Pseudorandomness with Proof of Destruction and Applications

Amit Behera[1], Zvika Brakerski[2], Or Sattath[1], and Omri Shmueli[3]

[1]Computer Science Department, Ben-Gurion University of the Negev
[2]Weizmann Institute of Science
[3]Tel Aviv University

April 20, 2023

Abstract

Two fundamental properties of quantum states that quantum information theory explores are *pseudorandomness* and *provability of destruction*. We introduce the notion of *quantum pseudorandom states with proofs of destruction* (PRSPD) that combines both these properties. Like standard pseudorandom states (PRS), these are efficiently generated quantum states that are indistinguishable from random, but they can also be measured to create a classical string. This string is verifiable (given the secret key) and certifies that the state has been destructed. We show that, similarly to PRS, PRSPD can be constructed from any post-quantum one-way function. As far as the authors are aware, this is the first construction of a family of states that satisfies both *pseudorandomness* and *provability of destruction*.

We show that many cryptographic applications that were shown based on PRS variants using *quantum* communication can be based on (variants of) PRSPD using only *classical* communication. This includes symmetric encryption, message authentication, one-time signatures, commitments, and classically verifiable private quantum coins.

Contents

1 Introduction

A *Pesudorandom States* family (PRS), introduced in [JLS18]) is an efficiently samplamable family of pure states such that for any polynomial t, t-copies of a (pure) quantum state $|\phi\rangle$ sampled uniformly at random from the family is computationally indistinguishable from t-copies of a truly random state sampled from the Haar measure. On the other hand, a *provably destructible* family of quantum states is accompanied by two efficient quantum algorithms, $\mathit{Destruct}$, and Ver, such that running $\mathit{Destruct}$ on a state $|\phi\rangle$ sampled from the family, produces a classical proof $s_\phi \leftarrow \mathit{Destruct}(|\phi\rangle)$ that can be verified using Ver, such that given a copy of a sampled state $|\phi\rangle$ one cannot output both, the state $|\phi\rangle$ and a valid proof of destruction s_ϕ. Proofs of destructions as defined above (or variants of it) have served as a crucial property for many unclonable primitives, such as tokenized digital signatures [BDS16, CLLZ21, Shm22a], classically verifiable quantum money [MVW12, Shm22b], quantum lightning and its applications [Zha21, CS20, RS19], one-shot signatures [AGKZ20], etc.

As far as the authors are aware, there is no construction of a family of states that satisfies both pseudorandomness and provability of destruction. Previous constructions of provably destructible distributions were provably *not* pseudorandom. This stems from the fact that such techniques involved sampling a state that maintains its security only when a single copy is given. In fact, in most of these constructions (such as in [BDS16, CLLZ21]), given $O(n)$ copies of the sampled state, it is possible to not only tell the state from a Haar-random state but to completely characterize and efficiently generate the sampled state. On the side of pseudorandomness, previous techniques focused on sampled states that are uniform (or close to uniform) superpositions, with randomly sampled phases of the amplitudes. Since all known proof generation mechanisms $\mathit{Destruct}$ in the literature are essentially, measurements in the computational basis, these constructions with uniform superposition can not be provably destructible. In this work, we study how to combine both these notions in a single primitive.

> Is it possible to construct a provably destructible family of quantum states that is also pseudorandom?

In classical cryptography, one-way functions (OWF) are considered a minimal assumption for computational-cryptography, and they are also sufficient for many applications. In the quantum setting, in contrast, (post-quantum) one-way functions are sufficient but do *not* appear to be necessary for a variety of cryptographic tasks such as symmetric encryption, digital signatures, message authentication codes, and commitments. Specifically, Ref. [JLS18] showed that one-way functions are sufficient to build PRS, but Kretschmer [Kre21] showed a black-box separation in the other direction, thus implying that OWFs are not necessary for PRS. Several recent works showed that PRS suffices to imply the aforementioned cryptographic applications (or variants thereof), without using OWF. For example, statistically-binding bit-commitment protocols have been shown based on PRS [AQY21, MY22a](see Section 1.4 for other related works). However, these constructions used a different syntax than their classical counterparts—in particular in requiring *quantum* communication.

One of the aims of this work is to investigate whether this change is necessary:

> Is it possible to achieve cryptographic applications without quantum communication based on a pseudorandom states variant?

Indeed, this question has also been recently addressed by Ananth, Gulati, Qian and Yuen [AGQY22], who have shown statistically binding bit commitment and pseudo-encryption with classical communication. Their constructions were based on variants of PRS (namely, short output PRS and short output PRFS).

1.1 Our Results

Our first contribution, in Section 2, is defining the notion of proofs of destruction in the context of pseudorandom states, which addresses the first question raised above, see Page 3. In a PRS with proof of destruction (PRSPD), we augment a $\mathcal{Destruct}$ algorithm, which takes the pseudorandom state, and generates a classical proof; and a $\mathcal{Ver}$ algorithm, which takes a proposed proof and a key, and either accepts or rejects. We require that valid proofs should be accepted with certainty. In terms of security, we add the Unforgeability-of-proofs requirement, which guarantees that given t copies of the pseudorandom state, it should be hard to produce $t+1$ distinct proofs of destruction. We extend the notion of proofs of destruction to a variant of PRS, called pseudorandom function-like states (PRFS), that was introduced in [AQY21] (see Section 1.4 for further discussion). In a PRFS, the seed k should allow to efficiently generate a state for any input x, such that the states generated for different x's should jointly be indistinguishable from a random state. Namely, an adversary can choose $x_1, \ldots, x_m$, and should not be able to distinguish between $\bigotimes_{i\in[m]} |\psi_{k,x_i}\rangle$, where $|\psi_{k,x}\rangle$ are generated from the PRFS family, and $\bigotimes_{i\in[m]} |\varphi_{x_i}\rangle$ where $|\varphi_x\rangle$ is sampled from the Haar measure. We import the notion of Unforgeability-of-proofs to PRFS and define the notion of PRFSPD. We then proceed, in Section 3, to show how to construct PRSPD and PRFSPD from any post-quantum one-way function, which requires extending existing proof techniques for the construction of these primitives. Currently, we do not have a candidate construction of PRSPD or PRFSPD that does not use one-way functions directly.

Finally, in Section 4, we show how pseudorandom states (and function-like states) with proof of destruction can be used to achieve almost all of the existing known applications of pseudorandom states (and function-like states, respectively), without the need for quantum communication, thereby addressing the second question mentioned above, see Page 4. Specifically, we construct :

1. Length-restricted one-time secure digital signatures (Section 4.1), and classically-verifiable private quantum coins[1] (Section 4.2) from any PRSPD.
2. A computational-hiding and statistically-binding bit commitment from PRSPD in which the proofs satisfy some *nice* properties, which we denote by PRSNPD—see Section 4.3 for details. While we do not know how to construct such PRSNPD from PRSPD or PRFSPD, our construction satisfies this niceness property.
3. CPA symmetric encryption (Section 4.5) and strong-CMA MAC (Section 4.4) from any PRFSPD. Note that this form of encryption is known to imply garbled circuits (Appendix D).

1.2 Our Techniques

Our construction of PRSPD is based on the following observation. Prior constructions starting with [JLS18] showed that a uniform superposition over all computational basis elements, with a random phase, constitutes a PRS. Formally, the family $|\psi_k\rangle = \frac{1}{\sqrt{N}} \sum_{y\in\{0,1\}^n} \omega_N^{\mathsf{PRF}_k(y)}|y\rangle$ is a PRS family whenever PRF_k is a post-quantum PRF from n bits to n bits, where $N = 2^n$ and ω_N is the N-th root of unity. We show that a state which is supported on a pseudorandom subset of computational basis elements is still a PRS. More precisely, for a pseudorandom permutation PRP on $4n$ bits, let $A_{k'} = \{\mathsf{PRP}_{k'}(z||0^{3n}) : z \in \{0,1\}^n\}$. We prove that the following states form a pseudorandom family:

$$|\psi_{k,k'}\rangle = \frac{1}{\sqrt{N}} \sum_{y\in A_{k'}} \omega_N^{\mathsf{PRF}_k(y)}|y\rangle. \tag{1}$$

This modification allows us to generate a proof of destruction as follows. The state $|\psi_{k,k'}\rangle$ is measured in the computational basis, resulting in a (uniformly random) element of $A_{k'}$, which we denote by p. The verification procedure for p is to apply $\mathsf{PRP}_{k'}^{-1}(p)$ and checking that the result is of the form $z||0^{3n}$ for some string z. We show that this construction satisfies the Unforgeability-of-proofs property.

[1] In this primitive, the verification is quantum, but sending the proof of possession to the verifier only requires classical communication.

We observe a property of our construction from which it is easy to deduce both—the pseudorandomness property and the unforgeability of proofs property. We recall a property of the Haar-random distribution over quantum states. The following distributions (over quantum states) are equivalent: (i) Sample an n-qubit Haar-random state and output t copies of this state. (ii) Sample t elements from $\{0,1\}^n$, according to some distribution, and output a superposition over all $t!$ permutations of this t-tuple. In fact, the distribution can be $i.i.d$ uniform over the domain, with only a negligible effect on the outcome.

Now, if we sample the t elements not from the entire domain, but rather from a large enough random sub-domain, the distribution over tuples will remain statistically indistinguishable. We can apply this logic twice: First, to derive pseudorandomness, since a random state over a random subdomain is indistinguishable from a random state over the entire domain. Second, to derive the unforgeability of proofs, since providing t samples of the PRSPD state is statistically indistinguishable from a process that only uses t classical values from the sub-domain. Thus, coming up with an additional element in this random sub-domain can be done with at most negligible probability for classical information-theoretic reasons. We further show that experiment (ii) above is statistically close to experiment (iii): Sampling an exponential size subdomain A and a random function f and preparing t copies of the state $|\psi_{A,f}\rangle \propto \sum_{x\in S} \omega_N^{f(x)}|x\rangle$. Experiment (iii) and experiment (iv) in which t copies of the PRSPD states in Eq. (1) can now be seen to be computationally indistinguishable, by the pseudorandomness properties of the PRF and PRP functions. Transitions (ii)-(iv) are formalized in our main technical lemma, Lemma 6.

Extending this idea to PRFSPD is done in a straightforward manner, starting from the PRFS construction of [AGQY22]. Our PRFSPD family can be thought of as $|\psi_{(k,k'),x}\rangle = \frac{1}{\sqrt{N}} \sum_{y\in A_{k',x}} \omega_N^{\mathsf{PRF}_k(x,y)}|y\rangle$, where $A_{k',x} = \{\mathsf{PRP}_{k'}(y||x||0^{3n}) : y \in \{0,1\}^n\}$ for $x \in \{0,1\}^n$. The destruction is done as before, and verification checks that p has the form $y||x||0^{3n}$.

How to use pseudorandom states with proof of destruction. In many cases, a template can be used to remove the quantum communication from a protocol involving pseudorandom states. Several protocols use PRS in the following manner. In the first partof the protocol, a pseudorandom state $|\psi_k\rangle$ is generated and sent via quantum communication. In a later stepof the protocol, a testing procedure is applied to check if the state is indeed $|\psi_k\rangle$. In order to remove the quantum communication, we send the (classical) proof of destruction of it (instead of the state itself). Furthermore, we replace the testing whether the state is the "correct" state, with verifying that the proof of destruction is valid. This approach can also be applied with PRFS, where the state is $|\psi_{k,x}\rangle$ used.

Next, we demonstrate the use of the template above with a concrete example. Ref. [AQY21] constructs a MAC scheme using PRFS, in which the secret key is a random k, $\mathcal{Sign}_k(m)$ generate a *quantum* signature $|\psi_{k,m}\rangle$, and $\mathcal{Verify}_k(m, |\varphi\rangle)$ is done by testing procedure discussed above, which tests whether $|\varphi\rangle$ is the expected state $|\psi_{k,m}\rangle$. In our scheme(see Section 4.4), $\mathcal{Sign}_k(m)$ is done by preparing $|\psi_{k,m}\rangle$, and the *classical* signature is the proof of destruction, denoted p, of this state. Clearly, the signature can now be sent via a classical channel. The testing procedure above is replaced with checking that p is a valid proof of destruction for (k, m).

The template above is indeed useful as a conceptual framework, but applying it sometimes requires consideration of the specifics of the primitives. Some specific challenges that need to be addressed are as follows.

1. Pseudorandom states are pseudorandom as quantum states, but the proofs of destruction are not required to be pseudorandom strings. For example, one can easily transform a PRSPD scheme to one in which the first bit of the proof of destruction is always 0.

   This issue comes up in the context of bit-commitment. Ref. [MY22a] shows a construction that can be viewed as a quantum analog of Naor's commitments from PRG. There, we need to make the additional assumptions that the proofs *are* pseudorandom in order to prove the hiding property—see Section 4.3.

   Recall that Naor's construction also requires a length-tripling PRG to prove the hiding. For analogous reasons, in our setting, we need a PRFSPD in which every key k accepts only a small fraction of the potential proofs.

   We define a PRSPD in which the proofs of destruction satisfy these *nice* properties as PRSNPD.

2. Pseudorandom states are known to be uncloneable [JLS18, Theorem 2], but the proofs of destruction are classical and, therefore, can trivially be copied. We are only guaranteed that generating *new* proofs of destruction is hard. This difference means that for our quantum coins scheme to be secure, the bank needs to keep a copy of the proofs that were already accepted, so these would be rejected in further attempts. In other words, unlike the quantum coin scheme proposed by [JLS18], our quantum coin protocol is *stateful*—see Section 4.2for details.

3. It can be shown that PRS are non-invertible in the following sense: Given $|\psi_k\rangle$, one cannot find k' such that $|\psi_{k'}\rangle$ has a non-negligible overlap with $|\psi_k\rangle$ [MY22a, Lemma 4.1]. An analogous property does not necessarily hold for proofs of destruction: Given a proof p for $|\psi_k\rangle$, one might be able to find k' such that p is a valid proof of this destruction for k'. For example, given a PRSPD scheme, one can

modify it so $k' = 00\dots0$ accepts all proofs of destruction. Since that particular k' has a negligible probability of getting sampled as the key, it has no effect on the security of the scheme. But now, given a proof of destruction p, it is trivial to find a k' such that the proof of destruction is accepted. This issue arises in the context of one-time digital signatures, which we expand upon next.

To illustrate an example of such a challenge, let us describe our construction of one-time signatures from PRSPD. We recall Lamport's one-time signature scheme and assume that we only wish to sign one-bit messages (the extension to multiple bits is by repetition, as in the classical case). The idea in Lamport's OWF-based scheme is to sample uniformly random x_0, x_1 as the signing key, set $y_0 = f(x_0), y_1 = f(x_1)$ as a verification key, and set the signature on message $m \in \{0, 1\}$ to be x_m. This was adapted to PRS by [MY22a], by replacing f with the PRS generator algorithm.

We wish to convert our quantum verification key to being classical using PRSPD. We achieve this by replacing the PRS states with their respective proofs of destruction p_0, p_1. The signature will be the key associated with the proof of destruction. However, contrary to the classical and PRS settings, we must take a different approach here. A forgery here consists of a PRSPD key k'_m which verifies p_m; indeed, if we were guaranteed that $k'_m = k_m$ then we would have been done since unforgeability of proofs would have been used in order to complete the security proof. However, this is not the case, and the unforgeability of proofs alone is insufficient: see Item 3 above.

To rule out "junk keys"—keys which accept too many proofs of destruction—we apply two modifications: the public key consists of a large number of proofs $\vec{p}_m$ for every value of m (where all proofs of destruction are generated using the same key), instead of just one. We know that all of these proofs of destruction will get accepted by the PRSPD verification by the key that generated these states. We also modify the signature verification algorithm so that given a signature k'_m, it first samples a large polynomial number of proofs of destruction with freshly random keys, and makes sure that k'_m is not verifying garbage (honestly generated keys will pass this test with overwhelming probability). Only after passing this test will the forgery be tested against $\vec{p}_m$. One can easily see that this method rules out simple "junk keys" that accept all proofs, as in the example described above. The full security proof uses a hybrid argument where the public key is not generated using a key k_m of the PRSPD, but instead, it is generated by applying the destruction algorithm to a *Haar random quantum state.*[2] This can only have a negligible change on the forgery probability by the pseudorandomness of the PRSPD. Interestingly, the construction by [MY22a] *did not* use the pseudorandomness property and relied on a weaker notion called

[2] One may be concerned that true random states are infeasible to generate, however for our purposes here we can use so-called "state-designs" instead of true random states.

one-way state generators. We then show that an adversary which receives such "garbage" proofs p_m (i.e. proofs which are generated by the proof of destruction procedure on Haar random states) cannot provide forgery.

1.3 Open Problems

- A primary motivation to study pseudorandom states is that it seems as a weaker assumption than one-way functions, on which quantum cryptography could be based upon. Unfortunately, this separation result only holds for some of the PRS-variants. No such separation result is known for short-output PRS and short-output PRFS. Note that some of the applications prior to ours rely upon those. Similarly, we did not prove a similar separation for PRSPD and PRFSPD, and these challenges are left as an open problem.

- Does PRSPD imply short-input PRFSPD, i.e., PRFSPD with logarithmic input length? Ref. [AQY21] constructs short-input PRFS from PRS generically by measuring the first $\log(\lambda)$ qubits and post-selecting the outcome being the input. The same approach may not work in the case of PRFSPD and PRSPD because the post-selection procedure as proposed in [AQY21] may not commute with the $\mathcal{D}estruct$ algorithm for general PRSPD. An alternate yet related approach would be to run the $\mathcal{D}estruct$ algorithm on the input state without measurement, then measure only the first $\log(\lambda)$ qubits, post-select on the outcome being the input, and output the state on the unmeasured registers as the PRFSPD state.

  The hope is that if the starting state was $\mathcal{H}aar$ random, then the state on the unmeasured bits will be Haar random. However, the destruct algorithms may use ancillae qubits, and therefore the overall process becomes *non-unitary*, even before the measurement. Since non-unitary processes do not preserve $\mathcal{H}aar$-random property, if we measure the first $\log(\lambda)$ registers, the state on the rest of the registers might not be statistically close to $\mathcal{H}aar$ random.

1.4 Related works

Quantum forms of pseudorandomness have seen rapid development, which we summarize in this section. All the results mentioned are depicted in Fig. 1 which also contains our main results.

The study of pseudorandom state generators (PRS) was initiated by Ji, Liu, and Song [JLS18]. They proved a construction based on the existence of post-quantum one-way functions. Ji, Liu, and Song's PRS construction were simplified in Ref. [BS19].

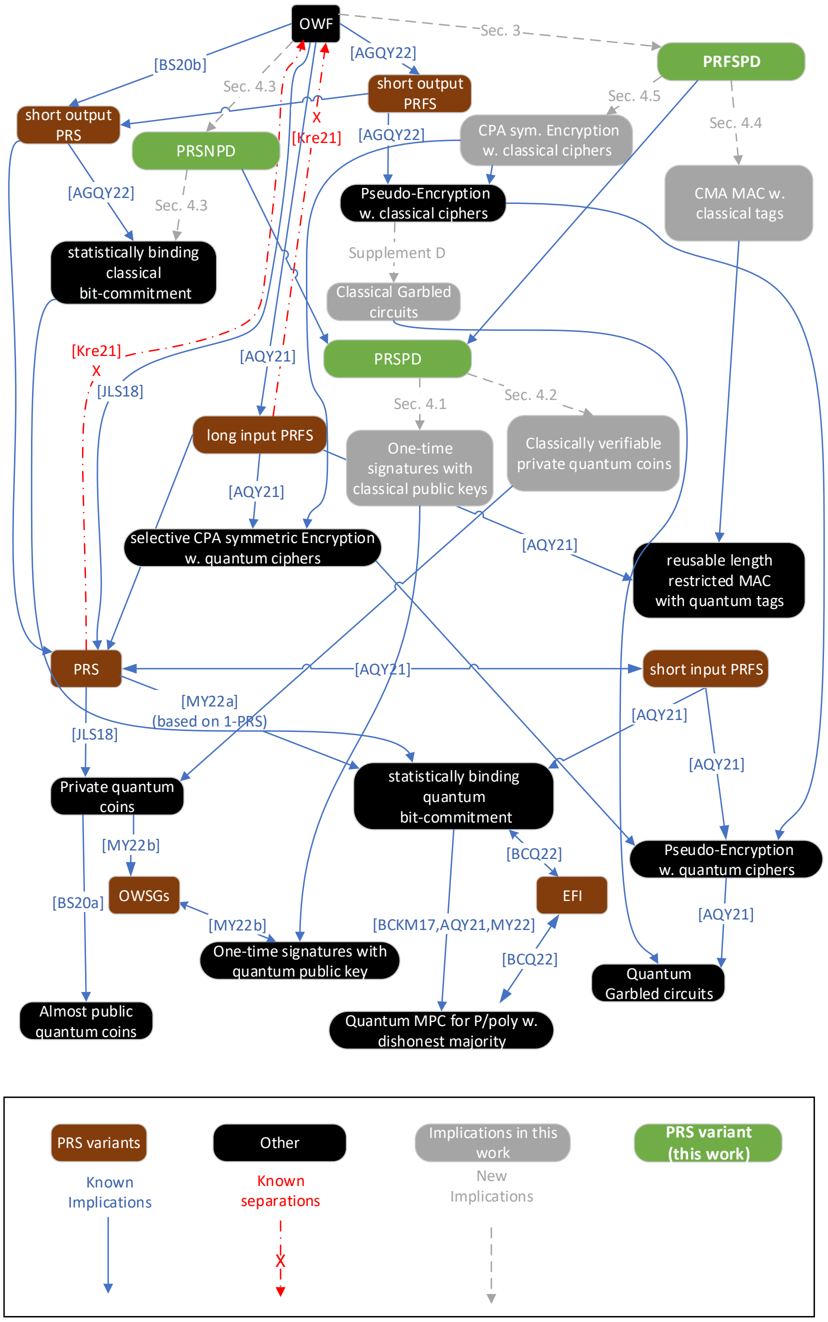
OWF
Sec. 3
[BS20b]
[AGQY22]
PRFSPD
Sec. 4.3
short output PRFS
short output PRS
Sec. 4.5
X
[Kre21]
CPA sym. Encryption w. classical ciphers
PRSNPD
[AGQY22]
Sec. 4.4
[AGQY22]
Sec. 4.3
Pseudo-Encryption w. classical ciphers
CMA MAC w. classical tags
statistically binding classical bit-commitment
Supplement D
Classical Garbled circuits
[Kre21]
X
[AQY21]
PRSPD
[JLS18]
Sec. 4.1
Sec. 4.2
long input PRFS
One-time signatures with classical public keys
Classically verifiable private quantum coins
[AQY21]
selective CPA symmetric Encryption w. quantum ciphers
[AQY21]
reusable length restricted MAC with quantum tags
PRS
[AQY21]
short input PRFS
[MY22a] (based on 1-PRS)
[AQY21]
[JLS18]
[AQY21]
Private quantum coins
statistically binding quantum bit-commitment
[MY22b]
[BCQ22]
Pseudo-Encryption w. quantum ciphers
[BS20a]
OWSGs
EFI
[MY22b]
[BCKM17,AQY21,MY22]
[AQY21]
One-time signatures with quantum public key
[BCQ22]
Quantum Garbled circuits
Almost public quantum coins
Quantum MPC for P/poly w. dishonest majority
PRS variants
Other
Implications in this work
PRS variant (this work)
Known Implications
Known separations
X
New Implications

Figure 1: Various applications of PRS and its variants. Best viewed in color. Unless stated otherwise, all applications are protocols that require *quantum communication*.

