and quantum ciphertexts.

Definition E.1 (Quantum SKE). *A quantum SKE scheme is a tuple of algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that*

- $\mathsf{KeyGen}(1^\lambda) \to \mathsf{sk}$: *It is a QPT algorithm that, on input the security parameter* λ*, outputs a classical secret key* sk.
- $\mathsf{Enc}(\mathsf{sk}, x) \to \mathsf{ct}$: *It is a QPT algorithm that, on input a key* sk *and a message* $x \in \{0,1\}^\ell$*, outputs a quantum ciphertext* ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to x'$: *It is a QPT algorithm that, on input* sk *and* ct*, outputs* $x' \in \{0,1\}^\ell$.

Correctness: *For any* $x \in \{0,1\}^\ell$,

$$\Pr[x \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) : \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda), \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, x)] \geq 1 - \mathsf{negl}(\lambda).$$

IND-CPA Security: *For any QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\begin{aligned}&|\Pr[1 \leftarrow \mathcal{A}_2^{\mathsf{Enc}(\mathsf{sk},\cdot)}(\mathsf{st}, \mathsf{ct}^*) : \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda), (x_0, x_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Enc}(\mathsf{sk},\cdot)}(1^\lambda), \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{sk}, x_0)]\\ &- \Pr[1 \leftarrow \mathcal{A}_2^{\mathsf{Enc}(\mathsf{sk},\cdot)}(\mathsf{st}, \mathsf{ct}^*) : \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda), (x_0, x_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Enc}(\mathsf{sk},\cdot)}(1^\lambda), \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{sk}, x_1)]|\\ &\leq \mathsf{negl}(\lambda)\end{aligned}$$

where $\mathsf{Enc}(\mathsf{sk}, \cdot)$ *is a classically-accessible oracle that takes* x *as input and returns* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, x)$.

Remark E.2. In the above definition, we only give a single-copy of ct^* to $\mathcal{A}_2$. However, by a standard hybrid argument, we can show that the above security implies the security against $\mathcal{A}_2$ that obtains t copies of ct^* for any polynomial t.

Theorem E.3. *If there exists IND-CPA secure QSKE, there exists QPOTP.*

Proof (sketch). By a standard hybrid argument, IND-CPA security implies the multi-instance security, where the adversary is given unbounded-polynomially many ciphertexts. Thus, by taking the message length to be much larger than the secret key length and encrypting the massage by using the IND-CPA secure QSKE in a bit-by-bit manner, we can obtain QPOTP. □

Combined with Theorems 6.5 and 6.7 we obtain the following theorem.

Theorem E.4. *If there exists IND-CPA secure QSKE, there exist OWSGs and EFI pairs.*

E.2 Quantum PKE

We define quantum PKE (with quantum public keys and quantum ciphertexts) following [KKNY12].

Definition E.5 (Quantum PKE [KKNY12]). *A quantum PKE scheme is a tuple of algorithms* $(\mathsf{SKGen}, \mathsf{PKGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that*

- $\mathsf{SKGen}(1^\lambda) \to \mathsf{sk}$: *It is a QPT algorithm that, on input the security parameter* λ*, outputs a classical secret key* sk.

- $\mathsf{PKGen}(\mathsf{sk}) \to \mathsf{pk}$: *It is a QPT algorithm that, on input a secret key* sk*, outputs a quantum public key* pk.
- $\mathsf{Enc}(\mathsf{pk}, x) \to \mathsf{ct}$: *It is a QPT algorithm that, on input a public key* pk *and a message* $x \in \{0,1\}^\ell$*, outputs a quantum ciphertext* ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to x'$: *It is a QPT algorithm that, on input* sk *and* ct*, outputs* $x' \in \{0,1\}^\ell$.