Kretschmer [Kre21] proved a separation between one-way functions and PRSs: there exists a quantum oracle relative to which PRSs exist, but one-way functions do not exist. In other words, there is no black-box reduction from PRS to one-way functions (while a black-box reduction in the other direction is implied by [JLS18]).

Several variants of PRS have been introduced, all of which are implied by post-quantum one-way functions. These different variants will play an important role when we discuss the applications. In Ref. [BS20b], the authors show how to construct a *scalable* PRS based on OWFs; in this context, scalability means that for any function $n(\lambda) \leq \lambda$, one can construct a PRS with $n(\lambda)$ qubits. Perhaps counter-intuitively, and unlike pseudorandom generators, constructing pseudorandom states with a smaller number of qubits n seems harder (and definitely does not follow from the definition).

In [AQY21], Ananth, Qian, and Yuen define pseudorandom function-like states (PRFS). An (d, n)-PRFS generator receives a key $k \in \{0,1\}^\lambda$ and an input $x \in \{0,1\}^d$ and outputs an n-qubit state[3]. In the security game, the adversary can choose (in advance) a set of inputs $x_1, \ldots, x_m$; the challenger either picks a random k and returns the PRFS states associated with $(k, x_1), (k, x_2), \ldots, (k, x_m)$, or samples a Haar random state for each distinct x_i, and send the states to the adversary. The adversary needs to distinguish between these two cases. They show how to construct a (n, d)-PRFS for $d = O(\log \lambda)$ from any $(n + d\lambda)$-PRS. We refer to PRFS in that regime (namely, $d = O(\log \lambda)$) as *short input* PRFS. They show how to construct a PRFS with $\omega(\log(\lambda))$ input length, which we refer to as long input PRFS, from any OWF. They also show that long input PRFS is separated from OWFs. It is not known how to construct long input PRFS from a short input PRFS. It is known that, similarly to vanilla PRS, *short* and *long* input PRFS are separated from post-quantum one-way functions [AQY21].

Several applications of PRSs have been shown. In [JLS18], it was shown that PRS implies a private quantum coin scheme—i.e., a private quantum money scheme in which all the quantum money states are exact copies. In [BS20a], an almost public quantum coin scheme was shown based on the existence of any private coin scheme. In this context, *public* means that users can verify a quantum coin without the bank. The scheme was *almost* public because it has several limitations. For example, it only achieves rational unforgeability; and the users must have coins in order to verify other coins. Note that there are no other *public* quantum money schemes based on one-way functions. Morimae and Yamakawa [MY22a] construct a length-restricted one-time signature (also known as Lamport signature) with a quantum public key.

PRFS has several applications, which depend on the parameters of the

[3] For technical reasons which are outside the scope of this work, the algorithm can output abort.

PRFS. We start with those which are implied by the weakest form, namely, short input PRFS. Ref. [AQY21] construct a symmetric pseudo-encryption with quantum ciphers, which achieves one-time security. Pseudo-encryption means that the key is *shorter* than the length of the encrypted message—which is impossible to achieve unconditionally. This result requires a $\omega(\log \lambda)$-PRS or alternatively, an (n, d)-PRFS with $d > \log \lambda$ and $n = \omega(\log \lambda)$. As observed by [AQY21], garbled circuits can be constructed from the symmetric pseudo-encryption mentioned above. Note that in this construction, the original circuit is classical, and the resulting garbled circuit is quantum. They also construct a statistically binding quantum bit-commitment from a $(2 \log \lambda) + \omega(\log \log \lambda))$-PRS (or, alternatively, an (n, d)-PRFS satisfying $2^d \cdot n \geq 7\lambda$); and by adapting the result in [BCKM21], they construct multi-party computation in the dishonest majority setting based on the same assumption.

Ref. [AQY21] also shows three other constructions based on *long input* PRFS: Symmetric encryption scheme secure against selective CPA with quantum ciphers based on $(\omega(\lambda), \omega(\lambda))$-PRFS; a reusable MAC with quantum tags, which is length restricted to $\ell(\lambda)$ bits, based on a (d, n)-PRFS with $d(\lambda) \geq \ell(\lambda)$ and $d = \omega(\log \lambda)$;

Recently, in [AGQY22], the authors report on statistically binding commitments and pseudo-encryption with *classical* communication. Their construction is based on *short output* PRFS, namely, $\log(\lambda)$ output and input sizes.

The notion of a PRS was used outside the context of cryptography in the study of the wormhole growth paradox [BFV20] and quantum machine learning.

From the results mentioned so far, there is no indication that PRS (or any of its variants) are *minimal* assumptions for the cryptographic task that they can be used to achieve. Two recent works address this aspect: (a) Ref. [BCQ22] shows that EFI pairs—efficiently samplable, statistically far but computationally indistinguishable pairs of mixed quantum states—are equivalent to statistically binding quantum commitments, oblivious transfer, and several other functionalities. (b) Ref. [MY22a, MY22b] proved that a one-way state generator (OWSG) is equivalent to one-time signatures with quantum public keys.

We mention that OWSGs are known to be implied from private quantum coins and that a variant called secretly-verifiable and statistically invertible one-way state generator (SV-SI-OWSG) is equivalent to EFI [MY22b].

2 Pseudorandom States and Function-like States with Proofs of Destruction

In this section, we define pseudorandom states and function-like states with proofs of destruction and study some important properties and distributions related to them.

2.1 Core definitions

Definition 1 (Pseudorandom state generator with proofs of destruction). *A PRSPD scheme with key-length $w(\lambda)$, output length $n(\lambda)$ and proof length $c(\lambda)$ is a tuple of QPT algorithms $(\mathcal{G}en, \mathcal{D}estruct, \mathcal{V}er)$ with the following syntax:*

1. *$|\psi_k\rangle \leftarrow \mathcal{G}en(k)$: takes a key $k \in \{0,1\}^{w(\lambda)}$, and outputs an $n(\lambda)$-qubit pure state*[4] *$|\psi_k\rangle$.*

2. *$p \leftarrow \mathcal{D}estruct(|\phi\rangle)$: takes an $n(\lambda)$-qubit quantum state $|\phi\rangle$, and outputs a $c(\lambda)$-bit classical string, p.*

3. *$b \leftarrow \mathcal{V}er(k,p)$: takes a key $k \in \{0,1\}^{w(\lambda)}$, a $c(\lambda)$-bit classical string p and outputs a boolean output b.*

Correctness. *A PRSPD scheme is said to be correct if*

$$\Pr_{k \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}} [1 \leftarrow \mathcal{V}er(k,p) \mid p \leftarrow \mathcal{D}estruct(|\psi_k\rangle); |\psi_k\rangle \leftarrow \mathcal{G}en(k)] = 1$$

Security.

1. $\mathsf{Pseudorandomness}$: *A PRSPD scheme is said to be pseudorandom if for any QPT adversary $\mathcal{A}$, and any polynomial $m(\lambda)$, there exists a negligible function* $\mathsf{negl}(\lambda)$, *such that*

$$\left| \Pr_{|\psi_k\rangle \leftarrow \mathcal{G}en(k); k \leftarrow \{0,1\}^w} [\mathcal{A}(|\psi_k\rangle^{\otimes m}) = 1] - \Pr_{|\phi\rangle \leftarrow \mu_{(\mathbb{C}^2)^{\otimes n}}} [\mathcal{A}(|\phi\rangle^{\otimes m}) = 1] \right| = \mathsf{negl}(\lambda).$$

2. $\mathsf{Unforgeability\text{-}of\text{-}proofs}$*: A PRSPD scheme satisfies* $\mathsf{Unforgeability\text{-}of\text{-}proofs}$ *if for any QPT adversary $\mathcal{A}$ in forging game (Game 1), there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{PRSPD}} = 1] = \mathsf{negl}(\lambda).$$

[4] The pseudorandom security guarantee implies that with overwhelming probability over the chosen key, the state should be negligibly close to a pure state in trace distance; otherwise, pseudorandomness of the state can be violated via Swap-test.

Game 1 $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{PRSPD}}$

```
Challenger samples $k \in \{0,1\}^{w(\lambda)}$ uniformly at random.
$\mathcal{A}^{\mathcal{G}en(k),\mathcal{V}er(k,\cdot)}(1^\lambda)$ outputs $p_1, p_2, \ldots, p_{t+1}$ to the challenger.
Adversary wins if: i) all $p_i$'s are distinct, ii) the number of queries made
    to the $\mathcal{G}en(k)$ oracle was t, and iii) $\mathcal{V}er(k, p_i) = 1$ for $1 \leq i \leq t+1$.
```

Definition 2 (Pseudorandom function-like state generator with proofs of destruction)**.** *A PRFSPD scheme with key-length $w(\lambda)$, input-length $d(\lambda)$, output length $n(\lambda)$ and proof length $c(\lambda)$ is a tuple of QPT algorithms $(\mathcal{G}en, \mathcal{D}estruct, \mathcal{V}er)$ with the following syntax:*

1. *$|\psi_k^x\rangle \leftarrow \mathcal{G}en(k,x)$: takes a key $k \in \{0,1\}^w$, an input string $x \in \{0,1\}^{d(\lambda)}$, and outputs an n-qubit pure state $|\psi_k^x\rangle$.*

2. *$p \leftarrow \mathcal{D}estruct(|\phi\rangle)$: takes an n-qubit quantum state $|\phi\rangle$ as input, and outputs a c-bit classical string, p.*

3. *$b \leftarrow \mathcal{V}er(k,x,p)$: takes a key $k \in \{0,1\}^w$, a d-bit input string x, a c-bit classical string p and outputs a Boolean output b.*

Correctness. *A PRFSPD scheme is said to be correct if for every $x \in \{0,1\}^d$,*

$$\Pr_{k \xleftarrow{u} \{0,1\}^w} [1 \leftarrow \mathcal{V}er(k,x,p) \mid p \leftarrow \mathcal{D}estruct(|\psi_k^x\rangle); |\psi_k^x\rangle \leftarrow \mathcal{G}en(k,x)] = 1$$

Security.

1. *Pseudorandomness: A PRFSPD scheme is said to be quantum adaptively pseudorandom if for any QPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\lambda)$, such that the following absolute value is bounded by $\mathsf{negl}(\lambda)$,*

$$\left| \Pr_{k \leftarrow \{0,1\}^w} [\mathcal{A}^{\mathcal{G}en(k,\cdot)}(1^\lambda) = 1] - \Pr_{\forall x \in \{0,1\}^d, |\phi^x\rangle \leftarrow \mu_{(\mathbb{C}^2)^{\otimes n}}} [\mathcal{A}^{|\mathcal{H}aar^{\{|\phi^x\rangle\}_{x\in\{0,1\}^d}}\rangle(\cdot)}(1^\lambda) = 1] \right|, \tag{2}$$

where $\forall x \in \{0,1\}^d$, $\mathcal{H}aar^{\{|\phi^x\rangle\}_{x\in\{0,1\}^d}}(x)$ outputs $|\phi^x\rangle$. Here $\mathcal{A}^{\mathcal{G}en(k,\cdot)}$ represents that $\mathcal{A}$ gets classical oracle access to $\mathcal{G}en(k,\cdot)$.

2. Unforgeability-of-proofs*: A PRFSPD scheme satisfies* Unforgeability-of-proofs *if for any QPT adversary $\mathcal{A}$ in forging game (Game 2), there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathit{PRFSPD}} = 1] = \mathsf{negl}(\lambda).$$

Game 2 $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{PRFSPD}}$

```
Given input $1^\lambda$, Challenger samples $k \leftarrow \{0,1\}^{w(\lambda)}$ uniformly at random.
Initialize an empty set of variables, $S$.
$\mathcal{A}$ gets oracle access to $\mathcal{Gen}(k,\cdot)$, $\mathcal{Ver}(k,\cdot,\cdot)$ as oracle.
for $\mathcal{Gen}$ query $x$ made by $\mathcal{A}$ do
    if $\exists$ variable $t_x \in S$ then $t_x = t_x + 1$.
    else Create a variable $t_x$ in $S$, initialized to 1.
    end if
end for
$\mathcal{A}$ outputs $x, p_1, p_2, \ldots, p_{t_x+1}$ to the challenger.
Challenger rejects if $p_i$'s are not distinct.
for $i \in [m+1]$ do $b_i \leftarrow \mathcal{Ver}(k, x, p_i)$
end for
Return $\wedge_{i=1}^{m+1} b_i$.
```

Remark 1. A pseudorandom state generator or PRS(respectively, pseudorandom function-like state generator or PRFS) is the same as PRSPD (respectively, PRFSPD), but without the $\mathcal{Destruct}$ and $\mathcal{Ver}$ algorithms, and the correctness and Unforgeability-of-proofs requirements.

Remark 2. The pseudorandomness guarantee for PRFSPD can be strengthened by giving the adversary *quantum* adaptive access (instead of classical adaptive access) to the $\mathcal{Gen}$ oracle in Eq. (2). This strengthened notion, called quantum adaptive pseudorandomness, has been considered for PRFS in Ref. [AGQY22]. In this work, we only consider classical adaptive pseudorandomness for PRFSPD because it is sufficient for all of our applications. Nevertheless, our construction of PRFSPD in Section 3 can be proven to be quantum adaptive pseudorandom via techniques similar to that in [AGQY22], namely Zhandry's *small-range distributions* [Zha12].

Remark 3 (PRFSPD Input Shortening and PRSPD). PRFSPD with input length d immediately implies PRFSPD with input length $d' \in \{0, 1, \cdots, d\}$ and in particular PRSPD. To see this, if $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ is a PRFSPD with input length d, then for any $d' \in \{0, 1, \cdots, d\}$, consider the d'-input-length scheme $\mathsf{PRFSPD}_{d'} = (\mathcal{Gen}', \mathcal{Destruct}, \mathcal{Ver}')$ where for $x' \in \{0,1\}^{d'}$:

- $\mathcal{Gen}'(\cdot, x') = \mathcal{Gen}(\cdot, (x' || 0^{d-d'}))$.
- $\mathcal{Ver}'(\cdot, x', \cdot) = \mathcal{Ver}(\cdot, (x' || 0^{d-d'}), \cdot)$.

This is similar to how pseudorandom function-like states imply pseudorandom states: A reduction that takes an adversary against the new scheme and attaches $d - d'$ zeros to its queries shows that we can use it in order to break the original scheme. Finally, PRFSPD with input length 0 exactly implies the definition of a PRSPD.

Remark 4 (Computational assumptions are necessary for PRSPD and PRFSPD)*.* Clearly, PRSPD implies PRS which cannot exist unconditionally [Kre21, AGQY22], hence PRSPD (therefore PRFSPD) cannot exist unconditionally.

2.2 Distributions related to the $\mathcal{Destruct}$ algorithm of PRSPD, PRFSPD, and Haar random states

Definition 3 (Correlated and independent destructions for Haar random states)**.** *For any algorithm $\mathcal{Destruct}$, that take a n-qubit state as input and outputs a c-bit classical string as output, and for every $t \in \mathsf{poly}(\lambda)$, $\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}$ is the distribution on $\{0,1\}^{ct}$ given by, $(f_1,\dots,f_t) \sim \mathsf{Correlated\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}$ where*

$$(f_1,\dots,f_t) \leftarrow \mathcal{Destruct}^{\otimes t}(|\phi\rangle^{\otimes t}); |\phi\rangle \sim \mu_{\mathcal{H}_n}.$$

For every $t \in \mathsf{poly}(\lambda)$, let $\mathsf{Product\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}$ be the t-fold product of $\mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}$ which is given by

$$f \sim \mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}} \equiv f \leftarrow \mathcal{Destruct}(|\phi\rangle); |\phi\rangle \sim \mu_{\mathcal{H}_n}.$$

Definition 4 (Correlated and independent destructions of PRSPD and PRFSPD)**.** *For any PRSPD family $\mathcal{PRSPD} = (\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ and for every $t \in \mathsf{poly}(\lambda)$, $\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRSPD}}$ is the distribution on $\{0,1\}^{ct}$ given by, $(f_1,\dots,f_t) \sim \mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRSPD}}$ where*

$$(f_1,\dots,f_t) \leftarrow \mathcal{Destruct}^{\otimes t}(|\psi_k\rangle^{\otimes t}); |\psi_k\rangle \leftarrow \mathcal{Gen}(k), \text{ where } k \xleftarrow{u} \{0,1\}^w.$$

For every $t \in \mathsf{poly}(\lambda)$, let $\mathsf{Product\text{-}Destruction}_t^{\mathcal{PRSPD}}$ be the t-fold product of $\mathsf{Product\text{-}Destruction}^{\mathcal{PRSPD}}$ which is given by

$$f \sim \mathsf{Product\text{-}Destruction}^{\mathcal{PRSPD}} \equiv f \leftarrow \mathcal{Destruct}(|\phi_k\rangle); |\phi_k\rangle \leftarrow \mathcal{Gen}(k), \text{ where } k \xleftarrow{u} \{0,1\}^w.$$

For any PRFSPD family $\mathcal{PRFSPD} = (\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$, for any $x \in \{0,1\}^d$ and for every $t \in \mathsf{poly}(\lambda)$, let $\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRFSPD},x} = \mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRFSPD}_x}$ and $\mathsf{Product\text{-}Destruction}^{\mathcal{PRFSPD},x} = \mathsf{Product\text{-}Destruction}^{\mathcal{PRFSPD}_x}$, where $\mathcal{PRFSPD}_x$ is the PRSPD scheme obtained out of $\mathcal{PRFSPD}$ by fixing the input to x, see Definition 2 and Remark 3.

2.3 Properties of Pseudorandom States and Function-like States with Proofs of Destruction

In this section, we state a few properties of PRSPD and PRFSPD, that would be important for the applications in Section 4. These properties (Lemmas 1 to 5) are true for arbitrary PRSPD and PRFSPD, but due to space constraints, we only sketch the proofs in this version. For simplicity, some of the proofs are sketched only for a special case, where the $\mathcal{D}\mathit{estruct}$ algorithm of the respective PRSPD or PRFSPD family measures the state in the computational basis, and outputs the measurement outcome. Note that the $\mathcal{D}\mathit{estruct}$ algorithm, in general, could be more complicated and involve ancillae registers. The proof for the general case is given in Appendix B in the supplementary materials.

Lemma 1 (PRSPD have well-distributed proofs)**.** *For every PRSPD scheme* $(\mathcal{G}\mathit{en}, \mathcal{D}\mathit{estruct}, \mathcal{V}\mathit{er})$ *with key length* $w(\lambda)$ *proof length* $c(\lambda)$*, for every* $a \in \{0,1\}^c$*, there exists a negligible function* $\mathsf{negl}(\lambda)_a$*,*

$$\Pr[K \overset{u}{\leftarrow} \{0,1\}^w : \mathcal{D}\mathit{estruct}(\mathcal{G}\mathit{en}(K)) = a] = \mathsf{negl}(\lambda)_a.$$

Furthermore, there exists a negligible function $\widetilde{\mathsf{negl}(\lambda)}_a$*, such that*

$$\Pr[|\phi\rangle \sim \mu_{\mathcal{H}_n} : \mathcal{D}\mathit{estruct}(|\phi\rangle) = a] = \widetilde{\mathsf{negl}(\lambda)}_a.$$

Proof sketch for the special case. The proof follows by combining pseudorandomness of the PRSPD with the observation that the $\mathcal{D}\mathit{estruct}$ algorithm on a $\mathcal{H}\mathit{aar}$ random state produces a uniformly random outcome. □

The proof for the general case is given in Appendix B on Page 46.

Lemma 2 (PRSPD proofs are distributed close to product distribution)**.** *Let* $\mathcal{PRSPD} = (\mathcal{G}\mathit{en}, \mathcal{D}\mathit{estruct}, \mathcal{V}\mathit{er})$ *be a PRSPD scheme with key length* $w(\lambda)$ *proof length* $c(\lambda)$*. For every* $t \in \mathsf{poly}(\lambda)$*, and* $a_1, \dots, a_t \in \{0,1\}^c$*,*[5]

$$\begin{aligned}&\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{H}\mathit{aar},\mathcal{D}\mathit{estruct}}}[(f_1,\dots,f_t) = (a_1,\dots,a_t)] \\ &\quad\le \frac{N^t}{\binom{N+t-1}{t}} \Pr_{\mathsf{Product\text{-}Destruction}_t^{\mathcal{H}\mathit{aar},\mathcal{D}\mathit{estruct}}}[(f_1,\dots,f_t) = (a_1,\dots,a_t)],\end{aligned}$$

where the subscript in the probability denotes the distribution of $(f_1,\dots,f_t)$*, and the distributions are as defined in Definition 3.*

Proof sketch for the special case. Observe that for any $a_1,\dots,a_t \in \{0,1\}^{c(\lambda)}$, there exists a unique $\vec{z}$ such that $\langle a_1,\dots,a_t|\mathrm{Sym}_t^{\vec{z}}\rangle$ is non-zero. Combining this observation with Eq. (16) and the fact that $\mathcal{D}\mathit{estruct}$ is just a measurement in the computational basis, we conclude that $\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{H}\mathit{aar},\mathcal{D}\mathit{estruct}}}[(f_1,\dots,f_t)$, i.e.,

$$\Pr[|\phi\rangle \sim \mu_{\mathcal{H}_n} : \mathcal{D}\mathit{estruct}^{\otimes t}(|\phi\rangle^{\otimes t}) = (a_1,\dots,a_t)] \ge \frac{1}{\binom{N+t-1}{t}},$$

[5] We believe that the distributions are in fact, statistically close due to the strong concentration of the Haar measure, but we have not been able to prove it. The lemma is a weaker version of this statement, but it suffices for our purposes.

where equality holds when $a_1 = \cdots = a_t$. Moreover, $\Pr_{\mathsf{Product\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}}[(f_1, \ldots, f_t) = (a_1, \ldots, a_t)]$ for every $a_1, \ldots, a_t \in \{0,1\}^{c(\lambda)}$. Hence,

$$\begin{aligned}&\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}}[(f_1, \ldots, f_t) = (a_1, \ldots, a_t)]\\ &\quad \leq \frac{N^t}{\binom{N+t-1}{t}} \Pr_{\mathsf{Product\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}}[(f_1, \ldots, f_t) = (a_1, \ldots, a_t)].\end{aligned}$$

□

The proof for the general case is given in Appendix B on Page 48.

Remark 5. For every $t \in \mathsf{poly}(\lambda)$, $\mathsf{Correlated\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}$ and $\mathsf{Product\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}$ are efficiently samplamable using a state t-design and a state 1-design for n-qubit quantum states, respectively.

Lemma 3 (PRSPD proofs are collision-free)**.** *For every PRSPD scheme $\mathcal{PRSPD} = (\mathscr{Gen}, \mathscr{Destruct}, \mathscr{Ver})$ with key length $w(\lambda)$, proof length $c(\lambda)$, and $t \in \mathsf{poly}(\lambda)$, there exists a negligible function $\widetilde{\mathsf{negl}}(\lambda)$,*

$$\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRSPD}}}[\mathsf{Collision}] \equiv \Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRSPD}}}[\exists i \neq j \mid f_i = f_j] = \widetilde{\mathsf{negl}}(\lambda),$$

where the subscript under the probability is the distribution on $f_1, \ldots, f_t$ and $\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRSPD}}$ is as defined in Definition 4.

Moreover, by the pseudorandomness of PRSPD (see Definition 1) there exists a negligible function $\widetilde{\mathsf{negl}}(\lambda)$ such that

$$\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}}[\mathsf{Collision}] \equiv \Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}}[\exists i \neq j \mid f_i = f_j] = \widetilde{\mathsf{negl}}(\lambda),$$

where $\mathsf{Correlated\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}$ is as defined in Definition 3.

Proof sketch for the special case. The moreover part follows by observing that for any $t \in \mathsf{poly}(n)$, measuring t-copies of a n-qubit $\mathscr{Haar}$ random state, is statistically close up to negligible distance (in n) to the t-fold product of the uniform distribution on $\{0,1\}^n$. Hence, the probability of observing indistinct t-outcomes is negligible. The rest of the proof follows due to the pseudorandomness of the PRSPD. □

The proof for the general case is given in Appendix B on Page 52.

Lemma 4 (PRFSPD proofs are collision-free)**.** *For every PRFSPD scheme $\mathcal{PRFSPD} = (\mathscr{Gen}, \mathscr{Destruct}, \mathscr{Ver})$ with key length $w(\lambda)$, input length $d(\lambda)$, proof length $c(\lambda)$, and $t \in \mathsf{poly}(\lambda)$, and $x \in \{0,1\}^d$ there exists a negligible function $\widetilde{\mathsf{negl}}(\lambda)$,*

$$\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRFSPD},x}}[\mathsf{Collision}_x] \equiv \Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRFSPD},x}}[\exists i \neq j \mid f_i^x = f_j^x] = \widetilde{\mathsf{negl}}(\lambda),$$

where $\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{PRFSPD},x}$ is as defined in Definition 4.

Proof. Given a PRFSPD scheme $\mathcal{PRFSPD}$ with input length $d(\lambda)$, and any fixed input $x \in \{0,1\}^d$, we can construct a PRSPD scheme $\mathcal{PRSPD}_x$ as per Remark 3. Applying Lemma 3 on $\mathcal{PRSPD}_x$, we get the desired result. □

Lemma 5 (Classical unforgeability of PRFSPD)**.** *Let PRFSPD = $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ be a Pseudorandom function-like state generator with proofs of destruction family. Then, for every QPT adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Classical\text{-}Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathit{PRFSPD}} = 1] = \mathsf{negl}(\lambda),$$

where $\mathsf{Classical\text{-}Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathit{PRFSPD}}$ *is defined in Algorithm 3.*

Game 3 $\mathsf{Classical\text{-}Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathsf{PRFSPD}}$

```
Given input $1^\lambda$, Challenger samples $k \leftarrow \{0,1\}^{w(\lambda)}$ uniformly at random.
    Challenger also initializes an empty set $S$.
Initialize an empty set of variables, $S$.
$\mathcal{A}$ gets oracle access to $\mathcal{Destruct}(\mathcal{Gen}(k,\cdot))$, $\mathcal{Ver}(k,\cdot,\cdot)$ as oracle.
for $\mathcal{Destruct}(\mathcal{Gen}(k,\cdot))$ query $x$ made by $\mathcal{A}$ do Add $(x, \sigma_x)$ to $S$, where
    $\sigma_x$ is the response of $\mathcal{Destruct}(\mathcal{Gen}(k,\cdot))$ oracle on input $x$.
end for
$\mathcal{A}$ outputs $x', \sigma'_x$ to the challenger.
Return 1 if $(x', \sigma'_x) \notin S$ and $\mathcal{Ver}(k, x', \sigma'_x) = 1$.
```

Proof sketch. The proof follows by combining Lemma 4 with the Unforgeability-of-proofs property of PRFSPD. □

The proof for the general case is given on Page 53.

3 Construction of PRFSPD from any Post-quantum One-Way Function

In this section, we construct PRFSPD (Definition 2) from post-quantum one-way functions. To be more precise, we build a PRFSPD from post-quantum pseudorandom permutations (Definition 13), and since post-quantum OWFs imply post-quantum PRPs [Zha16], our statement follows. Finally, recall Remark 3 which explains why a PRFSPD with input length $d(\lambda) = \lambda$ implies a PRFSPD with input length $0 \leq d'(\lambda) \leq d(\lambda)$, which also means that it implies a PRSPD.

Theorem 6 (Main Theorem)**.** *Assume there exist post-quantum one-way functions. Then, a PRFSPD scheme (Definition 2) with key length $w(\lambda) = \lambda$, input length $d(\lambda) = \lambda$, output length $n(\lambda) = 5 \cdot \lambda$ and proof length $c(\lambda) = 5 \cdot \lambda$, exists.*

The construction is given in Fig. 2 and Fig. 3. In the construction, as in the main theorem, we define the following lengths as a function of the security parameter: key length $w(\lambda) = \lambda$, input length $d(\lambda) = \lambda$, output length $n(\lambda) = 5 \cdot \lambda$ and proof length $c(\lambda) = 5 \cdot \lambda$. Our only cryptographic ingredient is a post-quantum pseudorandom permutation PRP on 5λ bits (Definition 13).

We next prove the security of the PRFSPD construction. This means two things: That the generated states are pseudorandom and that the classical proofs generated are unforgeable. To this end, we prove our main technical lemma that will easily imply both security aspects. Roughly, the lemma below implies that (1) classical access to the $\mathcal{G}en$ oracle, is computationally indistinguishable from an oracle that outputs truly random quantum states, and (2) for every input $x \in \{0,1\}^d$, generating more proofs of destruction than the number of queries that were made to the $\mathcal{G}en$ oracle for x is *information theoretically impossible.*

Lemma 6 (Main Technical Lemma)**.** *Let T a polynomial and let $\mathcal{A}$ a quantum polynomial-time algorithm that outputs a bit $b \in \{0,1\}$ and has classical oracle access to some arbitrary oracle with inputs in $\{0,1\}^d$, such that for every possible input $x \in \{0,1\}^d$, $\mathcal{A}$ makes either 0 or exactly T queries to the oracle on that input x. Then the following two distributions on b are computationally indistinguishable:*

- D_0 : *Sample $k \leftarrow \{0,1\}^\lambda$ uniformly at random. $\mathcal{A}$ has classical access to $\mathcal{G}en(k,\cdot)$, $\mathcal{V}er(k,\cdot,\cdot)$ (from Fig. 2 and Fig. 3, respectively), and makes either 0 or exactly T queries to each of the possible inputs $x \in \{0,1\}^d$ to $\mathcal{G}en(k,\cdot)$, and outputs a bit b.*

- D_1 : *For every $x \in \{0,1\}^d$, sample a T-sized multi-set of $\{0,1\}^{5\lambda}$: $(a_{x,1}, \cdots, a_{x,T})$, and generate the $T \cdot 5\lambda$-qubit state,*

$$|\pi_x\rangle := \sum_{\sigma \in S_T} |a_{x,\sigma(1)}, \cdots, a_{x,\sigma(T)}\rangle \ ,$$

and for each $x \in \{0,1\}^d$, partition the $T \cdot 5\lambda$-qubit state $|\pi_x\rangle$ into T sub-registers, each of size 5λ. The oracles are defined as follows: The generation oracle $\mathcal{G}en^*(\cdot)$, given $x \in \{0,1\}^d$ for the c-th query (for $c \in [T]$), outputs the c-th sub-register of the state $|\pi_x\rangle$. The proof verification oracle $\mathcal{V}er^*(\cdot,\cdot)$, given query $(x,q) \in \left(\{0,1\}^d \times \{0,1\}^{5\lambda}\right)$, outputs 1 iff $q \in \{a_{x,1}, \cdots, a_{x,T}\}$. $\mathcal{A}$ has classical access to $\mathcal{G}en^*(\cdot)$, $\mathcal{V}er^*(\cdot,\cdot)$, makes either 0 or exactly T queries to $\mathcal{G}en^*(\cdot)$ for every $x \in \{0,1\}^d$, and outputs a bit b.

The proof of the lemma is deferred to Appendix C on Page 55.

Proposition 1 (Security - Pseudorandomness)**.** *The PRFSPD scheme in Fig. 2, Fig. 3 maintains the pseudorandomness property (as in Definition 2).*

$\mathcal{Gen}(k,x)$: For security parameter $\lambda \in \mathbb{N}$, input $k \in \{0,1\}^{w(\lambda)}, x \in \{0,1\}^{d(\lambda)}$, execute the following.

1. Generate the uniform superposition $2^{-\frac{\lambda}{2}} \cdot \sum_{y \in \{0,1\}^\lambda} |y\rangle$ over λ qubits.
2. Apply the classical PRP circuit in superposition, with the superposition as input concatenated with $(x, 0^{3\lambda}) \in \{0,1\}^{4\lambda}$:
$$2^{-\frac{\lambda}{2}} \cdot \sum_{y \in \{0,1\}^\lambda} |y\rangle |x, 0^{3\lambda}\rangle |\mathsf{PRP}_k\left(y, x, 0^{3\lambda}\right)\rangle \ .$$
3. Apply the inverse of the classical PRP circuit in superposition to uncompute the left 5λ qubits and get:
$$2^{-\frac{\lambda}{2}} \cdot \sum_{y \in \{0,1\}^\lambda} |0^{5\lambda}\rangle |\mathsf{PRP}_k\left(y, x, 0^{3\lambda}\right)\rangle \ .$$
4. Apply the following circuit $C : \{0,1\}^{5\lambda} \to \{0,1\}^{5\lambda}$ in superposition: Given input, the circuit C computes $\mathsf{PRP}_k^{-1}(\cdot)$, then flips the rightmost 3λ bits, then applies the permutation $\mathsf{PRP}_k(\cdot)$. One can verify that the state we get is
$$2^{-\frac{\lambda}{2}} \cdot \sum_{y \in \{0,1\}^\lambda} |\mathsf{PRP}_k\left(y, x, 1^{3\lambda}\right)\rangle |\mathsf{PRP}_k\left(y, x, 0^{3\lambda}\right)\rangle \ .$$
5. Apply on the left 5λ qubits the unitary that for every $z \in \{0,1\}^{5\lambda}$ maps $U : |z\rangle \to \omega_{2^{5\lambda}}^{z} \cdot |z\rangle$ (this can be efficiently done with a phase kick-back algorithm, as explained in the proof of Theorem 1 in [JLS18]),
$$2^{-\frac{\lambda}{2}} \cdot \sum_{y \in \{0,1\}^\lambda} \omega_{2^{5\lambda}}^{\mathsf{PRP}_k\left(y,x,1^{3\lambda}\right)} \cdot \tag{3}$$
$$|\mathsf{PRP}_k\left(y, x, 1^{3\lambda}\right)\rangle |\mathsf{PRP}_k\left(y, x, 0^{3\lambda}\right)\rangle \ . \tag{4}$$
6. Apply the circuit C again in order to un-compute the left register and trace the remaining 5λ qubits to obtain the output state:
$$|\psi_k^x\rangle := 2^{-\frac{\lambda}{2}} \cdot \sum_{y \in \{0,1\}^\lambda} \omega_{2^{5\lambda}}^{\mathsf{PRP}_k\left(y,x,1^{3\lambda}\right)} \cdot |\mathsf{PRP}_k\left(y, x, 0^{3\lambda}\right)\rangle \ .$$

Figure 2: The state generation procedure of our Pseudorandom Function-Like States with Proof of Destruction.

Proof. Let $\mathcal{A}$ be a quantum polynomial-time adversary and let T be a polynomial bound on the running time of $\mathcal{A}$. We claim that one can assume

```
$\mathit{Destruct}(|\phi\rangle)$
    1. Measure the state $|\phi\rangle$ in the computational basis, and output the measurement outcome.
$\mathit{Ver}(k, x, q)$
    1. Compute $z := \mathsf{PRP}_k^{-1}(q) \in \{0,1\}^{5\lambda}$.
    2. Denote the bits of $z$ as $(z_1, z_2, \cdots, z_{5\lambda})$. Output 1 iff $(z_{\lambda+1}, \cdots, z_{5\lambda}) = \left(x||0^{3\lambda}\right)$.
```

Figure 3: The state destruction and classical proof verification procedures of our Pseudorandom Function-Like States with Proof of Destruction.

without the loss of generality that for every possible input $x \in \{0,1\}^d$, $\mathcal{A}$ makes either 0 or exactly T queries to the generation oracle $\mathit{Gen}(\cdot)$ oracle. The reason is as follows: We can think of a new adversary $\mathcal{A}'$ that, at the end of the execution of $\mathcal{A}$, takes the inputs in $\{0,1\}^d$ that $\mathcal{A}$ queried during its execution $x_1, \cdots, x_t \in \{0,1\}^d$ (for $t \in [T]$) without considering multiplicity (i.e., some of the inputs were possibly queried more than others) and for each of these inputs, complementing the number of times that it was queried (which, as we know is bounded by T by the fact that T is an upper bound on the total running time of $\mathcal{A}$) to be T - such adversary still breaks the pseudorandomness security guarantee and also satisfies the property that for every possible input in $\{0,1\}^d$, the input was queried either 0 or T times. Now, Lemma 6 *in particular* says that the output of $\mathcal{A}$ on the classical access to the generation function $\mathit{Gen}(k, \cdot)$ (which is part of D_0 in the lemma's statement) is computationally indistinguishable from the output of $\mathcal{A}$ when the access is to the generation function $\mathit{Gen}^*(\cdot)$ defined in the distribution D_1, in the statement of the Lemma 6.

One of the standard facts in quantum information theory is that the distribution generated by T copies of a truly random, 5λ-qubit Haar state is statistically equivalent (i.e. has trace distance 0) to the projection onto the $(5\lambda, T)$-symmetric subspace ([Har13, Prop. 6]). In turn, an orthonormal basis for the $(5\lambda, T)$-symmetric subspace is given by the set of states defined by all T-sized multi-sets of $\{0,1\}^{5\lambda}$: For each T-sized multi-set M of $\{0,1\}^{5\lambda}$, the corresponding state $|\psi_M\rangle$ is a $5\lambda \cdot T$-qubit state which is the uniform superposition over all of the possible permutations of the T elements of M (e.g. in case M is not only a multi-set but an actual T-sized set, and all of its elements are distinct, the number of such permutations is $T!$, and if M is T times the same element, the number of such permutations is 1).

It follows that the projection onto the $(5\lambda, T)$-symmetric subspace is exactly the mixed state that corresponds to the distribution induced by sampling a T-sized multi-set M of $\{0,1\}^{5\lambda}$ and outputting the quantum state $|\psi_M\rangle$. To conclude, for T queries, the oracle access $\mathit{Gen}^*(\cdot)$ is equivalent to the oracle that outputs T copies of a 5λ-qubit Haar random state, and since the output bit of $\mathcal{A}$ is indistinguishable between $\mathcal{Gen}_k(\cdot)$ and $\mathcal{Gen}^*(\cdot)$, then the output bit of $\mathcal{A}$ is also indistinguishable between $\mathcal{Gen}_k(\cdot)$ and an oracle that outputs Haar random states, in contradiction to the assumption that $\mathcal{A}$ breaks the pseudorandomness guarantee. □

Proposition 2 (Security - Unforgeability of Proofs). *The PRFSPD scheme in Fig. 2, Fig. 3 maintains the Unforgeability-of-proofs property (as in Definition 2).*

Proof. Assume toward contradiction there exists a quantum polynomial-time adversary $\mathcal{A}$ that breaks the proof-unforgeability property of the scheme, let ε the probability that the adversary obtains in winning the forging game (i.e. ε is non-negligible). If T is a polynomial bound on the running time of $\mathcal{A}$, note that we can assume without the loss of generality that for every possible input $x \in \{0,1\}^d$, $\mathcal{A}$ makes either 0 or exactly T queries to the generation oracle $\mathcal{Gen}_k(\cdot)$ and arbitrarily many queries to the verification oracle $\mathcal{Ver}_k(\cdot,\cdot)$ (in the boundaries of its running time). The reason this can be assumed w.l.o.g. is because we can consider a new adversary $\mathcal{A}'$ that uses $\mathcal{A}$ and (similarly to how we defined $\mathcal{A}'$ as a function of $\mathcal{A}$ in the proof of 1) complements the number of queries for each of its $t \in [T]$ previously-queried x's to being queried T times. However, this argument is a bit more delicate when it comes to showing how the new adversary $\mathcal{A}'$ breaks the proof-unforgeability property: At the end of the execution of the inner adversary $\mathcal{A}$, it outputs $x, p_1, \cdots, p_{t_x+1}$ (where t_x is the number of times that the input $x \in \{0,1\}^d$ was queried to $\mathcal{Gen}_k(\cdot)$ by the inner adversary $\mathcal{A}$) such that $\forall i \in [t_x + 1] : \mathcal{Ver}(k, x, p_i) = 1$. The outer adversary $\mathcal{A}'$ then takes the extra $\ell := T - t_x$ queries that it made (these are the queries it made in order to complement the number of queries for x from t_x to T, as part of the transformation of the inner adversary $\mathcal{A}$ to the outer adversary $\mathcal{A}'$) for the input string x and measures the ℓ copies it got in the computational basis, to obtain ℓ valid classical proofs of destruction for x.

Note that each of the ℓ copies is a uniform superposition over a set of size 2^λ, which means that the probability that any of the ℓ (which is a polynomial because T is a polynomial) proofs collides with the $t_x + 1$ proofs generated by the cheating inner adversary $\mathcal{A}$, is negligible. Thus, the new outer adversary $\mathcal{A}'$ makes a total of T queries on the input $x \in \{0,1\}^d$ but manages to generate $T + 1$ distinct classical proofs of destruction, which means it breaks the security with a non-negligible probability ε'. Finally, one can think of an even outer process $\mathcal{A}^*$, that uses $\mathcal{A}'$ to generate the $T+1$ proofs, then checks by itself their validity using the verification algorithm $\mathcal{Ver}(\cdot,\cdot)$, and outputs a bit whether or not the adversary $\mathcal{A}'$ won the forging game. Note that because $\mathcal{A}'$ wins the forging game with the non-negligible probability ε', then with the same probability, the output bit of $\mathcal{A}^*$ is 1.

Finally, by Lemma 6, it follows that the output bit of $\mathcal{A}^*$ in the above process is computationally indistinguishable from its output bit in the setting

D_1 defined in Lemma 6, where $\mathcal{A}^*$ gets access only to $\mathcal{Gen}^*(\cdot)$ and $\mathcal{Ver}^*(\cdot,\cdot)$ (rather than $\mathcal{Gen}_k(\cdot)$ and $\mathcal{Ver}_k(\cdot,\cdot)$). Now, given the access to the two oracles $\mathcal{Gen}^*(\cdot)$ and $\mathcal{Ver}^*(\cdot,\cdot)$ (i.e., in the setting of D_1), for any algorithm, even unbounded, the probability to output $T+1$ distinct strings that are all verified by the algorithm $\mathcal{Ver}^*(x,\cdot)$ for some $x \in \{0,1\}^d$ is zero, as by its definition, accepts at most T different elements. Our contradiction follows from the fact that ε' (the probability for $\mathcal{A}^*$ to output 1 in the setting D_0) is non-negligible, but has to be negligibly close to 0 (the probability for $\mathcal{A}^*$ to output 1 in the setting D_1, where it has access only to the oracles $\mathcal{Gen}^*(\cdot)$ and $\mathcal{Ver}^*(\cdot,\cdot)$). □

4 Applications: Cryptography with Classical Communication

In this section, all the constructions of the cryptographic primitives are fully black-box constructions with uniform security reductions [RTV04] from either PRSPD or PRFSPD, except for the construction of the statistically binding and computationally hiding bit-commitment scheme in Section 4.3, which is a fully black-box (with uniform reduction) construction from the particular class of PRSPD that satisfies Definitions 8 and 9 with suitable parameters. Therefore, the security guarantees of all these primitives hold even against non-uniform adversaries with quantum advice, assuming the same notion of security for the underlying PRSPD (or a special form of it) and PRFSPD. For simplicity, we only consider uniform adversaries from here onwards. Moreover, the outputs of all the algorithms in these constructions should be considered classical unless explicitly specified otherwise. Due to space constraints, some of the results and proofs have been moved to the Supplementary materials, see Section 4.5 and Appendix E.