Correctness: *For any* $x \in \{0,1\}^\ell$*,*

$$\Pr[x \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) : \mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda), \mathsf{pk} \leftarrow \mathsf{PKGen}(\mathsf{sk}), \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x)] \geq 1 - \mathsf{negl}(\lambda).$$

IND-CPA Security: *For any QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, and any polynomial* t*,*

$$\left| \Pr\left[1 \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{ct}^*) : \begin{array}{l} \mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda), \\ \mathsf{pk} \leftarrow \mathsf{PKGen}(\mathsf{sk}), \\ (x_0, x_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pk}^{\otimes t}), \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, x_0) \end{array} \right] - \Pr\left[1 \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{ct}^*) : \begin{array}{l} \mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda), \\ \mathsf{pk} \leftarrow \mathsf{PKGen}(\mathsf{sk}), \\ (x_0, x_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pk}^{\otimes t}), \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, x_1) \end{array} \right] \right| \leq \mathsf{negl}(\lambda).$$

IND-CPA QPKE implies IND-CPA QSKE in a trivial manner similarly to the classical setting. On the other hand, we prove that the other direction also holds, which means that the existence of IND-CPA QSKE and QPKE are equivalent.

Theorem E.6. *There exists IND-CPA secure QPKE if and only if there exists IND-CPA secure QSKE.*

Proof (sketch.) IND-CPA QPKE trivially implies IND-CPA QSKE by considering the combination of SKGen and PKGen of QPKE as KeyGen of QSKE. We prove the other direction. Let $(\mathsf{QSKE.KeyGen}, \mathsf{QSKE.Enc}, \mathsf{QSKE.Dec})$ be an IND-CPA secure QSKE with one-bit messages. We construct a QPKE scheme $(\mathsf{QPKE.PKGen}, \mathsf{QPKE.SKGen}, \mathsf{QPKE.Enc}$ with one-bit messages as follows. (Note that the message space of IND-CPA secure QPKE schemes can be extended to arbitrarily many bits by bit-wise encryption.)

- $\mathsf{QPKE.SKGen}(1^\lambda)$: Run $\mathsf{sk} \leftarrow \mathsf{QSKE.KeyGen}(1^\lambda)$ and outputs sk.
- $\mathsf{QPKE.PKGen}(\mathsf{sk})$: Run $\mathsf{ct}_0 \leftarrow \mathsf{QSKE.Enc}(\mathsf{sk}, 0)$ and $\mathsf{ct}_1 \leftarrow \mathsf{QSKE.Enc}(\mathsf{sk}, 1)$ and outputs $\mathsf{pk} := (\mathsf{ct}_0, \mathsf{ct}_1)$.
- $\mathsf{QPKE.Enc}(\mathsf{pk}, x)$: On input $\mathsf{pk} = (\mathsf{ct}_0, \mathsf{ct}_1)$ and $x \in \{0,1\}$, output $\mathsf{ct} = \mathsf{ct}_x$.
- $\mathsf{QPKE.Dec}(\mathsf{sk}, \mathsf{ct})$: Output $\mathsf{QSKE.Dec}(\mathsf{sk}, \mathsf{ct})$.

The correctness of QPKE immediately follows from that of QSKE. Noting that we can simulate arbitrarily many copies of $\mathsf{pk} = (\mathsf{ct}_0, \mathsf{ct}_1)$ by querying 0 and 1 to the encryption oracle $\mathsf{SKE.Enc}(\mathsf{sk}, \cdot)$ many times, there is a straightforward reduction from the IND-CPA security of QPKE to that of QSKE. □

Remark E.7. One may think that this theorem is surprisingly strong because it is unlikely that SKE implies PKE in the classical setting [IR90]. However, we would like to point out that QPKE as defined in [KKNY12] is not as useful as classical PKE because public keys are quantum. Their security model assumes that public keys are delivered to senders without being forged. This itself is the same as classical PKE, but the crucial difference is that forgery of public keys can be prevented by additionally using digital signatures in the classical setting, but that is not possible for QPKE because we cannot sign on quantum messages [AGM21]. In fact, the assumption that the adversary cannot forge public keys mean that it also cannot eavesdrop it because eavesdropping may change the state of the public key. Thus, their security model essentially assumes a secure channel to send public keys to the sender. If there is such a secure channel, we could simply send a key for SKE through that channel. Thus, this theorem should not be understood as a new useful feasibility result on PKE. We remark that the construction does not work if we consider QPKE with classical public keys [HMY23]. We conjecture that it is impossible to construct QPKE with classical public keys from QSKE under a certain class of black-box constructions.