None of the cryptographic primitives considered in the applications can exist unconditionally; see Appendix F for more details.

4.1 One-Time Signatures

Definition 5 (One-Time-Signature Adapted from [MY22a, Section 4.2])**.** *A* One-Time-Signature *scheme (*OTS*) is a triplet of QPT algorithms* $(\mathcal{Keygen}, \mathcal{Sign}, \mathcal{Verify})$ *with the following syntax:*

- $(sk, pk) \leftarrow \mathcal{Keygen}(1^\lambda)$*: samples a classical secret key* sk *and a classical public key* pk*.*
- $\mathsf{sig} \leftarrow \mathcal{Sign}(sk, m)$*: takes a secret key* sk*, a classical message* $m \in \{0,1\}^{\ell(\lambda)}$*, and outputs a classical signature* sig*.*
- $b \leftarrow \mathcal{Verify}(pk, m, \mathsf{sig})$*: takes a public key* pk*, a classical message* m*,*

signature string sig*, and outputs a boolean value, either accept* $(b = 1)$ *or reject* $(b = 0)$*.*

Statistical Correctness *For every message* $m \in \{0,1\}^{\ell(\lambda)}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$*, (also called the correctness precision) such that*

$$\Pr[sk, pk \leftarrow \mathcal{Keygen}(1^\lambda); \mathsf{sig} \leftarrow \mathcal{Sign}(sk, m) : \mathcal{Verify}(pk, \mathsf{sig}) = 1] \geq 1 - \mathsf{negl}(\lambda).$$

One-time Unforgeability *For every QPT adversary* $\mathcal{A}$ *in forging game (see Game 4), there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathsf{OTS}} = 1] = \mathsf{negl}(\lambda).$$

Game 4 $\mathsf{Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathsf{OTS}}$

```
1: Given input $1^\lambda$, Challenger samples $(sk, pk) \leftarrow \mathcal{Keygen}(1^\lambda)$ and gives $pk$ to the adversary.
2: $\mathcal{A}$ sends a message $m \in \{0,1\}^{\ell(\lambda)}$ to the challenger.
3: Challenger runs $\mathsf{sig} \leftarrow \mathcal{Sign}(sk, m)$ and sends $\mathsf{sig}$ to $\mathcal{A}$.
4: $\mathcal{A}$ outputs $(\widetilde{m}, \widetilde{\mathsf{sig}})$ to the challenger.
5: Challenger rejects if $\widetilde{m} = m$.
6: Return $b \leftarrow \mathcal{Verify}(pk, \widetilde{m}, \widetilde{\mathsf{sig}})$.
```

Remark 7 (Length-restriction in Definition 5)*.* Definition 5, as well as the definition in [MY22a], are length-restricted, i.e., we can only sign messages of a fixed length $\ell(\lambda)$. This is because the respective constructions can only sign fixed-length messages[6], and the known ways to generically transform a length-restricted one-time signature, to an unrestricted one-time signature that can sign messages of any length requires a Universal One-way Hash Function (UOWHFs), which seems like a stronger primitive than PRSPD or PRFSPD.

4.1.1 Construction from PRSPD

Next, we construct a One-Time-Signature scheme OTS from a PRSPD scheme $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$, with key-length $w(\lambda)$, and proof length $c(\lambda)$, see Figs. 2 and 3. For simplicity, we consider the message length $\ell(\lambda) = 1$, i.e., our construction can only sign single-bit messages. However, by simple repetition, we can extend this scheme to a scheme with message length $\ell(\lambda)$, for any arbitrary fixed polynomial $\ell(\lambda)$.

[6] The same holds for the One-Time-Signature construction in [MY22a] with quantum public-keys.

The naive approach would be to follow the template mentioned in the introduction (on Page 6), and transform the one-time digital signature with *quantum* public keys construction in [MY22a], to a one-time digital signature with classical public keys, by replacing the pseudorandom state that was used as the public key with a proof of destruction for it. However, the scheme thus obtained may not be secure for arbitrary PRSPD. We can add a dummy key to the PRSPD family that accepts all proofs of destruction on verification. Since we are only adding a single key, the security properties of the PRSPD would remain intact, but now the one-time signature scheme can be easily forged since the dummy key acts as a valid signature for both 0 and 1 no matter what the public keys are. We solve this issue by adding a check in the signature verification (see Item 1 in Fig. 4), that ensures that the signature string does not accept proofs of destruction that were sampled independently, and hence the attack with the dummy key does not work anymore. A typical key sampled during $\mathcal{Keygen}$ would have this property due to the Unforgeability-of-proofs property of the underlying PRSPD, hence statistical correctness holds (see Theorem 8).

Assumes: PRSPD scheme, $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$

$\mathcal{Keygen}(1^\lambda)$

1. Sample $k_0, k_1 \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}$.
2. For each $i \in \{0,1\}$, generate $\{q_j^i\}_{j\in[w]}$ where $q_j^i \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k_i))$ for every j independently.
3. Output $sk = (k_0, k_1)$ and $pk = \Big(\{q_j^0\}_{j\in[w]}, \{q_j^1\}_{j\in[w]}\Big)$.

$\mathcal{Sign}(sk, m)$

Interprete $sk = (k_0, k_1)$. Output k_m.

$\mathcal{Verify}(pk, m, \mathsf{sig})$

Run the following steps and accept if both pass.

1. Sample $k_1, \ldots, k_{w^2} \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}$, and for every $j \in [w^2]^7$, generate $r_j \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k_j))$. Run $\mathcal{Ver}(\mathsf{sig}, r_j)$ for every j and accept if all j verifications fail.
2. Interprete $pk = \Big(\{q_j^0\}_{j\in[w]}, \{q_j^1\}_{j\in[w]}\Big)$. Run $\mathcal{Ver}(\mathsf{sig}, q_j^m)$ or every j and accept if all j verifications pass.

Figure 4: One-time Signature scheme OTS.

Theorem 8 (Statistical correctness of OTS)**.** *The* One-Time-Signature OTS *is statistically correct (see Definition 5) if* $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ *satisfies correctness and* Unforgeability-of-proofs *(see Definition 1).*

Proof sketch of Theorem 8. Fix a message $m \in \{0,1\}$ arbitrarily. Let $\mathsf{sig} \leftarrow \mathcal{Sign}(sk, m)$ where $(sk, pk) \leftarrow \mathcal{Keygen}(1^\lambda)$. Then, Item 2 of $\mathcal{Verify}(pk, m, \mathsf{sig})$

would pass with certainty due to the correctness $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$, and Item 2 would pass with overwhelming probability due to the Unforgeability-of-proofs of $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$. Hence, $\mathcal{Verify}(pk, m, \mathsf{sig})$ would pass with an overwhelming probability. □

The full proof is given in Appendix E.1 on Page 63.

Theorem 9 (One-time Unforgeability of OTS)**.** *The* One-Time-Signature OTS *is* one-time unforgeable *if $\mathcal{PRSPD} = (\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Verify})$ is a PRSPD (see Definition 1).*

Proof. By a standard Lamport signature [Lam79] argument (also done in the security proof of One-Time-Signature with quantum public keys from PRS in [MY22a, Theorem 4.1]), any $\mathcal{A}$ in Game 4 for the scheme OTS, can be reduced to an adversary $\mathcal{B}$ in the game $\mathsf{Inverting\text{-}Exp}_{\lambda}^{\mathcal{B},\mathsf{PRSPD}}$ (see Game 5) for the underlying PRSPD scheme $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$, such that $\mathcal{B}$ wins with probability at least half the success probability of $\mathcal{A}$ (see [MY22a, Proof of Theorem 4.1]).

Game 5 $\mathsf{Inverting\text{-}Exp}_{\lambda}^{\mathcal{B},\mathcal{PRSPD}}$

```
Given input $1^\lambda$, Challenger samples $k \xleftarrow{u} \{0,1\}^{w(\lambda)}$.
Challenger gives $q_1, \ldots, q_w$ to the adversary, where $q_j \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k))$, for each $j \in [w]$.
$\mathcal{A}$ sends an alleged key $s$ to the challenger.
Challenger sets $a = 1$, $b = 1$.
for $j \in [w^2]$ do
    Challenger runs $k_j \xleftarrow{u} \{0,1\}^{w(\lambda)}$, $r_j \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k_j))$, and $a_j \leftarrow 1 - \mathcal{Ver}(s, r_j)$.
    Set $a \leftarrow a \cdot a_j$.
end for
for $j \in [w]$ do
    Challenger runs $b_j \leftarrow \mathcal{Ver}(s, q_j)$.
    Set $b \leftarrow b \cdot b_j$.
end for
Output $a \wedge b$.
```

Suppose $\Pr[\mathsf{Inverting\text{-}Exp}_{\lambda}^{\mathcal{B},\mathcal{PRSPD}} = 1] = p$. For every $t \in \mathsf{poly}(\lambda)$, recall that $\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}$, $\mathsf{Product\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}$ and $\mathsf{Product\text{-}Destruction}_t^{\mathcal{PRSPD}}$ be distributions on $\{0,1\}^{ct}$ as defined in Definitions 3 and 4, i.e.,

$$(f_1, \ldots, f_t) \sim \mathsf{Correlated\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}} \equiv (f_1, \ldots, f_t) \leftarrow \mathcal{Destruct}^{\otimes t}(|\psi\rangle); |\psi\rangle \sim \mu_{\mathcal{H}_n},$$

and $\mathsf{Product\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}$, $\mathsf{Product\text{-}Destruction}_t^{\mathcal{PRSPD}}$ are the t-fold product of $\mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}$ and $\mathsf{Product\text{-}Destruction}^{\mathcal{PRSPD}}$ such that $f \sim \mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}$ and $g \sim \mathsf{Product\text{-}Destruction}^{\mathcal{PRSPD}}$ are sampled as

$$f \leftarrow \mathcal{Destruct}(|\phi\rangle); |\phi\rangle \sim \mu_{\mathcal{H}_n}; g \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k)); k \xleftarrow{u} \{0,1\}^w,$$

respectively. By the pseudorandomness (see Definition 1) of the underlying PRSPD,

$$\mathsf{Product\text{-}Destruction}^{\mathcal{PRSPD}} \approx_c \mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}.$$

Hence, by standard hybrid arguments, for any $t \in \mathsf{poly}(\lambda)$, their t-fold products are computationally indistinguishable, i.e.,

$$\mathsf{Product\text{-}Destruction}_t^{\mathcal{PRSPD}} \approx_c \mathsf{Product\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}} \tag{5}$$

We consider the following hybrids $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2$ and let p_0, p_1, p_2 be the respective success probabilities of $\mathcal{B}$.

- $\mathcal{H}_0$: Same as $\mathsf{Inverting\text{-}Exp}_\lambda^{\mathcal{B},\mathcal{PRSPD}}$. Hence, $p_0 = p$.
- $\mathcal{H}_1$: Line 2 in Game 5 is changed as follows:
  ~~Challenger gives $q_1, \dots, q_w$ to the adversary, where $q_j \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k))$, for each $j \in [w]$.~~
  Challenger samples $|\phi\rangle \sim \mu_{\mathcal{H}_n}$, and gives $q_1, \dots, q_w$ to the adversary, where $q_j \leftarrow \mathcal{Destruct}(|\phi\rangle)$, for each $j \in [w]$.

  By the pseudorandomness guarantee (see Definition 1) of the underlying PRSPD, there exists a negligible function $\mathsf{negl}(\lambda)$ such that,

  $$|p_1 - p_0| \le \mathsf{negl}(\lambda).$$

- $\mathcal{H}_2$: Line 6 in Game 5 is changed as follows:
  ~~Challenger runs $k_j \xleftarrow{u} \{0,1\}^{w(\lambda)}$, $r_j \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k_j))$, and $a_j \leftarrow 1 - \mathcal{Ver}(s, r_j)$.~~
  Challenger runs $|\phi_j\rangle \sim \mu_{\mathcal{H}_n}$, $r_j \leftarrow \mathcal{Destruct}(|\phi_j\rangle)$, and $a_j \leftarrow 1 - \mathcal{Ver}(s, r_j)$.

  In other words, we change the distribution on $(r_1, \dots, r_{w^2})$ from $\mathsf{Product\text{-}Destruction}_{w^2}^{\mathcal{PRSPD}}$ to $\mathsf{Product\text{-}Destruction}_{w^2}^{\mathcal{Haar},\mathcal{Destruct}}$. Since $(w(\lambda))^2 \in \mathsf{poly}(\lambda)$, by Eq. (5),

  $$\mathsf{Product\text{-}Destruction}_{w^2}^{\mathcal{PRSPD}} \approx_c \mathsf{Product\text{-}Destruction}_{w^2}^{\mathcal{Haar},\mathcal{Destruct}}. \tag{6}$$

  Given $(r_1, \dots, r_{w^2})$ from the respective distribution, the rest of $\mathcal{H}_1$ and $\mathcal{H}_2$ are identical and efficiently simulatable (by Remark 5), hence, by Eq. (6), there exists a negligible function $\mathsf{negl}(\lambda)$ such that, $|p_2 - p_1| \le \mathsf{negl}(\lambda)$.

Therefore,

$$|p_2 - p_0| \leq |p_2 - p_1| + |p_1 - p_0| = \mathsf{negl}(\lambda) + \widetilde{\mathsf{negl}(\lambda)},$$

which is negligible. Hence, it suffices to prove that p_2 is negligible. Let S denote the alleged key that $\mathcal{B}$ responds to in $\mathcal{H}_2$. Let $\mathsf{Success}_{\geq \frac{1}{w}}, \mathsf{Success}_{< \frac{1}{w}} \subset \{0,1\}^w$ be the set of keys defined as

$$\mathsf{Success}_{\geq \frac{1}{w}} \equiv \left\{ s \in \{0,1\}^w \mid \Pr_{f \sim \mathsf{Product\text{-}Destruction}^{\mathcal{H}aar, \mathcal{D}estruct}} [\mathcal{V}er(s,f)] \geq \frac{1}{w} \right\}, \tag{7}$$

$$\mathsf{Success}_{< \frac{1}{w}} \equiv \{0,1\}^w \setminus \mathsf{Success}_{\geq \frac{1}{w}}. \tag{8}$$

For every $s \in \{0,1\}^w, d \in \{0,1\}^c$, let $\mathsf{Success}(s,d)$ and $\mathsf{Fail}(s,d)$ be boolean random variables indicating $\mathcal{V}er(s,d) = 1$ and $\mathcal{V}er(s,d) = 0$, respectively. We next use the following lemma that we prove in Appendix E.1 on Page 65, to complete the proof.

Lemma 7. *There exists negligible functions, $\epsilon(\lambda)$ and $\delta(\lambda)$ such that,*

$$\Pr\left[\left(\wedge_{j \in [w]} \mathsf{Success}(S, q_j) \right) \wedge S \in \mathsf{Success}_{< \frac{1}{w}} \right] \leq \epsilon(\lambda),$$

$$\Pr\left[\left(\wedge_{j \in [w^2]} \mathsf{Fail}(S, r_j) \right) = 1 \wedge S \in \mathsf{Success}_{\geq \frac{1}{w}} \right] \leq \delta(\lambda).$$

Note that, $p_2 = \Pr\left[\left(\wedge_{j \in [w]} \mathsf{Success}(S, q_j) \right) \wedge \left(\wedge_{j \in [w^2]} \mathsf{Fail}(S, r_j) \right) = 1 \right]$. Since $\mathsf{Success}_{< \frac{1}{w}}$, $\mathsf{Success}_{\geq \frac{1}{w}}$ partitions the keyspace, p_2 is at most

$$\Pr\left[\left(\wedge_{j \in [w]} \mathsf{Success}(S, q_j) \right) \wedge S \in \mathsf{Success}_{< \frac{1}{w}} \right] + \Pr\left[\left(\wedge_{j \in [w^2]} \mathsf{Fail}(S, r_j) \right) = 1 \wedge S \in \mathsf{Success}_{\geq \frac{1}{w}} \right],$$

which is negligible by Lemma 7. □

Remark 10 (Extending to q-times security). For any fixed $q(\lambda) \in \mathsf{poly}(\lambda)$, a $\mathsf{One\text{-}Time\text{-}Signature}$ (Definition 5) OTS can be extended in a black-box manner, to a digital signature scheme OTS_q which is q-times secure, meaning $\mathcal{A}$ cannot forge in Game 4 even if she has access to q-signing (adaptive) queries instead of just one. The proof follows by adapting the proof for the case of digital signatures with *quantum* public key, given in [MY22b, Theorem 4.2.].

4.2 Non-interactive classically verifiable private quantum coins

A private quantum coin scheme with non-interactive classical verification has its own advantages and disadvantages compared to a vanilla quantum coin scheme. The obvious advantage is that it saves communication costs during verification, but this comes at a price: the bank cannot have multiple branches since it must have a stateful memory to ensure that the same classical certificate is not used multiple times to pass verification; otherwise, unforgeability is meaningless.

Definition 6. *A* Non-interactive Classically-Verifiable private private quantum coin *scheme (*NCV-Coin*) is a quadruple of QPT algorithms,* $(\mathcal{Keygen}, \mathcal{Mint}, \mathcal{Cert\text{-}Gen}, \mathcal{Cert\text{-}Verify})$ *and has the following syntax:*

- $k \leftarrow \mathcal{Keygen}(1^\lambda)$*: takes as input the security parameter and samples a classical key* k*, where* $w(\lambda)$ *is the key length.*
- $|\psi_k\rangle \leftarrow \mathcal{Mint}(k)$*: takes a key* $k \in \{0,1\}^w$ *and outputs an* n*-qubit pure state,* $|\psi_k\rangle$ *as the quantum coin state.*
- $\mathsf{cert} \leftarrow \mathcal{Cert\text{-}Gen}(|\psi\rangle)$*: takes an* n*-qubit quantum state* $|\psi\rangle$ *and outputs a classical certificate string* $\mathsf{cert} \in \{0,1\}^c$*.*
- $b \leftarrow \mathcal{Cert\text{-}Verify}(k, \mathsf{cert})$*: takes a key* k *and a certificate* $\mathsf{cert} \in \{0,1\}^c$ *and outputs a boolean value, either accept (*$b = 1$*) or reject (*$b = 0$*).*

Statistical Correctness

1. $\Pr[k \leftarrow \mathcal{Keygen}(1^\lambda); |\psi_k\rangle \leftarrow \mathcal{Mint}(k); \mathsf{cert} \leftarrow \mathcal{Cert\text{-}Gen}(|\psi_k\rangle) : \mathcal{Cert\text{-}Verify}(k, \mathsf{cert}) = 1] = 1.$
2. *Moreover, for every* $q(\lambda) \in \mathsf{poly}(\lambda)$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that,*

   $$\Pr[\exists i \neq j, \mathsf{cert}_i = \mathsf{cert}_j] \leq \mathsf{negl}(\lambda),$$

   where $\mathsf{cert}_1, \ldots, \mathsf{cert}_q$ *are defined according to the following process:* $k \leftarrow \mathcal{Keygen}(1^\lambda); |\psi_k\rangle \leftarrow \mathcal{Mint}(k); \mathsf{cert}_1, \ldots, \mathsf{cert}_q \leftarrow \mathcal{Cert\text{-}Gen}^{\otimes q}(|\psi_k\rangle^{\otimes q})$*.*

   i.e., an honest user who gets q *quantum coins must get* q *distinct certificates on destructing them, with overwhelming probability.*

Adaptive Unforgeability *For every QPT adversary* $\mathcal{A}$*, in* $\mathsf{Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathsf{NCV\text{-}Coin}}$ *(see Game 6), there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that,*

$$\Pr[\mathsf{Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathsf{NCV\text{-}Coin}}] = \mathsf{negl}(\lambda).$$

[8] This captures the fact that while the adversary can approach the bank several times for verification, the bank only accepts classical certificates for verification.

[9] Without this check, the forging game can be won trivially: take one coin from the $\mathcal{Mint}$, destruct the coin to get a certificate cert, and submit two copies of cert. By statistical correctness, both verifications will pass with overwhelming probability.

Game 6 $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{NCV\text{-}Coin}}$

```
1: Given input $1^\lambda$, Challenger samples $k \leftarrow \mathit{Keygen}(\lambda)$.
2: $\mathcal{A}$ sends $m \in \mathbb{N}$ to the challenger.
3: Challenger runs $|\psi_k\rangle^{\otimes m} \leftarrow \mathit{Mint}(k)^{\otimes m}$ and sends $|\psi_k\rangle^{\otimes m}$ to $\mathcal{A}$.
4: $\mathcal{A}$ gets classical oracle access[8] to $\mathit{Cert\text{-}Verify}(k, \cdot)$ as an oracle.
5: $\mathcal{A}$ outputs $\mathsf{cert}_1, \mathsf{cert}_2, \ldots, \mathsf{cert}_{m+1}$ to the challenger.
6: Challenger checks if $\mathsf{cert}_1, \mathsf{cert}_2, \ldots, \mathsf{cert}_{m+1}$ are distinct, and if not, rejects.[9]
7: for $i \in [m+1]$ do Challenger computes $b_i \leftarrow \mathit{Cert\text{-}Verify}(k, \mathsf{cert}_i)$
8: end for
9: Return $\wedge_{i=1}^{m+1} b_i$.
```

4.2.1 Construction from PRSPD

In this section, we construct a Non-interactive classically-verifiable private quantum coins scheme NCV-Coin from a PRFSPD scheme $(\mathit{Gen}, \mathit{Destruct}, \mathit{Ver})$.