F Proof of Theorem 3.14

In this section, we prove Theorem 3.14. We remark that the following proof is almost identical to that of [CHS05, Lemma 1] except for the modification explained in the proof sketch in Section 3.2.

Proof of Theorem 3.14. We prepare several definitions. We write $\mathsf{puz}^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k)$ to mean that we run $\mathsf{puz} \leftarrow \mathsf{PuzzleGen}(k)$ t times to generate $\mathsf{puz}^{\otimes t}$. For $\vec{k} = (k_1, \dots, k_i)$ and k, we define $\vec{k} \circ k := (k_1, \dots, k_i, k)$. We set $t' := \lceil \frac{6q \ln(6q)}{\delta^n} \rceil \cdot t$. The construction of $\mathcal{A}'$ is given in Figure 2. Note that t' copies of puz are indeed sufficient for $\mathcal{A}'$ since it feeds t copies of puz to $\mathcal{A}$ at most $L = \lceil \frac{6q \ln(6q)}{\delta^{n-v+1}} \rceil \le \lceil \frac{6q \ln(6q)}{\delta^n} \rceil$ times. We analyze $\mathcal{A}'$ below.

For a sequence $\vec{k} = (k_1, \dots, k_i)$ of $i \le n-1$ check keys, we define

$$\mathsf{rsp}(\vec{k}) := \Pr\left[\begin{array}{l|l} \forall j \in \{i+1, \dots, n\} & (k_{i+1}, \dots, k_n) \leftarrow \mathsf{CheckGen}^{n-i}(1^\lambda) \\ \mathsf{Ver}(\mathsf{ans}_j, k_j) = \top & \mathsf{puz}_j^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k_j) \text{ for } j \in [n] \\ & (\mathsf{ans}_1, \dots, \mathsf{ans}_n) \leftarrow \mathcal{A}(\mathsf{puz}_1^{\otimes t}, \dots, \mathsf{puz}_n^{\otimes t}) \end{array} \right].$$

For a sequence $\vec{k} = (k_1, \dots, k_{i-1})$ of $i-1 \le n-2$ check keys, we define a subset $\mathsf{Ext}_i(\vec{k})$ of check keys as follows:

$$\mathsf{Ext}_i(\vec{k}) := \left\{ k : \mathsf{rsp}(\vec{k} \circ k) \ge \delta^{n-i} \left(1 + \frac{1}{6q}\right) \right\}.$$

Let $\mathsf{prefix} = (k_1, \dots, k_{v-1})$ be the prefix at the end of preprocessing phase. Let prefix_i be the random variable defined to be $(k_1, \dots, k_i)$ if $i \le v-1$ and otherwise $\bot$. For convenience, we assign to prefix_0 a special symbol Λ that is different from $\bot$. For $i \in [n-1]$, let Wrong_i be the event that $\mathsf{prefix}_{i-1} \neq \bot$ and one of the following holds:

1. $\mathsf{prefix}_i \neq \bot$ and $\mathsf{rsp}(\mathsf{prefix}_i) < \delta^{n-i}\left(1 - \frac{1}{6q}\right)$.
2. $\mathsf{prefix}_i = \bot$ and $\Pr[k \in \mathsf{Ext}_i(\mathsf{prefix}_{i-1}) : k \leftarrow \mathsf{CheckGen}(1^\lambda)] \ge \frac{\delta^{n-i+1}}{6q}$.

By the Chernoff bound, we can show that

$$\Pr[\mathsf{Wrong}_i] \le \frac{\delta}{6qn} \tag{16}$$

for all $i \in [n-1]$. We omit the proof since it is exactly the same as the proof of [CHS05, Claim 2].

We move onto the analysis of the online phase. We say that $\mathcal{A}'$ succeeds if its answer passes the verification.[27] We prove that $\mathcal{A}'$ succeeds with probability at least $\delta\left(1-\frac{5}{6q}\right)$ unless Wrong_i occurs for some $i \in [n-1]$. If this is proven, it finishes the proof of Theorem 3.14 since the probability that Wrong_i occurs for some $i \in [n-1]$ is at most $\frac{\delta}{6q}$ by the union bound and Equation (16).

Suppose that the preprocessing phase generates prefix $\mathsf{prefix}_{v-1} = (k_1, \dots, k_{v-1})$ of length $v-1$.