Assumes: PRSPD scheme, $(\mathit{Gen}, \mathit{Destruct}, \mathit{Ver})$
$\mathit{Keygen}(1^\lambda)$: Output $k \overset{u}{\leftarrow} \{0,1\}^w$.
$\mathit{Mint}(k)$: Output $|\psi_k\rangle \leftarrow \mathit{Gen}(k)$.
$\mathit{Cert\text{-}Gen}(|\psi\rangle)$: Output $\mathsf{cert} \leftarrow \mathit{Destruct}(|\psi\rangle)$.
$\mathit{Verify}(k, \mathsf{cert})$: Output $b \leftarrow \mathit{Ver}(k, \mathsf{cert})$.

Figure 5: NCV-Coin.

Remark 11. Any classically-verifiable private quantum coins scheme is forgeable without the check in Line 6 in $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{NCV\text{-}Coin}}$ because we can use one classical certificate obtained by destructing one money state to pass verification multiple times. Due to this check, the bank, in the real-world scenario, needs to be stateful and cannot have multiple branches that can verify the user's money.

Interestingly, we can add a quantum algorithm $\mathit{State\text{-}Verify}$ to our construction in Fig. 5 that directly verifies the money state. $\mathit{State\text{-}Verify}$ is the same as the verification algorithm in quantum coins construction from PRS in [JLS18], i.e., it applies a projective measurement that accepts the alleged input state ρ with probability $\langle\phi_k|\rho|\phi_k\rangle$, where $|\phi_k\rangle$ is the true coin state. Since every PRSPD is a PRS, this construction with $\mathit{State\text{-}Verify}$ algorithm, as a vanilla quantum coins scheme, is the same as the construction in [JLS18] based on PRS and hence is unforgeable in the vanilla quantum money sense. Therefore to conclude, in our quantum coins scheme, a user can choose two ways to verify her coin a) classical verification mode, where she sends the proof of possession to the main bank for verification using classical communication. Note that here the bank cannot have multiple non-communicating branches. b) quantum communication mode, where the user sends the quantum coin itself for verification and hence needs quantum communication. Here the bank can have multiple non-communicating branches to verify these quantum states.

Theorem 12 (Statistical Correctness of NCV-Coin)**.** *The* Non-interactive classically-verifiable private quantum coins *scheme* NCV-Coin *is statistically correct (see Definition 6) if* $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ *satisfies correctness and* Unforgeability-of-proofs *(see Definition 1).*

Proof. Item 1 in the correctness definition (see Definition 6) follows directly from the correctness of $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$, and Item 2 follows from *Lemma* 4 for $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$.

□

Theorem 13 (Adaptive Unforgeability of NCV-Coin)**.** *The* Non-interactive classically-verifiable private quantum coins NCV-Coin *is* adaptively unforgeable *if* $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Verify})$ *is a PRSPD (see Definition 1).*

Proof. The forging game $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{NCV\text{-}Coin}}$ given in Game 6 is exactly the cloning game $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathcal{PRSPD}}$ given in Game 1), where $\mathcal{PRSPD} = (\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ once $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{NCV\text{-}Coin}}$ is rewritten in terms of the underlying PRSPD. Hence, the result follows directly from the Unforgeability-of-proofs of $\mathcal{PRSPD}$. □

4.3 Statistically binding commitments with classical communication

Definition 7 (Statistically-binding and computationally-hiding bit commitments [AGQY22])**.** *A* statistically-binding computational-hiding bit-commitment *($\mathcal{BC}$) is a pair of interactive QPT protocols $(\mathcal{C}_\lambda, \mathcal{R}_\lambda)_{\lambda\in\mathbb{N}}$ with two phases Commit and Reveal with the following syntax:*

- $\sigma_{\mathcal{C},\mathcal{R}} \leftarrow \mathit{Commit}(\mathcal{C}_\lambda(b), \mathcal{R}_\lambda)$. *In this phase, $\mathcal{C}$ takes a bit b as input. $\mathcal{C}$ engages in a classical interactive protocol with the $\mathcal{R}$, at the end of which they output the committed state $\sigma_{\mathcal{C},\mathcal{R}}$ on the committer and receiver registers $\mathcal{C}$ and $\mathcal{R}$ respectively.*
- $b \cup \{\perp\} \leftarrow \mathit{Reveal}(\mathcal{C}_\lambda, \mathcal{R}_\lambda, \sigma_{\mathcal{C},\mathcal{R}})$*: $\mathcal{C}$ and $\mathcal{R}$ take a committed state $\sigma_{\mathcal{C},\mathcal{R}}$ and runs a classical interactive protocol at the end of which $\mathcal{R}$ output a bit b.*

Correctness *For every $b \in \{0,1\}$,*

$$\Pr[\sigma_{\mathcal{C},\mathcal{R}} \leftarrow \mathit{Commit}(\mathcal{C}_\lambda(b), \mathcal{R}_\lambda); b' \leftarrow \mathit{Reveal}(\mathcal{C}_\lambda, \mathcal{R}_\lambda, \sigma_{\mathcal{C},\mathcal{R}}) : b = b'] = 1$$

Computational Hiding *For every malicious* QPT *receiver* $\{\mathcal{R}_\lambda^*\}_\lambda$ *and for every distinguisher* $\{\mathcal{D}_\lambda\}_\lambda$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left| \Pr_{\sigma_{\mathcal{C},\mathcal{R}^*}\, Commit(\mathcal{C}_\lambda(0),\mathcal{R}_\lambda^*)}[\mathcal{D}_\lambda(\sigma_{\mathcal{R}^*}) = 1] - \Pr_{\sigma_{\mathcal{C},\mathcal{R}^*}\, Commit(\mathcal{C}_\lambda(1),\mathcal{R}_\lambda^*)}[\mathcal{D}_\lambda(\sigma_{\mathcal{R}^*}) = 1] \right| \leq \mathsf{negl}(\lambda).$$

Statistical Binding *For every malicious* QPT *committer* $\{\mathcal{C}_\lambda^*\}_\lambda$*, there exists a (possibly inefficient) extractor* $\mathcal{E}$ *(that outputs either a bit or* $\perp$*) and a negligible function* $\epsilon(\lambda)$ *such that*

$$\Pr[\mu \neq b^* \wedge \mu \neq \perp \mid (\tau, \sigma_{\mathcal{C}^*,\mathcal{R}}) \leftarrow Commit(\mathcal{C}_\lambda^*, \mathcal{R}_\lambda), b^* \leftarrow \mathcal{E}(\tau), \mu \leftarrow Reveal(\mathcal{C}^*, \mathcal{R}, \sigma_{\mathcal{C}^*,\mathcal{R}})] \leq \epsilon(\lambda).$$

4.3.1 PRSNPD

Our commitment scheme requires a PRSPD in which the proof of destruction satisfies some additional properties, which we call PRSNPD. We do not manage to generically prove that any PRSPD implies a PRSNPD. We do manage to show that our construction of PRSPD from Section 3 is not only a PRSPD, but a PRSNPD.

Definition 8 (PRSPD with bounded proofs)**.** *A PRSPD with* $(Q(\lambda), M(\lambda))-$ *bounded proofs is a PRSPD scheme satisfying the additional property that for every* $k \in \{0,1\}^{w(\lambda)}$*,*

$$\left|\{p \in \{0,1\}^{c(\lambda)} \mid \Pr[\mathcal{V}er(k,p) = 1] \geq Q\}\right| \leq M(\lambda),$$

where the probability is over the measurements of $\mathcal{V}er$*.*

Definition 9 (PRSPD with Pseudorandom-proofs)**.** *A PRSPD with* Pseudorandom-proofs *is a PRSPD scheme if for every polynomial function* $t(\lambda)$*,*

$$\{f_i\}_{\forall i \in [t], d_i \leftarrow \mathcal{D}estruct(\mathcal{G}en(k)); k \overset{u}{\leftarrow} \{0,1\}^w} \approx_c \{u_1, \ldots, u_t\}_{\forall i \in [t], u_i \overset{u}{\leftarrow} \{0,1\}^c}.$$

Note that by the pseudorandomness guarantee of a PRSPD (see Definition 1),

$$\{f_i\}_{\forall i \in [t], d_i \leftarrow \mathcal{D}estruct(\mathcal{G}en(k)); k \overset{u}{\leftarrow} \{0,1\}^w} \approx_c \{f_i\}_{\forall i \in [t], f_i \leftarrow \mathcal{D}estruct(|\psi\rangle)}, |\psi\rangle \sim \mu_{\mathcal{H}_n},$$

where $\mathcal{H}_n = (\mathbb{C}^2)^{\otimes n}$. Therefore, we can view Definition 9 as a property of $\mathcal{D}estruct$ algorithm that the distribution $\{f_i\}_{\forall i \in [t], f_i \leftarrow \mathcal{D}estruct(|\psi\rangle)}, |\psi\rangle \sim \mu_{\mathcal{H}_n}$ it generates, is statistically close to a product of uniform distribution.

Definition 10 (Pseudorandom states with *nice* proofs of destruction)**.** *A Pseudorandom states with nice proofs of destruction (PRSNPD) is a* $\left(1 - \frac{1}{r(\lambda)}, 2^{m(\lambda)}\right)$ *-bounded proofs PRSPD (see Definition 8) with* Pseudorandom-proofs *(see Definition 9) such that* $r(\lambda) \in \mathsf{poly}(\lambda)$ *and* $c(\lambda) - 2w(\lambda) - 2m(\lambda) \in \omega(\log(\lambda))$*, where* $w(\lambda)$ *and* $c(\lambda)$ *are the key length and proof length respectively.*

Proposition 3. *Assume there exist post-quantum one-way functions. Then, a PRSNPD scheme (Definition 10) with key length* $w(\lambda) = \lambda$*, input length* $d(\lambda) = \lambda$*, output length* $n(\lambda) = 5 \cdot \lambda$ *and proof length* $c(\lambda) = 5 \cdot \lambda$ *exists.*

Proof. In our main Theorem 6, we prove that there is a PRFSPD with the above parameters. First, Remark 3 explains how the PRFSPD can be easily turned into a PRSPD: by simply querying the generation algorithm $\mathcal{Gen}(k, \cdot)$ only with $x = 0^d$, i.e. the generation algorithm of the PRSPD is $\mathcal{Gen}(k, 0^d)$, and the proof verification is $\mathcal{Ver}(k, 0^d, \cdot)$. It remains to observe that this same construction from Section 3 satisfies the niceness properties: For the property of pseudorandom proofs, note that the proof generation algorithm $\mathcal{Destruct}$ is simply a measurement in the standard basis. This, combined with the fact that the quantum states are pseudorandom, implies that a measurement in the standard basis yields a pseudorandom classical string.

To see why the construction is also bounded-proofs, note that the verification algorithm $\mathcal{Ver}(k, 0^d, \cdot)$ accepts exactly 2^λ classical strings with probability 1 (that is, the set $\{y \in \{0,1\}^\lambda : \mathsf{PRP}_k(y, 0^d, 0^{3\lambda})\}$) and rejects the rest. The proof length is $c(\lambda) := 5 \cdot \lambda$ and the key length is $w(\lambda) := \lambda$, which means that $c(\lambda) - 2 \cdot w(\lambda) - 2 \cdot \lambda = \lambda$, which is $\omega(\log(\lambda))$, as needed. □

4.3.2 Construction of a statistically-binding and computationally-hiding bit commitments from PRSNPD

We start with a Pseudorandom states with nice proofs of destruction (PRSNPD) (see Definition 10), i.e., a PRSPD with Pseudorandom-proofs and $\left(1 - \frac{1}{r(\lambda)}, 2^{m(\lambda)}\right)$-bounded proofs PRSPD such that $r \in \mathsf{poly}(\lambda)$ and $m(\lambda)$ satisfies $c(\lambda) - 2w(\lambda) - 2m(\lambda) \in \omega(\log(\lambda))$, where $w(\lambda)$ and $c(\lambda)$ are key length and proof length respectively. The construction given in Fig. 6, is obtained by adapting the construction of statistically-binding computational-hiding bit commitments from PRGs [Nao89] to PRSPD.

Theorem 14 (Correctness of $\mathcal{BC}$)**.** *The* bit-commitment $\mathcal{BC}$ *is correct (see Definition 7) if* $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ *satisfies correctness (see Definition 1).*

The proof is immediate from the correctness of $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$; hence we omit the proof.

Assumes: PRSNPD scheme, $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$
Commit$(\mathcal{C}_\lambda(b), \mathcal{R}_\lambda)$

1. The receiver $\mathcal{R}_\lambda$ samples $c_1, c_2, \ldots, c_{\tilde{r}} \overset{u}{\leftarrow} \{0,1\}^c$ independently and uniformly, and sends them to the committer $\mathcal{C}_\lambda$.
2. $\mathcal{C}_\lambda$ samples $k \overset{u}{\leftarrow} \{0,1\}^w$ and generates $\tilde{r}$ proofs of destructions $(p_1, \ldots, p_{\tilde{r}}) \leftarrow \mathcal{Destruct}^{\otimes \tilde{r}}\left((\mathcal{Gen}(k))^{\otimes \tilde{r}}\right)$, where $\tilde{r}(\lambda) = \lambda \cdot r(\lambda)$.
3. If $b = 0$, $\mathcal{C}_\lambda$ sends $s_0 = (p_1, \ldots, p_{\tilde{r}})$ to $\mathcal{R}_\lambda$, else sends $s_1 = (c_1 \oplus p_1, \ldots, c_{\tilde{r}} \oplus p_{\tilde{r}})$ to $\mathcal{R}_\lambda$.
4. The state held by the registers of $\mathcal{C}_\lambda$ and $\mathcal{R}_\lambda$ at the end of this phase is the classical string $\sigma_{\mathcal{C},\mathcal{R}} = \sigma_{\mathcal{C}} \otimes \sigma_{\mathcal{R}}$, where $\sigma_{\mathcal{C}} = k$, and $\sigma_{\mathcal{R}} = s_b, c_1, \ldots, c_{\tilde{r}}$.

Reveal$(\mathcal{C}_\lambda, \mathcal{R}_\lambda, \sigma_{\mathcal{C},\mathcal{R}})$

1. Interpret the classical string $\sigma_{\mathcal{C},\mathcal{R}} = \sigma_{\mathcal{C}} \otimes \sigma_{\mathcal{R}}$, where $\sigma_{\mathcal{C}} = \tilde{k}$ is a w-bit string, and $\sigma_{\mathcal{C}} = \tilde{p}_1, \ldots \tilde{p}_{\tilde{r}}, \tilde{c}_1, \ldots, \tilde{c}_{\tilde{r}}$ a concatenation of $2\tilde{r}$ many c-bit string.
2. $\mathcal{C}_\lambda$ sends $\tilde{k}$ to $\mathcal{R}$.
3. $\mathcal{R}_\lambda$ runs $\mathcal{Ver}(\tilde{k}, \tilde{p}_i)$ for every $i \in [\tilde{r}]$. If all the $\mathcal{Ver}$ runs accept, then output 0, else $\mathcal{R}_\lambda$ runs $\mathcal{Ver}(\tilde{k}, \tilde{p}_i \oplus \tilde{c}_i)$ for every $i \in [\tilde{r}]$. If all the $\mathcal{Ver}$ runs accept, then output 0, else output $\perp$.

Figure 6: Bit-commitment scheme $\mathcal{BC}$.

Theorem 15 (Computational hiding of $\mathcal{BC}$)**.** *The* bit-commitment $\mathcal{BC}$ *is* computational-hiding *if* $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Verify})$ *satisfies* Pseudorandom-proofs *property (see Definition 9).*

The proof is the same as the proof of computational-hiding for the PRG-based construction in [Nao89]. Due to lack of space, we omit the proof from this version. The full proof is given in Appendix E.2 on Page 67.

Theorem 16 (Statistical binding of $\mathcal{BC}$)**.** *The* bit-commitment $\mathcal{BC}$ *is* statistically-binding *if* $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Verify})$ *has* $\left(1 - \frac{1}{r(\lambda)}, 2^{m(\lambda)}\right)$*-bounded proofs property (see Definition 8).*

Proof sketch of Theorem 16. Let $\mathcal{C}^*_\lambda$ be the malicious committer and $\mathcal{R}_\lambda$ be the honest receiver. Let Bad-String $\subset \{0,1\}^c$ be the set of all strings e such that there exists $k_1, k_2 \in \{0,1\}^w$, and f_1, f_2 such that $f_2 = e \oplus f_1$, i.e., $f_1 \oplus f_2 = c$ and

$$\Pr[\mathcal{Ver}(k_1, f_1) = 1] \geq \left(1 - \frac{1}{r(\lambda)}\right), \quad \Pr[\mathcal{Ver}(k_2, f_2) = 1] \geq \left(1 - \frac{1}{r(\lambda)}\right).$$

Let $\mathsf{Good\text{-}String} = \overline{\mathsf{Bad\text{-}String}}$. Observe that if the random strings sent from $\mathscr{R}_\lambda$ in the *Commit* phase are all in $\mathsf{Good\text{-}String}$, then the malicious committer cannot reveal to both 0 and 1 in the *Reveal* phase except with negligible probability. The parameters for the bounded proofs property have been chosen such that $\mathsf{Good\text{-}String}$ accounts for an overwhelming fraction of $\{0,1\}^c$; hence all the random strings from $\mathscr{R}_\lambda$, which are polynomially many, will indeed be in $\mathsf{Good\text{-}String}$ with overwhelming probability.

□

4.4 CMA-Secure MAC

Definition 11 (Length-restricted strong CMA-secure MAC(Adapted from [Gol04, Definition 6.2.1])). *A length-restricted* CMA *secure* MAC *scheme* $(\mathcal{M})$ *with message length* $d(\lambda)$[10]*, key-length* $w(\lambda)$*, and tag-length* $c(\lambda)$ *is a tuple of* QPT *algorithms* $(\mathscr{Sign}, \mathscr{Verify})$ *with the following syntax:*

- $\mathsf{sig} \leftarrow \mathscr{Sign}(k, m)$*: takes a key* $k \in \{0,1\}^{w(\lambda)}$*, a message* $m \in \{0,1\}^{d(\lambda)}$*, and outputs a tag* $\mathsf{sig} \in \{0,1\}^c$*.*
- $b \leftarrow \mathscr{Verify}(k, m, \mathsf{sig})$*: takes a key* k*, a message* m*, a tag* sig*, and outputs a boolean value, either accept* $(b = 1)$ *or reject* $(b = 0)$*.*

Correctness. *For every message* $m \in \{0,1\}^{d(\lambda)}$*,*

$$\Pr[k \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}; \mathsf{sig} \leftarrow \mathscr{Sign}(k, m) : \mathscr{Verify}(k, \mathsf{sig}) = 1] = 1.$$

Strong CMA **Unforgeability.** *For every* QPT *adversary* $\mathcal{A}$ *in the forging game (see Game 7), there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Strong\text{-}CMA\text{-}Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathcal{M}} = 1] = \mathsf{negl}(\lambda).$$

Game 7 $\mathsf{Strong\text{-}CMA\text{-}Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathcal{M}}$

```
Given input $1^\lambda$, the challenger samples $k \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}$. The challenger also initializes an empty set $S$.
$\mathcal{A}$ gets classical oracle access to $\mathscr{Verify}(k, \cdot)$ and $\mathscr{Sign}(k, \cdot)$.
for $\mathscr{Sign}(k, \cdot)$ query $x$ made by $\mathcal{A}$ do
    Add $(x, \sigma_x)$ to $S$, where $\sigma_x$ is the response of $\mathscr{Sign}(k, \cdot)$ oracle on input $x$.
end for
$\mathcal{A}$ outputs $x', \sigma'_x$ to the challenger.
Return 1 if $(x', \sigma'_x) \notin S$ and $\mathscr{Verify}(k, x', \sigma'_x) = 1$.
```

[10] This is referred to as d-restricted MAC in [Gol04].

Remark 17 (Strong vs. vanilla CMA security)*.* Vanilla CMA security considers a similar forging game in which the adversary wins if she produces (m, σ) that passes verification and that m was never queried to the $\mathcal{S}ign$ oracle. In comparison, Definition 11 is a stronger notion because the adversary wins the forging game even if she produces a valid (m, σ) such that m was queried to the $\mathcal{S}ign$ oracle as long as σ was not received as a response in any of the m-queries she did to the $\mathcal{S}ign$ oracle. Hence, we call this notion the strong CMA security, also referred to as super-secure MACs[11] in [Gol04, Section 6.5.2]. These notions are not known to be equivalent in general. However, the prominent classical MAC constructions have deterministic signing procedures, i.e., every message has a unique signature string that passes verification, and all other strings are rejected. For such MAC schemes, strong and vanilla CMA security are equivalent. This is not the case for our construction. Hence we consider the strongest possible definition.

Remark 18 (Access to the verification oracle)*.* In the classical CMA security definitions, the adversary is not given access to the verification because the classical MAC schemes usually have deterministic signing procedures; hence the verification oracle can be simulated using the signing oracle, see [Gol04, Proposition 6.1.3]. However, MAC schemes with quantum algorithms do not have a deterministic signing in general; hence, we provide the adversary in $\mathsf{Strong\text{-}CMA\text{-}Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathcal{M}}$ explicit access to the verification oracle.

4.4.1 Construction from PRFSPD

Next we construct a length-restricted CMA secure MAC scheme with input-length $d(\lambda)$, key-length $w(\lambda)$ and tag-length $c(\lambda)$ from a PRFSPD scheme $(\mathcal{G}en, \mathcal{D}estruct, \mathcal{V}er)$ with key-length $w(\lambda)$, input-length $d(\lambda)$ and proof-length $c(\lambda)$. The construction given in Fig. 7, combines the quantum MAC construction in [AQY21] with the proof of destruction property of PRFSPD, to get an improved construction in the following two aspects. Firstly, the tags in our construction are classical, whereas [AQY21] requires quantum tags. Additionally, our construction satisfies strong-CMA security while [AQY21] considers vanilla CMA security. We also briefly mention that our construction supports any poly-size message, whereas the one in [AQY21] is length-restricted. We note that we remove this length restriction using a standard technique, which is applicable to their construction as well.