When $v = n$, we have $\mathsf{prefix}_i \neq \bot$ for all $i \in [n-1]$. Thus, if we assume that Wrong_{n-1} does not occur, we have

$$\mathsf{rsp}(\mathsf{prefix}_{n-1}) \geq \delta\left(1-\frac{1}{6q}\right).$$

This directly means that the probability that $\mathcal{A}$' succeeds is at least $\delta\left(1-\frac{1}{6q}\right) \geq \delta\left(1-\frac{5}{6q}\right)$.

In the following, we consider the case where $v \leq n-1$. In this case, we have $\mathsf{prefix}_i \neq \bot$ for all $i \in [v-1]$ and $\mathsf{prefix}_v = \bot$. Then if we assume that neither of Wrong_{v-1} or Wrong_v occurs, we have

$$\mathsf{rsp}(\mathsf{prefix}_{v-1}) \geq \delta^{n-v+1}\left(1-\frac{1}{6q}\right) \tag{17}$$

and

$$\Pr[k \in E : k \leftarrow \mathsf{CheckGen}(1^\lambda)] < \frac{\delta^{n-v+1}}{6q} \tag{18}$$

where we define $E \coloneqq \mathsf{Ext}_v(\mathsf{prefix}_{v-1})$ for convenience. In the following, we fix $\mathsf{prefix}_{v-1} = (k_1, \dots, k_{v-1})$ that satisfies Equations (17) and (18) and show that $\mathcal{A}$' succeeds with probability at least $\delta\left(1-\frac{5}{6q}\right)$ for any such fixed prefix. As explained above, this suffices for completing the proof of Theorem 3.14.

For any check key k, we define

$$w(k) \coloneqq \Pr[\mathsf{CheckGen}(1^\lambda) = k],$$

$$s(k) \coloneqq \Pr\left[\begin{array}{ll} \forall j \in \{v, \dots, n\} & k_v \coloneqq k \\ \mathsf{Ver}(\mathsf{ans}_j, k_j) = \top & : (k_{v+1}, \dots, k_n) \leftarrow \mathsf{CheckGen}^{n-v}(1^\lambda) \\ & \mathsf{puz}_j^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k_j) \text{ for } j \in [n] \\ & (\mathsf{ans}_1, \dots, \mathsf{ans}_n) \leftarrow \mathcal{A}(\mathsf{puz}_1^{\otimes t}, \dots, \mathsf{puz}_n^{\otimes t}) \end{array}\right]$$

$$a(k) \coloneqq \Pr\left[\begin{array}{ll} \forall j \in \{v+1, \dots, n\} & k_v \coloneqq k \\ \mathsf{Ver}(\mathsf{ans}_j, k_j) = \top & : (k_{v+1}, \dots, k_n) \leftarrow \mathsf{CheckGen}^{n-v}(1^\lambda) \\ & \mathsf{puz}_j^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k_j) \text{ for } j \in [n] \\ & (\mathsf{ans}_1, \dots, \mathsf{ans}_n) \leftarrow \mathcal{A}(\mathsf{puz}_1^{\otimes t}, \dots, \mathsf{puz}_n^{\otimes t}) \end{array}\right].$$

Note that $\mathsf{prefix}_{v-1} = (k_1, \dots, k_{v-1})$ is fixed and hardwired in the above definitions. By the definitions of $w(k)$, $s(k)$, and $\mathsf{rsp}(\mathsf{prefix}_{v-1})$ and Equation (17), we have

$$\sum_k w(k)s(k) = \mathsf{rsp}(\mathsf{prefix}_{v-1}) \geq \delta^{n-v+1}\left(1-\frac{1}{6q}\right). \tag{19}$$

[27] Note that the success or failure of $\mathcal{A}'$ is not determined by the execution of $\mathcal{A}'$ itself. It is determined after running the verification algorithm to verify the answer output by $\mathcal{A}'$. This is because we consider non-deterministic verification algorithm unlike [CHS05]. Thus, whenever we refer to the success of $\mathcal{A}$', we implicitly run the verification algorithm on its output.