Theorem 19 (Correctness of $\mathcal{M}$)**.** *The length-restricted* MAC *scheme* $\mathcal{M}$ *is correct (see Definition 11) if* $(\mathcal{G}en, \mathcal{D}estruct, \mathcal{V}er)$ *satisfies correctness (see Definition 2).*

The proof is immediate from the correctness of the proof of destruction of the underlying PRFSPD, and hence we omit the proof.

[11] We use the term *strong* in place of *super* because *strong* is the more colloquially accepted term.

Assumes: PRFSPD scheme, $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$
$\mathcal{Sign}(k, m)$
Output $\mathsf{sig} = \mathcal{Destruct}(\mathcal{Gen}(k, m))$.
$\mathcal{Verify}(k, m, \mathsf{sig})$
Output $\mathcal{Ver}(k, m, \mathsf{sig})$).

Figure 7: MAC scheme $\mathcal{M}$.

Theorem 20 (CMA Security of $\mathcal{M}$)**.** *The length-restricted* MAC *scheme* $\mathcal{M}$ *is* CMA*-secure if* $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ *is a* PRFSPD *(see Definition 1).*

Proof. The proof directly follows from the classical-unclonablility of proofs for PRFSPD given in Lemma 5. □

Remark 21 (Unrestricted MAC)*.* Note that the input-length of the MAC, $d(\lambda) \in \omega(\log(\lambda))$. Hence, we can extend the MAC scheme to sign messages of arbitrary polynomial length by dividing the message into blocks and signing them individually; see [Gol04, Theorem 6.2.2] for more details. Therefore, we conclude that PRFSPD implies CMA MAC, in a black-box manner.

4.5 CPA-Secure Symmetric Encryption

In this section, we will construct CPA-secure symmetric bit-encryption from PRFSPD, which can be easily extended to a CPA-secure and even CCA-2 encryption for arbitrary message-length, see Remarks 23 and 24.

Definition 12 (CPA-secure symmetric bit-encryption(Adapted from [Gol04, Definition 5.4.9]))**.** *A* CPA *secure symmetric bit-encryption* $\mathcal{E}$ *with key space* $\{0,1\}^{w(\lambda)}$*, and cipher space* $\{0,1\}^{c(\lambda)}$ *is a tuple of* QPT *algorithms* $(\mathcal{Enc}, \mathcal{Dec})$ *with the following syntax:*

- $\mathsf{ct} \leftarrow \mathcal{Enc}(k, m)$*: takes a key* k *and a message bit* m*, and outputs a classical cipher text* ct*.*
- $m \leftarrow \mathcal{Dec}(k, \mathsf{ct})$*: takes a key* k*, a cipher text* ct*, and outputs a message bit* m*.*

Correctness: *For every message* $m \in \{0,1\}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$*, such that*

$$\Pr[k \leftarrow \{0,1\}^{w(\lambda)}; \mathsf{ct} \leftarrow \mathcal{Enc}(k, m); m' \leftarrow \mathcal{Dec}(k, \mathsf{ct}) : m = m'] \geq 1 - \mathsf{negl}(\lambda).$$

CPA **security** *For every* QPT *adversary* $\mathcal{A}$ *in the distinguishability game (see Game 8, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Distinguish\text{-}Exp}_{\lambda}^{\mathcal{A},\mathcal{E}} = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Game 8 $\mathsf{Distinguish\text{-}Exp}_{\lambda}^{\mathcal{A},\mathcal{E}}$

```
Given input $1^{\lambda}$, the challenger samples $k \xleftarrow{u} \{0,1\}^{w(\lambda)}$.
$\mathcal{A}$ gets classical oracle access to $\mathit{Enc}(k,\cdot)$.
Challenger samples a bit $b$ and computes $\mathsf{ct}_b \leftarrow \mathit{Enc}(k,b)$ and sends $\mathsf{ct}_b$ to $\mathcal{A}$.
$\mathcal{A}$ outputs $b'$ to the challenger.
The output of the experiment is 1 if $b = b'$.
```

4.5.1 Construction from PRFSPD

Let $(\mathit{Gen}, \mathit{Destruct}, \mathit{Ver})$ be a PRFSPD with input length $d(\lambda) \in \omega(\log(\lambda))$, and key-length $w(\lambda)$. We will give a construction of CPA secure symmetric bit-encryption from such a PRFSPD with key-length $w(\lambda)$.

In a nutshell, our construction combines the ideas in [AQY21], with the proof of destruction property of PRFSPD state, to make the ciphers classical. The construction is given in Fig. 8.

Assumes: PRFSPD scheme, $(\mathit{Gen}, \mathit{Destruct}, \mathit{Ver})$

$\mathit{Enc}(k,b)$

1. Sample $r \leftarrow \{0,1\}^{d(\lambda)-1}$.
2. Output $\mathsf{ct} = (r, \mathit{Destruct}(\mathit{Gen}(k, r\|b)))$.

$\mathit{Dec}(k,\mathsf{ct})$

1. Interpret ct as r', c', where $r' \in \{0,1\}^{d(\lambda)-1}$.
2. Run $\mathit{Ver}(k, r'\|1, c)$. If accepted output 1 else 0.

Figure 8: Symmetric bit-encryption $\mathcal{E}$.

Proposition 4 (Correctness of $\mathcal{E}$)**.** *The symmetric bit-encryption scheme $\mathcal{E}$ is correct (see Definition 12) if $(\mathit{Gen}, \mathit{Destruct}, \mathit{Ver})$ satisfies correctness (see Definition 2).*

Proof. By the correctness of the underlying PRFSPD, the correctness holds for encryptions of 1, i.e.,

$$\Pr[k \leftarrow \{0,1\}^{w(\lambda)}; \mathsf{ct} \leftarrow \mathit{Enc}(k,1); m' \leftarrow \mathit{Dec}(k,\mathsf{ct}) : 1 = m'] = 1.$$

Next for encryptions of 0, it suffices to show that there exists a negligible function $\mathsf{negl}(\lambda)$ such that,

$$\mathsf{prob}_0 \equiv \Pr[k \leftarrow \{0,1\}^{w(\lambda)}; \mathsf{ct} \leftarrow \mathit{Enc}(k,0); m' \leftarrow \mathit{Dec}(k,\mathsf{ct}) : 1 = m'] = \mathsf{negl}(\lambda). \tag{9}$$

The last equation can be proven using the Unforgeability-of-proofs property of the underlying PRFSPD as follows. We construct an adversary $\mathcal{A}$ in the cloning game $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{PRFSPD}}$ (see Game 2) that samples $r_0 \xleftarrow{u} \{0,1\}^{d-1}$ and queries the $\mathcal{G}en$ oracle at $r_0\|1$ and gets a state $|\psi^{r_0\|0}\rangle$. $\mathcal{A}$ runs $\mathcal{D}estruct$ on $|\psi^{r_0\|0}\rangle$ to get a proof $p \leftarrow \mathcal{D}estruct(|\psi^{r_0\|0}\rangle)$, and finally submits $r\|0, p$. Note that $\mathcal{A}$ never queried $r\|0$ to $\mathcal{G}en$ before, so she wins the cloning game if $r\|0, p$ passes the PRFSPD verification which by design, happens with probability exactly prob_0. Hence by the Unforgeability-of-proofs property of PRFSPD, prob_0 must be negligible. □

Theorem 22 (CPA Security of $\mathcal{E}$)**.** *The symmetric bit-encryption scheme $\mathcal{E}$ is* CPA*-secure (see Definition 12) if $(\mathcal{G}en, \mathcal{D}estruct, \mathcal{V}erify)$ is a PRFSPD (see Definition 1).*

Proof. The proof follows essentially from the pseudorandomness of $(\mathcal{G}en, \mathcal{D}estruct, \mathcal{V}erify)$ (see Definition 2). We will consider the following sequence of hybrids:

$\mathcal{H}_0$ This is the real security game $\mathsf{Distinguish\text{-}Exp}_{\lambda}^{\mathcal{A},\mathcal{E}}$. Since we are considering bit encryption, the challenger simply samples a bit b and feeds $\mathcal{A}$ the encryption of b, i.e., $(r, \mathcal{D}estruct(\mathcal{G}en(k, r_{ch}\|b)))$ at the challenge phase. The adversary $\mathcal{A}$ is given classical access to the CPA oracle $\mathcal{E}nc(k,\cdot)$ which she can query both before and after the challenge phase. Let $b_1, \ldots, b_q$ be the queries $\mathcal{A}$ makes to the CPA oracle where $q \in \mathsf{poly}(\lambda)$ (since $\mathcal{A}$ is polynomially bounded), and

$$\{\mathcal{E}nc(k, b_1), \ldots \mathcal{E}nc(k, b_q)\} = \{(r_1, \mathcal{D}estruct(\mathcal{G}en(k, r_1\|b_1))), \ldots (r_q, \mathcal{D}estruct(\mathcal{G}en(k, r_q\|b_q)))\},$$

be the respective responses from the oracle, where $r_1, \ldots, r_q$ are chosen uniformly. $\mathcal{A}$ succeeds if she submits a bit b' at the end, such that $b = b'$.

$\mathcal{H}_1$ In this hybrid, we only change the distribution on $r, r_1, \ldots, r_q$. The challenger samples r independently, and then for every $i \in [q]$, r_i is chosen uniformly from $\{0,1\}^{d(\lambda)-1} \setminus \{r, r_1, \ldots, r_{i-1}\}$, where $\{r, r_1, \ldots, r_{i-1}\}$ should be interpreted as $\{r\}$ for $i = 1$. Note that the distributions on $(r, r_1, \ldots, r_q)$ in the hybrid have negligible statistical distance from the uniformly random distribution that we had in $\mathcal{H}_0$ because $q \in \mathsf{poly}(\lambda)$ and the length of r is $d(\lambda) - 1 \in \omega(\log \lambda)$. Hence, the success probability of $\mathcal{A}$ in $\mathcal{H}_0$ and $\mathcal{H}_1$ are negligibly close.

$\mathcal{H}_2$ In this hybrid, we replace
$\{(r_1, \mathcal{D}estruct(\mathcal{G}en(k, r_1\|b_1))), \ldots (r_q, \mathcal{D}estruct(\mathcal{G}en(k, r_q\|b_q))), (r, \mathcal{D}estruct(\mathcal{G}en(k, r\|b))\}$
with
$\{(r_1, \mathcal{D}estruct(|\phi_1\rangle), \ldots (r_q, \mathcal{D}estruct(|\phi_q\rangle), (r, \mathcal{D}estruct(|\phi_{q+1}\rangle\}$ where $|\phi\rangle \sim \mu_{\mathcal{H}_n}$ and for each $i \in [q]$, $|\phi_i\rangle \sim \mu_{\mathcal{H}_n}$ independently.

Let the difference in the success probabilities of the $\mathcal{A}$ in $\mathcal{H}_1$ and $\mathcal{H}_2$ be p. We can construct an adversary $\mathcal{B}$ who can violate the pseudorandomness (see Definition 2) of $(\mathcal{G}en, \mathcal{D}estruct, \mathcal{V}er)$ with distinguishing advantage p. $\mathcal{B}$

simulates $\mathcal{A}$ and when she queries b_i in the i^{th} query, $\mathcal{B}$ generates r_i uniformly from $\{0,1\}^{d(\lambda)-1} \setminus r_1, \ldots, r_{i-1}$, and queries the oracle (which she needs to distinguish) on $r_i \| b_i$ and performs $\mathscr{Destruct}$ on the output she receives and feeds the obtained string to $\mathcal{A}$. Moreover, $\mathcal{B}$ plays the role of the challenger and samples a uniformly random bit b, and feeds the encryption of b using the challenge oracle that she has access to. $\mathcal{B}$ outputs 1 if the output of the $\mathcal{A}$ is the same as b. Clearly, the distinguishing probability is p. Hence, p must be negligible.

Now note that in $\mathcal{H}_2$, the challenge bit b is information-theoretically hidden from $\mathcal{A}$. Therefore, her success probability in $\mathcal{H}_2$ must be at most $\frac{1}{2}$.

Hence, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr[\mathcal{A} \text{ wins } \mathcal{H}_0] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

□

Remark 23 (Encryption of arbitrarily long messages). Any CPA-secure bit encryption scheme can be extended to a CPA-secure encryption to arbitrarily long messages via bit-by-bit encryption, see [Gol04, Section 5.3.2.2]. Hence, we conclude that there is a black-box construction of CPA-secure encryption for arbitrary message lengths, from PRFSPD.

Remark 24 (CCA-2 security of $\mathcal{E}$). By combining the strong MAC scheme from PRFSPD (see Theorems 19 and 20) with the CPA-secure encryption scheme mentioned in the previous remark using the Encrypt-then-MAC, we conclude that there is a black-box construction of CCA-2 secure encryption for arbitrarily long messages from PRFSPD.

Acknowledgments

Amit Behera and Or Sattath were supported by the Israeli Science Foundation (ISF) grant No. 682/18 and 2137/19, and by the Cyber Security Research Center at Ben-Gurion University. Zvika Brakerski is supported by the Israel Science Foundation (Grant No. 3426/21), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482). Omri Shmueli is supported by the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreements No. 101042417, acronym SPP, and No. 756482, acronym REACT), by Israeli Science Foundation (ISF) grants 18/484 and 19/2137, by Len Blavatnik and the Blavatnik Family Foundation, and by the Clore Israel Foundation. The authors would like to thank the anonymous Crypto 2023 reviewers for their valuable and insightful comments.

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 756482).

References

[Aar18] S. Aaronson. Shadow Tomography of Quantum States. In I. Diakonikolas, D. Kempe, and M. Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 325–338. ACM, 2018, arXiv: `1711.01053`.

[AGKZ20] R. Amos, M. Georgiou, A. Kiayias, and M. Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath, and J. Chuzhoy, editors, *Proccedings of the Annual ACM SIGACT Symposium on Theory of Computing,*, pages 255–268. ACM, 2020, Cryptology ePrint Archive: `Report 2020/107`.

[AGQY22] P. Ananth, A. Gulati, L. Qian, and H. Yuen. Pseudorandom (Function-Like) Quantum State Generators: New Definitions and Applications, 2022, arXiv: `2211.01444`.

[AQY21] P. Ananth, L. Qian, and H. Yuen. Cryptography from Pseudorandom Quantum States, 2021, arXiv: `2112.10020`.

[BCKM21] J. Bartusek, A. Coladangelo, D. Khurana, and F. Ma. One-Way Functions Imply Secure Computation in a Quantum World. In T. Malkin and C. Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 467–496. Springer, 2021, arXiv: `2011.13486`.

[BCQ22] Z. Brakerski, R. Canetti, and L. Qian. On the computational hardness needed for quantum cryptography, 2022, arXiv: `2209.04101`.

[BDS16] S. Ben-David and O. Sattath. Quantum Tokens for Digital Signatures. QCrypt 2017, 2016.

[BFV20] A. Bouland, B. Fefferman, and U. V. Vazirani. Computational Pseudorandomness, the Wormhole Growth Paradox, and Constraints on the AdS/CFT Duality (Abstract). In T. Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 63:1–63:2. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[BMR90] D. Beaver, S. Micali, and P. Rogaway. The Round Complexity of Secure Protocols (Extended Abstract). In H. Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 503–513. ACM, 1990.

[BS19] Z. Brakerski and O. Shmueli. (Pseudo) Random Quantum States with Binary Phase. In *Theory of Cryptography - 17th International Conference, TCC 2019*, LNCS, pages 229–250. Springer, 2019, arXiv: `1906.10611`.

[BS20a] A. Behera and O. Sattath. Almost Public Coins. QIP 2021, 2020, arXiv: `2002.12438`.

[BS20b] Z. Brakerski and O. Shmueli. Scalable Pseudorandom Quantum States. In D. Micciancio and T. Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020,* *Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 417–440. Springer, 2020, arXiv: `2004.01976`.

[CLLZ21] A. Coladangelo, J. Liu, Q. Liu, and M. Zhandry. Hidden Cosets and Applications to Unclonable Cryptography. In T. Malkin and C. Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 556–584. Springer, 2021, Cryptology ePrint Archive: `Report 2021/946`.

[CS20] A. Coladangelo and O. Sattath. A Quantum Money Solution to the Blockchain Scalability Problem. *Quantum*, 4:297, 2020.

[Gol04] O. Goldreich. *The Foundations of Cryptography - Volume 2: Basic Applications.* Cambridge University Press, 2004.

[Har13] A. W. Harrow. The Church of the Symmetric Subspace, 2013, arXiv: `1308.6595`.

[JLS18] Z. Ji, Y. Liu, and F. Song. Pseudorandom Quantum States. In H. Shacham and A. Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 126–152. Springer, 2018, arXiv: `1711.00385`.

[Kre21] W. Kretschmer. Quantum Pseudorandomness and Classical Complexity. In M. Hsieh, editor, *16th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2021, July 5-8, 2021, Virtual Conference*, volume 197 of *LIPIcs*, pages 2:1–2:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, arXiv: `2103.09320`.

[Lam79] L. Lamport. Constructing Digital Signatures from a One Way Function. Technical Report CSL-98, SRI International, October 1979. This paper was published by IEEE in the Proceedings of HICSS-43 in January, 2010.

[LC97] H.-K. Lo and H. F. Chau. Is Quantum Bit Commitment Really Possible? *Physical Review Letters*, 78(17):3410–3413, Apr 1997, arXiv: `quant-ph/9603004`.

[May97] D. Mayers. Unconditionally Secure Quantum Bit Commitment is Impossible. *Physical Review Letters*, 78(17):3414–3417, April 1997, arXiv: `quant-ph/9605044`.

[MVW12] A. Molina, T. Vidick, and J. Watrous. Optimal Counterfeiting Attacks and Generalizations for Wiesner's Quantum Money. In K. I. et al., editor, *Theory of Quantum Computation, Communication, and Cryptography, TQC*, volume 7582 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2012, arXiv: `1202.4010`.

[MY22a] T. Morimae and T. Yamakawa. Quantum Commitments and Signatures Without One-Way Functions. In Y. Dodis and T. Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 269–295. Springer, August 2022, arXiv: `2112.06369`.

[MY22b] T. Morimae and Y. Yamakawa. One-Wayness in Quantum Cryptography, October 2022, arXiv: `2210.03394`.

[Nao89] M. Naor. Bit Commitment Using Pseudo-Randomness. In G. Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Pro-*

ceedings, volume 435 of *Lecture Notes in Computer Science*, pages 128–136. Springer, 1989.

[RS19] R. Radian and O. Sattath. Semi-Quantum Money. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies, AFT 2019, Zurich, Switzerland, October 21-23, 2019*, pages 132–146. ACM, 2019, arXiv: `1908.08889`.

[RTV04] O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of Reducibility between Cryptographic Primitives. In M. Naor, editor, *TCC 2004, Cambridge, MA, USA Proceedings*, volume 2951 of *LNCS*, pages 1–20. Springer, 2004.

[Shm22a] O. Shmueli. Public-key Quantum money with a classical bank. In S. Leonardi and A. Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 790–803. ACM, 2022, Cryptology ePrint Archive: `Report 2021/1427`.

[Shm22b] O. Shmueli. Semi-Quantum Tokenized Signatures. Cryptology ePrint Archive, Report 2022/228, 2022. `https://ia.cr/2022/228`.

[Wat18] J. Watrous. *The Theory of Quantum Information.* Cambridge University Press, April 2018.

[Zha12] M. Zhandry. How to Construct Quantum Random Functions. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 679–687. IEEE Computer Society, 2012.

[Zha16] M. Zhandry. A Note on Quantum-Secure PRPs, 2016, Cryptology ePrint Archive: `http://eprint.iacr.org/2016/1076`.

[Zha21] M. Zhandry. Quantum Lightning Never Strikes the Same State Twice. Or: Quantum Money from Cryptographic Assumptions. *J. Cryptol.*, 34(1):6, 2021, arXiv: `1711.02276`.

A Notations, standard cryptographic definitions and facts

For any finite set S, we use $s \overset{u}{\leftarrow} S$ to denote uniformly random sampling from the set S. Next, we recall several definitions and results from cryptography that are necessary for this work.

A.1 Cryptographic primitives

We assume that for any algorithm in a cryptographic scheme except for the bit-commitment scheme in Section 4.3, the security parameter can be computed efficiently from the input length. Hence, we follow the convention that the algorithms, except for the key generation (since it only receives the security parameter as input) and the interactive algorithms in bit-commitment schemes, do not receive the security parameter additionally as an input.

We use English alphabets such as $c(\lambda), d(\lambda), w(\lambda), n(\lambda)$, etc., to denote parameters in cryptographic primitives, that are functions of the security parameter, λ. However, we drop λ from their description in the proofs for brevity; for example, w instead of $w(\lambda)$. Pseudorandom functions (PRF) and pseudorandom permutations (PRP) are important constructions in classical cryptography. Intuitively, they are families of functions or permutations that look like truly random functions or permutations to polynomial-time machines. In the quantum case, we need a strong requirement that they still look random even to polynomial-time quantum algorithms.

Definition 13 (Quantum-Secure Pseudorandom Functions and Permutations)**.** *Let $\mathcal{K}$, $\mathcal{X}$, $\mathcal{Y}$ be the key space, the domain, and the range, all implicitly depending on the security parameter λ. A keyed family of functions $\{\mathsf{PRF}_k : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}}$ is a quantum-secure pseudorandom function (PRF) if for any polynomial-time quantum oracle algorithm $\mathcal{A}$,*

$$\left| \Pr_{k\leftarrow\mathcal{K}}[\mathcal{A}^{\mathsf{PRF}_k}(1^\lambda) = 1] - \Pr_{f\leftarrow\mathcal{Y}^\mathcal{X}}[\mathcal{A}^f(1^\lambda) = 1] \right| = \mathsf{negl}(\lambda). \tag{10}$$

Similarly, a keyed family of permutations $\{\mathsf{PRP}_k \in S_\mathcal{X}\}_{k\in\mathcal{K}}$ is a quantum-secure pseudorandom permutation (PRP) if for any polynomial-time quantum oracle algorithm $\mathcal{A}$,

$$\left| \Pr_{k\leftarrow\mathcal{K}}[\mathcal{A}^{\mathsf{PRP}_k,\mathsf{PRP}_k^{-1}}(1^\lambda) = 1] - \Pr_{P\leftarrow S_\mathcal{X}}[\mathcal{A}^{P,P^{-1}}(1^\lambda) = 1] \right| = \mathsf{negl}(\lambda). \tag{11}$$

In addition, both PRF_k and PRP_k are polynomial-time computable.