By the definition of E and $a(k)$, for any $k \notin E$, we have

$$a(k) \leq \delta^{n-v}\left(1+\frac{1}{6q}\right). \tag{20}$$

By the definitions of $s(k)$ and $a(k)$, we have

$$\Pr[\mathsf{Ver}(\mathcal{A}'(\mathsf{puz}^{\otimes t'}),k)=\top|\mathcal{A}'(\mathsf{puz}^{\otimes t'}) \text{ does not abort}] = \frac{s(k)}{a(k)}. \tag{21}$$

We let

$$B := \left\{k : s(k) < \frac{\delta^{n-v+1}}{6q}\right\}. \tag{22}$$

The for any $k \notin B$, we have

$$\Pr[\mathcal{A}'(\mathsf{puz}^{\otimes t'}) \text{ aborts} : \mathsf{puz}^{\otimes t'} \leftarrow \mathsf{PuzzleGen}^{\otimes t'}(k)] < \left(1-\frac{\delta^{n-v+1}}{6q}\right)^{\lceil \frac{6q\ln(6q)}{\delta^{n-v+1}}\rceil} < \frac{1}{6q}. \tag{23}$$

Next, we have

$$\begin{aligned}\sum_{k\in B\cup E} w(k)s(k) &\leq \sum_{k\in B} w(k)s(k) + \sum_{k\in E} w(k)s(k)\\ &\leq \sum_{k\in B} w(k)\frac{\delta^{n-v+1}}{6q} + \sum_{k\in E} w(k)\\ &\leq \frac{\delta^{n-v+1}}{6q} + \frac{\delta^{n-v+1}}{6q} = \frac{\delta^{n-v+1}}{3q}\end{aligned}$$

where the second inequality follows from Equation (22) and the third inequality follows from Equation (18). Combined with Equation (19), we have

$$\sum_{k\notin B\cup E} w(k)s(k) \geq \delta^{n-v+1}\left(1-\frac{1}{2q}\right). \tag{24}$$

Then we have the following where $\mathsf{puz} \leftarrow \mathsf{PuzzleGen}(k)$ whenever puz appears:

$$
\begin{aligned}
&\Pr_k[\mathcal{A}'(\mathsf{puz}^{\otimes t'}) \text{ succeeds}] \\
&= \sum_k w(k) \Pr[\mathsf{Ver}(\mathcal{A}'(\mathsf{puz}^{\otimes t'}), k) = \top] \\
&\geq \sum_{k \notin B \cup E} w(k) \Pr[\mathcal{A}'(\mathsf{puz}^{\otimes t'}) \text{ does not abort}] \Pr[\mathsf{Ver}(\mathcal{A}'(\mathsf{puz}^{\otimes t'}), k) = \top | \mathcal{A}'(\mathsf{puz}^{\otimes t'}) \text{ does not abort}] \\
&\geq \sum_{k \notin B \cup E} w(k) \left(1 - \frac{1}{6q}\right) \frac{s(k)}{a(k)} \\
&\geq \sum_{k \notin B \cup E} w(k) \left(1 - \frac{1}{6q}\right) \frac{s(k)}{\delta^{n-v}\left(1 + \frac{1}{6q}\right)} \\
&= \frac{1 - \frac{1}{6q}}{\delta^{n-v}\left(1 + \frac{1}{6q}\right)} \sum_{k \notin B \cup E} w(k) s(k) \\
&\geq \frac{1 - \frac{1}{6q}}{\delta^{n-v}\left(1 + \frac{1}{6q}\right)} \delta^{n-v+1} \left(1 - \frac{1}{2q}\right) \\
&> \delta \left(1 - \frac{5}{6q}\right)
\end{aligned}
$$

where the second inequality follows from Equations (21) and (23), the third inequality follows from Equation (20), and the fourth inequality follows from Equation (24). As already explained, the above lower bound on the success probability suffices for completing the proof of Theorem 3.14. □

The Adversary $\mathcal{A}'(\mathsf{puz}^{\otimes t'})$
Preprocessing Phase

1. Initialize prefix to be an empty vector.
2. For $i = 1$ to $n-1$, do the following:
   (a) Run $k^* \leftarrow \mathsf{Extend}(\mathsf{prefix}, i)$.
   (b) If $k^* = \perp$, set $v \leftarrow i$ and go to **Online Phase**.
   (c) Update $\mathsf{prefix} \leftarrow \mathsf{prefix} \circ k^*$.
3. Set $v \leftarrow n$ and go to **Online Phase**.