Theorem 25. *PRFs and PRPs exist if quantum-secure one-way functions exist.*

Zhandry proved the existence of PRFs, assuming the existence of one-way functions that are hard to invert, even for quantum algorithms [Zha12]. Assuming PRF, one can construct PRP using various *shuffling* constructions [Zha16].

A.2 Quantum Information

For any $n \in \mathbb{N}$, we use $\mathcal{H}_n$ to denote the Hilbert space on n-qubit registers, i.e., $\mathcal{H}_n = \mathbb{C}^{2^{\otimes n}}$, and N to denote N, the dimension of $\mathbb{C}^{2^{\otimes n}}$. Note that the optimal distinguishing probability between two n-qubit quantum (possibly mixed) states ρ_0 and ρ_1 is given by their trace distance $\mathrm{D}(\rho_0, \rho_1)$, defined as

$$\mathrm{D}(\rho_0, \rho_1) \stackrel{\text{def}}{=} \frac{1}{2} \left\| \rho_0 - \rho_1 \right\|_1 . \tag{12}$$

We now turn to discuss standard properties of symmetric subspaces; for an in-depth discussion, see [Har13]. For Hilbert space $\mathcal{H}_n$ of dimension N, i.e., it represents an n-qubit system, and integer t, we use $\vee^t\mathcal{H}_n$ to denote the symmetric subspace of $\mathcal{H}_n^{\otimes t}$, the subspace of states that are invariant under permutations of the subsystems. Let $\mathcal{X}$ be the set $\{0, 1, \ldots, N-1\}$ such that $\mathcal{H}_n$ is the span of $\{|x\rangle\}_{x\in\mathcal{X}}$.

For any subset $A \subset \mathcal{X}$, we use $\mathcal{H}_A$ to denote the subspace $\mathsf{Span}(A)$, and $\vee^t \mathcal{H}_A$ to denote the symmetric subspace of $\mathcal{H}_A^{\otimes t}$.

For any $t \in \mathbb{N}$, let $\mathbb{N}_t^A$ be the set of all vectors $\vec{z}$ in $\mathbb{N}^A$ such that $\sum_{j \in A} z_j = t$. We will abbreviate $\mathbb{N}_t^A$ as $\mathbb{N}_t$ for the special case $A = \mathcal{X}$. For any $\mathbf{x} = (x_1, x_2, \ldots, x_t) \in A^t$, denote $k(\mathbf{x})$ to be the associated vector in $\mathbb{N}_t^A$, i.e., the y^{th} coordinate of $\vec{z}$ is the number of $\mathbf{x}_j$ that are y. For any $\vec{z} \in \mathbb{N}_t$, define the state

$$|\mathrm{Sym}_t^{\vec{z}}\rangle = \sqrt{\frac{1}{\binom{t}{\vec{z}}}} \sum_{x \in \mathcal{X}^t : k(x) = \vec{z}} |x\rangle. \tag{13}$$

For $\vec{z} \in \mathbb{N}_t^A$, $|\mathrm{Sym}_t^{\vec{z}}\rangle$ can be written as $\sqrt{\frac{1}{\binom{t}{\vec{z}}}} \sum_{x \in A^t : k(x) = \vec{z}} |x\rangle$.

The set of states

$$\left\{|\mathrm{Sym}_t^{\vec{z}}\rangle\right\}_{\vec{z} \in \mathbb{N}_t^A}, \left\{|\mathrm{Sym}_t^{\vec{z}}\rangle\right\}_{\vec{z} \in \mathbb{N}_t} \tag{14}$$

forms an orthonormal basis of the symmetric subspace $\vee^t \mathcal{H}_A$ and $\vee^t \mathcal{H}_n$, respectively. This implies that the dimension of the symmetric subspace $\vee^t \mathcal{H}_A$ is $|\mathbb{N}_t^A| = \binom{|A|+t-1}{t}$. In particular,

$$\dim\left(\vee^t \mathcal{H}_n\right) = |\mathbb{N}_t^{\mathcal{X}}| = \binom{N+t-1}{t}. \tag{15}$$

Let Π_t^{Sym} be the projection onto the symmetric subspace $\vee^t \mathcal{H}_n$, and for any $A \subset \mathcal{X}$, let $\Pi_t^{\mathrm{Sym},A}$ be the orthogonal projection onto $\vee^t \mathcal{H}_A$.

Let $\mu_{\mathcal{H}_n}$ be the Haar measure on $\mathcal{H}_n$, and $\mu_{\mathcal{H}_A}$ be the induced measure on $\mathcal{H}_A$, we have

$$\int (|\psi\rangle\langle\psi|)^{\otimes t} d\mu_{\mathcal{H}_n}(\psi) = \binom{N+t-1}{t}^{-1} \Pi_t^{\mathrm{Sym}} = \rho_t^{\mathrm{Sym}} = \binom{N+t-1}{t}^{-1} \sum_{\vec{z}} |\mathrm{Sym}_t^{\vec{z}}\rangle\langle\mathrm{Sym}_t^{\vec{z}}|. \tag{16}$$

$$\int (|\psi\rangle\langle\psi|)^{\otimes t} d\mu_{\mathcal{H}_A}(\psi) = \binom{|A|+t-1}{t}^{-1} \Pi_t^{\mathrm{Sym},A} = \rho_t^{\mathrm{Sym},A} \binom{N+t-1}{t}^{-1} \sum_{\vec{z} \in \mathbb{N}_t^A} |\mathrm{Sym}_t^{\vec{z}}\rangle\langle\mathrm{Sym}_t^{\vec{z}}|. \tag{17}$$

B Proof of the lemmas in Section 2.3

Proof of Lemma 1. Suppose not and there exists $a \in \{0,1\}^c$, such that

$$\Pr[K \overset{u}{\leftarrow} \{0,1\}^w; \mathcal{D}estruct(\mathcal{G}en(K)) = a] = s_a$$

is non-negligible.

Note that, in the $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{PRSPD}}$ game (see Algorithm 1), $a = \mathcal{Destruct}(\mathcal{Gen}(k))$ with probability s_a. By the correctness guarantee, for any arbitrary fixed $k \in \{0,1\}^w$,

$$\Pr[\mathcal{Destruct}(\mathcal{Gen}(k)) = a] \neq 0 \implies \Pr[\mathcal{Ver}(k,a) = 1] = 1.$$

Hence, for any arbitrary fixed $k \in \{0,1\}^w$,

$$\Pr[\mathcal{Ver}(k,a) = 1] \geq \Pr[\mathcal{Destruct}(\mathcal{Gen}(k)) = a]. \tag{18}$$

Therefore,

$$\Pr[K \xleftarrow{u} \{0,1\}^w : \mathcal{Ver}(K,a) = 1] = \sum_{k\in\{0,1\}^w} \frac{1}{2^w} \Pr[\mathcal{Ver}(k,a) = 1] \tag{19}$$

$$\geq \sum_{k\in\{0,1\}^w} \frac{1}{2^w} \Pr[\mathcal{Destruct}(\mathcal{Gen}(k)) = a] \qquad \text{By Eq. (18)} \tag{20}$$

$$= \Pr[K \xleftarrow{u} \{0,1\}^w ; \mathcal{Destruct}(\mathcal{Gen}(K)) = a] = s_a. \tag{21}$$

We will construct an efficient algorithm $\mathcal{A}$ against the $\mathsf{Unforgeability\text{-}of\text{-}proofs}$ property. $\mathcal{A}$ takes zero copies of the PRSPD state. She samples a key $\widetilde{K}$ uniformly at random and performs $\mathcal{Destruct}$ on it, getting the classical result F. She submits an alleged proof F. Let K be the actual key used by the challenger.

With probability s_a, F, takes the value a. Since the event $F = a$ and $\mathcal{Ver}(K,a) = 1$ are independent,

$$\Pr[\mathcal{Ver}(K,F) = 1] \geq \Pr[\mathcal{Ver}(K,a) = 1 \wedge F = a] = \Pr[\mathcal{Ver}(K,a) = 1] \Pr[F = a] \geq s_a \cdot s_a = {s_a}^2,$$

which is non-negligible since s_a is non-negligible, giving us a contradiction to $\mathsf{Unforgeability\text{-}of\text{-}proofs}$ (see Definition 1).

For the next part, we use pseudorandomness of PRSPD (see Definition 1).

Suppose there exists $a \in \{0,1\}^c$, such that $\Pr[|\phi\rangle \sim \mu_{\mathcal{H}_n} : \mathcal{Destruct}(|\phi\rangle) = a] = \widetilde{s}_a$ is non-negligible.

We will construct an efficient distinguisher $\mathcal{B}$ as follows. $\mathcal{B}$ samples a state 1-design for n-qubit states and performs $\mathcal{Destruct}$ to get a string $\widetilde{F}$. Hence with probability $\widetilde{s}_a$, $\widetilde{F} = a$. Next given a challenge n-qubit state $|\phi\rangle$, she performs $\mathcal{Destruct}$ on it to get $E \leftarrow \mathcal{Destruct}(|\phi\rangle)$ and checks if $E = \widetilde{F}$. Note that if $|\phi\rangle$ were Haar random, $\Pr[E = a] = \widetilde{s}_a$, whereas if $|\phi\rangle$ were from the PRSPD family, then $\Pr[E = a] = \mathsf{negl}(\lambda)$ for some negligible function, by the first part of the lemma.

Since the event $\widetilde{F} = a$ is independent of the distribution on $|\phi\rangle$, we conclude that $\mathcal{B}$ distinguishes with probability at least

$$\Pr[\widetilde{F} = a](\widetilde{s_a} - \mathsf{negl}(\lambda)) = \widetilde{s}_a(\widetilde{s}_a - \mathsf{negl}(\lambda)),$$

which is non-negligible if $\widetilde{s}_a$ is non-negligible, contradicting the pseudorandomness of PRSPD (see Definition 1). $\square$

Proof of Lemma 2. We will view the $\mathcal{D}\mathit{estruct}$ algorithm as applying a measurement on n-qubit PRSPD state and m-qubit ancilla initialized to $|0^m\rangle$. Let X denote the entire $m{+}n$-qubit system. By Naimark's theorem (see [Wat18, Theorem 2.42, Corollary 2.43]), this is equivalent to adding an output register Y with c-qubits initialized to 0 and then applying a projective measurement of the form:

$$\{B_a\}_{a\in\{0,1\}^c} \equiv \{U^\dagger(\mathbb{1}_{\mathcal{X}} \otimes |a\rangle\langle a|_{\mathcal{Y}})U\}_{a\in\{0,1\}^c},$$

for some unitary operator $U \in \mathrm{U}(\mathcal{X}\otimes\mathcal{Y})$, where $\mathcal{X}$ and $\mathcal{Y}$ denotes the Hilbert space corresponding to registers X and Y respectively. Hence $\{B_a\}_{a\in\{0,1\}^c}$ are set of orthonormal projectors and hence for every a, there exists orthonormal vectors $\{|\psi_j^a\rangle\}_{j\in[2^{n+m}]}$ such that

$$B_a = \sum_{j\in[2^{n+m}]} U^\dagger(|j\rangle\langle j|_{\mathcal{X}} \otimes |a\rangle\langle a|_{\mathcal{Y}})U = \sum_{j\in[2^{n+m}]} |\psi_j^a\rangle\langle\psi_j^a|,$$

where for every $j \in [2^{n+m}]$, $|\psi_j^a\rangle = U^\dagger|j\rangle_{\mathcal{X}} \otimes |a\rangle_{\mathcal{Y}}$. Note that $f_1, \ldots, f_t \sim \textsf{Correlated-Destruction}_t^{\mathcal{H}\mathit{aar},\mathcal{D}\mathit{estruct}}$ is obtained by performing a measurement $\{\otimes_{i\in[t]}B_{a_i}\}_{a_1,\ldots,a_t}$ on $|\phi\rangle = (|0^{c+m}\rangle \otimes |\psi\rangle)^{\otimes t}$, for a Haar random state $|\psi\rangle$.

Therefore, for any $a_1, \ldots, a_t$,

$$\begin{aligned}
&\Pr_{\textsf{Correlated-Destruction}_t^{\mathcal{H}\mathit{aar},\mathcal{D}\mathit{estruct}}}[f_1, \ldots, f_t = a_1, \ldots, a_t] && (22)\\
&= \int \mathrm{Tr}\left(\otimes_{i\in[t]}B_{a_i}\left(|0^{c+m}\rangle\langle 0^{c+m}| \otimes |\psi\rangle\langle\psi|\right)^{\otimes t}\right) d\mu_{\mathcal{H}}(\psi) && (23)\\
&= \int \mathrm{Tr}\left(\otimes_{i\in[t]}B_{a_i}\left(|\psi_A\rangle\langle\psi_A|\right)^{\otimes t}\right) d\mu_{\mathcal{H}_A}(\psi), && (24)\\
& && (25)
\end{aligned}$$

where $A = \{0^{c+m}\|x \mid x \in \{0,1\}^n\}$. Therefore $|A| = 2^n = N$.

By the additivity of trace, and Eq. (17), the above is

$$\begin{aligned}
&\Pr_{\textsf{Correlated-Destruction}_t^{\mathcal{H}\mathit{aar},\mathcal{D}\mathit{estruct}}}[f_1, \ldots, f_t = a_1, \ldots, a_t] && (26)\\
&= \int \mathrm{Tr}\left(\otimes_{i\in[t]}B_{a_i}\left(|\psi_A\rangle\langle\psi_A|\right)^{\otimes t}\right) d\mu_{\mathcal{H}_A}(\psi) && (27)\\
&= \mathrm{Tr}\left(\otimes_{i\in[t]}B_{a_i}\left(\binom{N+t-1}{t}^{-1}\Pi_t^{\mathrm{Sym},A}\right)\right). && (28)\\
& && (29)
\end{aligned}$$

Next, we use the following formulation of $\Pi_t^{\mathrm{Sym},A}$, (see [Har13, Eq. 2], and for a proof, see [Har13, Proposition 1]):

$$\Pi_t^{\mathrm{Sym},A} = \frac{1}{t!} \sum_{\sigma \in S_t} \sum_{x_1,\ldots,x_t \in A} |x_{\sigma^{-1}(1)} \ldots x_{\sigma^{-1}(t)}\rangle\langle x_1, \ldots, x_t|.$$

We will use Perm_τ to denote the unitary that permutes the registers as per τ for any $\tau \in S_t$, the group of permutations over t objects[12], i.e., for $B \equiv \{0,1\}^{c+m+n}$, Hence[13],

$$\mathsf{Perm}_\tau = \sum_{x_1,\ldots,x_t \in B} |x_{\tau(1)} \ldots x_{\tau(t)}\rangle\langle x_1 \ldots x_t| = \sum_{z_1,\ldots,z_t \in B} |z_1 \ldots z_t\rangle\langle z_{\tau^{-1}(1)} \ldots z_{\tau^{-1}(t)}|. \tag{30}$$

Let $\tilde{N} = 2^{m+n}$ in the following equations. Continuing,

$$\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{H}aar,\mathcal{D}estruct}}[f_1, \ldots, f_t = a_1, \ldots, a_t] \tag{31}$$

$$= \binom{N+t-1}{t}^{-1} \frac{1}{t!} \sum_{\sigma \in S_t} \sum_{x_1,\ldots,x_t \in A} \mathrm{Tr}\left(\left(\otimes_{i \in [t]} B_{a_i}\right) |x_{\sigma^{-1}(1)} \ldots x_{\sigma^{-1}(t)}\rangle\langle x_1, \ldots, x_t|\right) \tag{32}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{x_1,\ldots,x_t \in A} \mathrm{Tr}\left(\left(\otimes_{i \in [t]} B_{a_i}\right) \mathsf{Perm}_{\sigma^{-1}} |x_1, \ldots, x_t\rangle\langle x_1, \ldots, x_t|\right) \tag{33}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{x_1,\ldots,x_t \in A} \mathrm{Tr}\left(\left(\otimes_{i \in [t]} \left(\sum_{j_i \in [\tilde{N}]} |\psi_{j_i}^{a_i}\rangle\langle\psi_{j_i}^{a_i}|\right)\right) \mathsf{Perm}_{\sigma^{-1}} \left(|x_1, \ldots, x_t\rangle\langle x_1, \ldots, x_t|\right)\right) \tag{34}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{x_1,\ldots,x_t \in A} \mathrm{Tr}\left(\left(\sum_{j_1,\ldots,j_t \in [\tilde{N}]} \left(\otimes_{i \in [t]} |\psi_{j_i}^{a_i}\rangle\langle\psi_{j_i}^{a_i}|\right)\right) \mathsf{Perm}_{\sigma^{-1}} |x_1, \ldots, x_t\rangle\langle x_1, \ldots, x_t|\right) \tag{35}$$

$$\tag{36}$$

Note that, for any permutation $\sigma \in S_t$,

$$\mathsf{Perm}_\sigma^\dagger = (\mathsf{Perm}_\sigma)^{-1} = \mathsf{Perm}_{\sigma^{-1}}.$$

[12] The quantum operation of permuting registers is unitary because it can be realized by the composition of SWAP gates on pairs of registers. This is because any permutation can be written as a composition of transpositions, and permuting registers as per a transposition corresponds to applying a SWAP gate on two particular registers.

[13] The equation follows from the fact that permuting the registers maps the basis vector $|x_1, \ldots, x_t\rangle \mapsto |x_{\tau(1)}, \ldots, x_{\tau(t)}\rangle$.

Hence, for any set of pure states $\{|\alpha_j\rangle\}_{j\in[t]}$ and $\{|\beta_j\rangle\}_{j\in[t]}$, and permutation $\sigma \in S_t$,

$$\begin{aligned}
&\otimes_{j\in[t]} \left(|\alpha_j\rangle\langle\beta_j|\right) \mathsf{Perm}_{\sigma^{-1}} && (37)\\
&= \left(\otimes_{j\in[t]}|\alpha_j\rangle\right) \left(\otimes_{j\in[t]}\langle\beta_j|\right) \mathsf{Perm}_{\sigma^{-1}} && (38)\\
&= \left(\otimes_{j\in[t]}|\alpha_j\rangle\right) \left(\otimes_{j\in[t]}\langle\beta_j|\right) \mathsf{Perm}_{\sigma}^{\dagger} && (39)\\
&= \left(\otimes_{j\in[t]}|\alpha_j\rangle\right) \left(\otimes_{j\in[t]}\langle\beta_{\sigma(j)}|\right) = \otimes_{j\in[t]} \left(|\alpha_j\rangle\langle\beta_{\sigma(j)}|\right). && (40)
\end{aligned}$$

Hence, continuing we get,

$$\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}}[f_1,\ldots,f_t = a_1,\ldots,a_t] \tag{41}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma\in S_t} \sum_{x_1,\ldots,x_t\in A} \mathrm{Tr}\left(\left(\sum_{j_1,\ldots,j_t\in[\tilde{N}]} \left(\otimes_{i\in[t]}|\psi_{j_i}^{a_i}\rangle\langle\psi_{j_i}^{a_i}|\right)\right) \mathsf{Perm}_{\sigma^{-1}}|x_1,\ldots,x_t\rangle\langle x_1,\ldots,x_t|\right) \tag{42}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma\in S_t} \sum_{x_1,\ldots,x_t\in A} \mathrm{Tr}\left(\left(\sum_{j_1,\ldots,j_t\in[\tilde{N}]} \left(\otimes_{i\in[t]}|\psi_{j_i}^{a_i}\rangle\langle\psi_{j_{\sigma(i)}}^{a_{\sigma(i)}}|\right)\right) |x_1,\ldots,x_t\rangle\langle x_1,\ldots,x_t|\right) \tag{43}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma\in S_t} \mathrm{Tr}\left(\left(\sum_{j_1,\ldots,j_t\in[\tilde{N}]} \left(\otimes_{i\in[t]}|\psi_{j_i}^{a_i}\rangle\langle\psi_{j_{\sigma(i)}}^{a_{\sigma(i)}}|\right)\right) \sum_{x_1,\ldots,x_t\in A}|x_1,\ldots,x_t\rangle\langle x_1,\ldots,x_t|\right) \tag{44}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma\in S_t} \mathrm{Tr}\left(\left(\sum_{j_1,\ldots,j_t\in[\tilde{N}]} \left(\otimes_{i\in[t]}|\psi_{j_i}^{a_i}\rangle\langle\psi_{j_{\sigma(i)}}^{a_{\sigma(i)}}|\right)\right) \left(\otimes_{i\in[t]}|0^{c+m}\rangle\langle 0^{c+m}| \otimes I_N\right)\right) \tag{45}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma\in S_t} \sum_{j_1,\ldots,j_t\in[\tilde{N}]} \mathrm{Tr}\left(\otimes_{i\in[t]} \left(|\psi_{j_i}^{a_i}\rangle\langle\psi_{j_{\sigma(i)}}^{a_{\sigma(i)}}| \cdot \left(|0^{c+m}\rangle\langle 0^{c+m}| \otimes I_N\right)\right)\right) \tag{46}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma\in S_t} \sum_{j_1,\ldots,j_t\in[\tilde{N}]} \mathrm{Tr}\left(\otimes_{i\in[t]} \left(\left(\langle 0^{c+m}| \otimes I_N\right) |\psi_{j_i}^{a_i}\rangle\langle\psi_{j_{\sigma(i)}}^{a_{\sigma(i)}}| \left(|0^{c+m}\rangle \otimes I_N\right)\right)\right). \tag{47}$$