Online Phase

1. If $v = n$, do the following:
   (a) Parse $\mathsf{prefix} = (k_1, \ldots, k_{n-1})$.
   (b) Run $\mathsf{puz}_i^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k_i)$ for $i \in [n-1]$.
   (c) Run $(\mathsf{ans}_1, \ldots, \mathsf{ans}_n) \leftarrow \mathcal{A}(\mathsf{puz}_1^{\otimes t}, \ldots, \mathsf{puz}_{n-1}^{\otimes t}, \mathsf{puz}^{\otimes t})$.
   (d) Output ans_n.
2. Otherwise, repeat the following $L = \lceil \frac{6q \ln(6q)}{\delta^{n-v+1}} \rceil$ times.
   (a) Parse $\mathsf{prefix} = (k_1, \ldots, k_{v-1})$.
   (b) Run $\mathsf{puz}_i^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k_i)$ for $i \in [v-1]$.
   (c) For $i = v+1$ to n, do the following:
      i. Run $k_i \leftarrow \mathsf{CheckGen}(1^\lambda)$
      ii. Run $\mathsf{puz}_i^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k_i)$.
   (d) Run $(\mathsf{ans}_1, \ldots, \mathsf{ans}_n) \leftarrow \mathcal{A}(\mathsf{puz}_1^{\otimes t}, \ldots, \mathsf{puz}_{v-1}^{\otimes t}, \mathsf{puz}^{\otimes t}, \mathsf{puz}_{v+1}^{\otimes t}, \ldots, \mathsf{puz}_n^{\otimes t})$.
   (e) Run $d_i \leftarrow \mathsf{Ver}(\mathsf{ans}_i, k_i)$ for $i \in \{v+1 \ldots, n\}$.
   (f) Output ans_v if $d_i = \top$ for all $i \in \{v+1 \ldots, n\}$.

   If none of the above repetitions outputs ans_v, then abort.

The Subroutine $\mathsf{Extend}(\mathsf{prefix}, i)$

1. Repeat the following $N_i = \lceil \frac{6q}{\delta^{n-i+1}} \ln(\frac{18qn}{\delta}) \rceil$ times:
   (a) Run $k^* \leftarrow \mathsf{CheckGen}(1^\lambda)$.
   (b) Run $\bar{\mu}_{k^*} \leftarrow \mathsf{Estimate}(\mathsf{prefix} \circ k^*, i)$
   (c) If $\bar{\mu}_{k^*} \geq \delta^{n-i}$, output k^*.
2. Output $\perp$.

The Subroutine $\mathsf{Estimate}(\mathsf{prefix}, i)$

1. Parse $\mathsf{prefix} = (k_1, \ldots, k_i)$.
2. Initialize $\mathsf{count} \leftarrow 0$.
3. Repeat the following $M_i = \lceil \frac{84q^2}{\delta^{n-i}} \ln(\frac{18qnN_i}{\delta}) \rceil$ times:
   (a) Run $\mathsf{puz}_j^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k_j)$ for $j \in [i]$.
   (b) For $j = i+1$ to n, do the following:
      i. Run $k_j \leftarrow \mathsf{CheckGen}(1^\lambda)$.
      ii. Run $\mathsf{puz}_j^{\otimes t} \leftarrow \mathsf{PuzzleGen}^{\otimes t}(k_j)$.
   (c) Run $(\mathsf{ans}_1, \ldots, \mathsf{ans}_n) \leftarrow \mathcal{A}(\mathsf{puz}_1^{\otimes t}, \ldots, \mathsf{puz}_n^{\otimes t})$.
   (d) If $\mathsf{Ver}(\mathsf{ans}_j, k_j) = \top$ for all $j \in \{i+1, \ldots, n\}$, increment $\mathsf{count} \leftarrow \mathsf{count} + 1$.
4. Output $\frac{\mathsf{count}}{M_i}$.

Figure 2: The Adversary $\mathcal{A}'$ for $(\mathsf{CheckGen}, \mathsf{PuzzleGen}, \mathsf{Ver})$