Let $W_i = \left(\langle 0^{c+m}| \otimes I_N\right)|\psi_{j_i}^{a_i}\rangle$ for every $i \in [t]$. Continuing,

$$\Pr_{\textsf{Correlated-Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}}[f_1, \ldots, f_t = a_1, \ldots, a_t] \tag{48}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{j_1,\ldots,j_t \in [\tilde{N}]} \mathrm{Tr}\left(\otimes_{i\in[t]}\left(\left(\langle 0^{c+m}| \otimes I_N\right)|\psi_{j_i}^{a_i}\rangle\langle\psi_{j_{\sigma(i)}}^{a_{\sigma(i)}}|\left(|0^{c+m}\rangle \otimes I_N\right)\right)\right) \tag{49}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{j_1,\ldots,j_t \in [\tilde{N}]} \mathrm{Tr}\left(\otimes_{i\in[t]}\left(W_i W_{\sigma(i)}^\dagger\right)\right) \tag{50}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{j_1,\ldots,j_t \in [\tilde{N}]} \prod_{i\in[t]} \left(\mathrm{Tr}\left(W_i W_{\sigma(i)}^\dagger\right)\right) \tag{51}$$

$$\leq \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{j_1,\ldots,j_t \in [\tilde{N}]} \prod_{i\in[t]} \left(\sqrt{\mathrm{Tr}\left(W_i W_i^\dagger\right)}\sqrt{\mathrm{Tr}\left(W_{\sigma(i)} W_{\sigma(i)}^\dagger\right)}\right). \tag{52}$$

The last inequality is due to Cauchy-Schwarz for the Hilbert-Schmidt inner product, i.e., for every two complex square matrices A, B,

$$\mathrm{Tr}(AB^\dagger) = \mathrm{Tr}(B^\dagger A) \leq \sqrt{\mathrm{Tr}(A^\dagger A)}\sqrt{\mathrm{Tr}(B^\dagger B)}.$$

Note that for every $\sigma \in S_t$,

$$\prod_{i\in[t]} \left(\sqrt{\mathrm{Tr}\left(W_i W_i^\dagger\right)}\sqrt{\mathrm{Tr}\left(W_{\sigma(i)} W_{\sigma(i)}^\dagger\right)}\right) = \prod_{i\in[t]} \mathrm{Tr}\left(W_i W_i^\dagger\right).$$

Continuing using this fact,

$$\Pr_{\textsf{Correlated-Destruction}_t^{\mathcal{Haar},\mathcal{Destruct}}}[f_1, \ldots, f_t = a_1, \ldots, a_t] \tag{53}$$

$$\leq \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{j_1,\ldots,j_t \in [\tilde{N}]} \prod_{i\in[t]} \left(\sqrt{\mathrm{Tr}\left(W_i W_i^\dagger\right)}\sqrt{\mathrm{Tr}\left(W_{\sigma(i)} W_{\sigma(i)}^\dagger\right)}\right) \tag{54}$$

$$= \frac{(N-1)!}{(N+t-1)!} \sum_{\sigma \in S_t} \sum_{j_1,\ldots,j_t \in [\tilde{N}]} \prod_{i\in[t]} \mathrm{Tr}\left(W_i W_i^\dagger\right) \tag{55}$$

$$= \frac{(N-1)!t!}{(N+t-1)!} \sum_{j_1,\ldots,j_t \in [\tilde{N}]} \prod_{i\in[t]} \mathrm{Tr}\left(W_i W_i^\dagger\right) \tag{56}$$

$$= \frac{1}{\binom{N+t-1}{t}} \prod_{i\in[t]} \left(\sum_{j_i \in [\tilde{N}]} \mathrm{Tr}\left(W_i W_i^\dagger\right)\right) \tag{57}$$

$$= \frac{1}{\binom{N+t-1}{t}} \prod_{i\in[t]} \left(\mathrm{Tr}\left(\sum_{j_i \in [\tilde{N}]} W_i W_i^\dagger\right)\right) \tag{58}$$

$$\tag{59}$$

Next, for every $a_1, \ldots, a_t$,

$$
\begin{aligned}
&\Pr_{\mathsf{Product\text{-}Destruction}_t^{\mathcal{H}aar,\mathcal{D}estruct}}[(f_1, \ldots, f_t) = (a_1, \ldots, a_t)] && (60)\\
&= \prod_{i\in[t]} \left[\int \mathrm{Tr}\left(B_{a_i}|\psi\rangle\langle\psi|\right) d\mu_{\mathcal{H}_A}(\psi)\right] && (61)\\
&= \prod_{i\in[t]} \left[\mathrm{Tr}\left(B_{a_i}\frac{|0^{c+m}\rangle\langle 0^{c+m}| \otimes Id_N}{N}\right)\right] && (62)\\
&= \frac{1}{N^t}\prod_{i\in[t]} \left[\mathrm{Tr}\left(B_{a_i}\left(|0^{c+m}\rangle\langle 0^{c+m}| \otimes Id_N\right)\right)\right] && (63)\\
&= \frac{1}{N^t}\prod_{i\in[t]} \mathrm{Tr}\left(\left(\sum_{j_i\in[\tilde{N}]} |\psi_{j_i}^{a_i}\rangle\langle\psi_{j_i}^{a_i}|\right)|0^{c+m}\rangle\langle 0^{c+m}| \otimes Id_N\right) && (64)\\
&= \frac{1}{N^t}\prod_{i\in[t]} \left(\mathrm{Tr}\left(\sum_{j_i\in[\tilde{N}]} \left(\left(\langle 0^{c+m} \otimes I_N|\right)|\psi_{j_i}^{a_i}\rangle\langle\psi_{j_{\sigma(i)}}^{a_{\sigma(i)}}|\left(|0^{c+m}\rangle \otimes I_N\right)\right)\right)\right) && (65)\\
&= \frac{1}{N^t}\prod_{i\in[t]} \left(\mathrm{Tr}\left(\sum_{j_i\in[\tilde{N}]} W_i W_i^{\dagger}\right)\right). && (66)
\end{aligned}
$$

Therefore, for every $a_1, \ldots, a_t$,

$$
\frac{\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathcal{H}aar,\mathcal{D}estruct}}[(f_1, \ldots, f_t) = (a_1, \ldots, a_t)]}{\Pr_{\mathsf{Product\text{-}Destruction}_t^{\mathcal{H}aar,\mathcal{D}estruct}}[(f_1, \ldots, f_t) = (a_1, \ldots, a_t)]}
\leq \frac{\frac{1}{\binom{N+t-1}{t}}\prod_{i\in[t]}\left(\mathrm{Tr}\left(\sum_{j_i\in[\tilde{N}]} W_i W_i^{\dagger}\right)\right)}{\frac{1}{N^t}\prod_{i\in[t]}\left(\mathrm{Tr}\left(\sum_{j_i\in[\tilde{N}]} W_i W_i^{\dagger}\right)\right)} = \frac{N^t}{\binom{N+t-1}{t}}.
$$

□

Proof of Lemma 3. By Lemma 1, for every $a \in \{0,1\}^c$, there exists a negligible function $\mathsf{negl}(\lambda)_a$ such that

$$\Pr_F[f = a] = \mathsf{negl}(\lambda)_a.$$

Hence, there exists a negligible function $\mathsf{negl}(\lambda)$, such that

$$\left[\max_a \Pr_{f\sim F}[f = a])\right] = \mathsf{negl}(\lambda).$$

Therefore the probability of seeing a colliding outcome, i.e.,

$$\Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}}[\mathsf{Collision}] \equiv \Pr_{\mathsf{Correlated\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}}[\exists i \neq j \mid f_i = f_j] \tag{67}$$

$$\leq \sum_{i \neq j} \Pr_{(f_1,\ldots,f_t)\sim\mathsf{Correlated\text{-}Destruction}_t^{\mathscr{Haar},\mathscr{Destruct}}}[f_i = f_j] \tag{68}$$

$$= \sum_{i \neq j} \Pr_{(f_i,f_j)\sim\mathsf{Correlated\text{-}Destruction}_2^{\mathscr{Haar},\mathscr{Destruct}}}[f_i = f_j] \tag{69}$$

$$= \sum_{i \neq j} \sum_{a \in \{0,1\}^w} \Pr_{(f_i,f_j)\sim\mathsf{Correlated\text{-}Destruction}_2^{\mathscr{Haar},\mathscr{Destruct}}}[f_i = a, f_j = a] \tag{70}$$

$$\leq \sum_{i \neq j} \sum_{a \in \{0,1\}^w} \frac{N^2}{\binom{N}{2}} \Pr_{(f_i,f_j)\sim\mathsf{Product\text{-}Destruction}_2^{\mathscr{Haar},\mathscr{Destruct}}}[f_i = a, f_j = a] \qquad \text{By Lemma 2} \tag{71}$$

$$= \sum_{i \neq j} \sum_{a \in \{0,1\}^w} \frac{N^2}{\binom{N}{2}} \left(\Pr_{f\sim F}[f = a] \right)^2 \tag{72}$$

$$\leq \frac{N^2}{\binom{N+1}{2}} \sum_{i \neq j} \sum_{a \in \{0,1\}^c} (\Pr_{f\sim F}[f = a]) \left[\max_a \Pr_{f\sim F}[f = a]) \right] \tag{73}$$

$$= \frac{N^2}{\binom{N+1}{2}} \sum_{i \neq j} \left[\max_a \Pr_{f\sim F}[f = a]) \right] \tag{74}$$

$$= \frac{N^2}{\binom{N+1}{2}} \binom{t}{2} \left[\max_a \Pr_{f\sim F}[f = a]) \right] \tag{75}$$

$$\leq \frac{N^2}{\binom{N+1}{2}} \binom{t}{2} \mathsf{negl}(\lambda) \tag{76}$$

$$= \frac{N(t(t-1))}{N+1} \mathsf{negl}(\lambda), \tag{77}$$

$$\tag{78}$$

which is negligible for any $t \in \mathsf{poly}(\lambda)$. □

Proof of Lemma 5. Suppose there exists an adversary $\mathcal{A}$ in $\mathsf{Classical\text{-}Forging\text{-}Exp}_\lambda^{;\mathsf{PRFSPD}}$. We will construct an adversary $\mathcal{B}$ in the game $\mathsf{Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathsf{PRFSPD}}$ such that $\mathcal{A}$ and $\mathcal{B}$ have the same success probability up to negligible factor in the corresponding security games. $\mathcal{B}$ runs $\mathcal{A}$ and for every query x to $\mathscr{Destruct}(\mathscr{Gen}(k, \cdot))$, $\mathcal{B}$ makes a query x to $\mathscr{Gen}(k, \cdot)$, and then performs $\mathscr{Destruct}$ on the oracle output to get a classical string σ_x and feeds it to $\mathcal{A}$ as the output of her query. If x was never queried before, $\mathcal{B}$ creates a set S_x and adds σ_x to S_x. Otherwise, $\mathcal{B}$ checks if $\sigma_x \notin S_x$ in which case she adds σ_x to S_x, but if $\sigma_x \in S_x$, $\mathcal{B}$ aborts. $\mathcal{B}$ answers $\mathscr{Ver}(k, \cdot)$ queries using the $\mathscr{Ver}(k, \cdot)$ oracle she has access to. At the end when $\mathcal{A}$ outputs a $x', \sigma_{x'}$, $\mathcal{B}$ sends $x, \tilde{S}_{x'}$ to the challenger, where $\tilde{S}_{x'} = S_{x'} \cup \{\sigma_{x'}\}$ if $S_{x'}$ exists, otherwise $\tilde{S}_{x'} = \{\sigma_{x'}\}$.

Let $\mathsf{Win}_{\mathcal{B}}$ and $\mathsf{Win}_{\mathcal{A}}$ be the events $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{B},\mathsf{PRFSPD}} = 1$ and $\mathsf{Classical\text{-}Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{PRFSPD}} = 1$ respectively, and $Abort_{\mathcal{B}}$ be the event that $\mathcal{B}$ aborts. We will show that there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$|Pr[\mathsf{Win}_{\mathcal{B}}] \geq \Pr[\mathsf{Win}_{\mathcal{A}}] - \mathsf{negl}(\lambda).$$

Note that $\mathcal{B}$ aborts only if there is an x, the PRFSPD state corresponding to which yields a collision in the proofs upon destruction. For every $x \in \{0,1\}^d$, let $\mathsf{Collision}_{\mathcal{B}}^{x}$ be the event that there exist distinct indices $q, \tilde{q}$ such that the q^{th} and the $\tilde{q}^{th}$ queries made by $\mathcal{A}$ were for x, and she received the same output for both the queries. By Lemma 4, for every x, there exists a negligible function $\mathsf{negl}(\lambda)_x$ such that $\Pr[\mathsf{Collision}_{\mathcal{B}}^{x}] = \mathsf{negl}(\lambda)_x$. Let Q be the set of all $x \in \{0,1\}^d$ queried by $\mathcal{A}$ to the $\mathcal{D}estruct(\mathcal{G}en(\cdot))$ oracle. Since there were only polynomially many $x \in Q$,

$$\Pr[Abort_{\mathcal{B}}] = \Pr[\bigcup_{x \in Q} \mathsf{Collision}_{\mathcal{B}}^{x}] \leq \sum_{x \in Q} \mathsf{negl}(\lambda)_x = \mathsf{negl}(\lambda),$$

where $\mathsf{negl}(\lambda)$ is some negligible function.

Let $\mathsf{Collision}_{\mathcal{A}}$ be the analogous event of $Abort_{\mathcal{B}}$ in the actual game, i.e., $\mathsf{Classical\text{-}Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{PRFSPD}}$, meaning $\mathsf{Collision}_{\mathcal{A}}$ is the event that there exists x and there exists the distinct indices $q, \tilde{q}$ such that the q^{th} and the $\tilde{q}^{th}$ queries made by $\mathcal{A}$ were for x and she received the same output for both the queries. Hence,

$$\Pr[\mathsf{Collision}_{\mathcal{A}}] = \Pr[Abort_{\mathcal{B}}] = \mathsf{negl}(\lambda). \tag{79}$$

Let $X', \sigma_{X'}$ be the random variable representing the final output of $\mathcal{A}$ in the simulation. Let $\mathsf{Good}_{\mathcal{B}}$ be the event that either $S_{X'}$ did not exist or $\sigma'_X \notin S_{X'}$, (i.e., $(X', \sigma_{X'})$ does not correspond to any past query) and $(X', \sigma_{X'})$ passes verification. Conditioned on the event $\overline{Abort_{\mathcal{B}}}$, $\mathsf{Good}_{\mathcal{B}}$ implies that $\mathcal{B}$ submits $X', \tilde{S}_{X'}$ at the end, such that all the elements in the set $\tilde{S}_{X'}$ would pass verification with respect to X', and $|\tilde{S}_{X'}|$ is strictly larger than the number of X' queries made to $\mathcal{G}en()$ by $\mathcal{B}$, thus implying $\mathsf{Win}_{\mathcal{B}}$. Hence,

$$\Pr[\mathsf{Good}_{\mathcal{B}} \wedge \overline{Abort_{\mathcal{B}}}] \leq \Pr[Win_{\mathcal{B}} \wedge \overline{Abort_{\mathcal{B}}}.$$

Note that $\mathsf{Good}_{\mathcal{B}}$ corresponds to the event $Win_{\mathcal{A}}$ in the actual game, and as mentioned in Eq. (79), $\overline{Abort_{\mathcal{B}}}$ corresponds to $\overline{\mathsf{Collision}_{\mathcal{A}}}$ (see Eq. (79)). Hence,

$$\Pr[Win_{\mathcal{A}} \wedge \overline{\mathsf{Collision}_{\mathcal{A}}}] = \Pr[\mathsf{Good}_{\mathcal{B}} \wedge \overline{Abort_{\mathcal{B}}}] \leq \Pr[Win_{\mathcal{B}} \wedge \overline{Abort_{\mathcal{B}}}]. \tag{80}$$

Therefore,

$$\begin{aligned}
\Pr[Win_{\mathcal{B}}] &\geq \Pr[Win_{\mathcal{B}} \wedge \overline{Abort_{\mathcal{B}}}] && (81)\\
&\geq \Pr[Win_{\mathcal{A}} \wedge \overline{\mathsf{Collision}}] \qquad \text{By Eq. (80).} && (82)\\
&\geq \Pr[\mathsf{Win}_{\mathcal{A}}] - \Pr[\mathsf{Collision}_{\mathcal{A}}] && (83)\\
&= \Pr[\mathsf{Win}_{\mathcal{A}}] - \mathsf{negl}(\lambda). \qquad \text{By Eq. (79).} && (84)
\end{aligned}$$

□

C Proof of the technical lemma from Section 3

Proof of Lemma 6. Assume toward contradiction there exists a polynomial T and a quantum polynomial-time adversary $\mathcal{A}$ which violate the lemma, let ε the advantage that the adversary gets (i.e. the distribution on its output bit b in the setting D_0 has non-negligible statistical distance ε from its output bit b in the setting D_1). Recall that for every possible input $x \in \{0,1\}^d$, $\mathcal{A}$ makes either 0 or exactly T queries to the oracle and consider the following hybrid distributions over the output bit of the adversary at the end of the process.

- Hyb_0 : The original distribution D_0. For a uniformly random $k \leftarrow \{0,1\}^\lambda$, the adversary $\mathcal{A}$ gets classical oracle access to the generation function $\mathcal{G}en(k,\cdot)$ (from Fig. 2) and the classical proof verification function $\mathcal{V}er(k,\cdot,\cdot)$ (from Fig. 3). Recall that the inputs to $\mathcal{G}en(k,\cdot)$ are of the form $x \in \{0,1\}^d$, and the inputs to $\mathcal{V}er(k,\cdot,\cdot)$ are of the form $(x,q) \in \left(\{0,1\}^d \times \{0,1\}^{5\lambda}\right)$.
- Hyb_1 : Moving to a truly random permutation. Identical to the previous hybrid, with the only change that the PRP PRP_k (in both $\mathcal{G}en(k,\cdot)$, $\mathcal{V}er(k,\cdot,\cdot)$) is swapped with a uniformly random permutation $P \leftarrow S_{2^{5\lambda}}$ on the set $\{0,1\}^{5\lambda}$.
- Hyb_2 : Moving to random disjoint sets. In this hybrid we discard the permutation P and execute the following: The process starts with sampling uniformly random subsets of $\{0,1\}^{5\lambda}$, $\{A_x, B_x\}_{x\in\{0,1\}^d}$, each of size 2^λ, conditioned on that all $2 \cdot 2^d$ sets are *disjoint from one another.*
  - When the adversary applies a query $x \in \{0,1\}^d$ to $\mathcal{G}en(\cdot)$, the output is the 5λ-qubit state:
  $$2^{-\frac{\lambda}{2}} \cdot \sum_{j\in\{0,1\}^\lambda} \omega_{2^{5\lambda}}^{b_{x,j}} \cdot |a_{x,j}\rangle \enspace ,$$
  where the elements in the sets A_x, B_x are denoted by $A_x := \{a_{x,j}\}_{j\in\{0,1\}^\lambda}$, $B_x := \{b_{x,j}\}_{j\in\{0,1\}^\lambda}$, respectively.

  - When the adversary applies a query (x, q) to $\mathcal{Ver}(\cdot, \cdot)$, the output is 1 iff $q \in A_x$.

- Hyb_3 : Moving to random i.i.d sets. This process is identical to the previous, with one change: All $2 \cdot 2^d$ sets $\{A_x, B_x\}_{x \in \{0,1\}^d}$ are sampled as uniformly random i.i.d. subsets of $\{0,1\}^{5\lambda}$ of size 2^λ (we remove the condition of the sets being disjoint).

- Hyb_4 : Defining the classical function f and moving to the small-range distribution of it. Consider both, the generation and verification oracles which are both inefficient quantum algorithms at this current hybrid - both of them use as a black-box, quantumly-queriable classical oracle f, which for input $|x, y\rangle$ (for $x \in \{0,1\}^d$, $y \in \{0,1\}^{|(A_x, B_x)|}$), outputs $|x, y \oplus (A_x, B_x)\rangle$, where (A_x, B_x) is the full classical description of the sets A_x, B_x. Let us call this function f, and sampling the random sets A_x, B_x (for all $x \in \{0,1\}^d$) only comes down to sampling the outputs of f.

  Now, for the difference between the previous hybrid and the current one: The only difference is that we move from a distribution that for every $x \in \{0,1\}^d$, instead of applying $f(\cdot)$ to get (A_x, B_x), it applies $f(\cdot)$ to the small-range distribution version of these functions, for the parameter $r := \frac{300 \cdot t(\mathcal{A})^3 \cdot 4}{\varepsilon}$, where $t(\mathcal{A})$ is the (polynomial) running time of $\mathcal{A}$. Concretely: The current process starts with sampling r random elements $z_1, \cdots z_r$ in $\{0,1\}^d$ followed by sampling a random $i_x \leftarrow [r]$ for every $x \in \{0,1\}^d$. During the execution of the hybrid, given a query $x \in \{0,1\}^d$ to $f(\cdot)$, we output $f(z_{i_x})$.

- Hyb_5 : Using fewer sets for the generation of states. In this hybrid we stop thinking about getting the description of our sets through f, and just sample the sets at the beginning of the process. We will get back to describing our process using the function f later. Observe that in the previous hybrid process, we used only the $2 \cdot r$ sets $\{A_{z_i}, B_{z_i}\}_{i \in [r]}$ rather than all of the $2 \cdot 2^d$ sets $\{A_x, B_x\}_{x \in \{0,1\}^d}$ the process sampled. In this process, instead of sampling all of the $2 \cdot 2^d$ sets $\{A_x, B_x\}_{x \in \{0,1\}^d}$, we sample only $2 \cdot r$ sets: $\{A_{z_i}, B_{z_i}\}_{i \in [r]}$, and the process carries on as in the previous.

- Hyb_6 : Using fewer elements from each set, for the generation of states. In this process we still sample the sets $\{A_{z_i}, B_{z_i}\}_{i \in [r]}$, but we don't generate superpositions of them. At the beginning of the process, we execute:

  1. For every $i \in [r]$, sample a uniformly random T-sized *multi-set* $(a_{i,1}, \cdots, a_{i,T})$ of A_{z_i} (uniformly random over all of the T-size multi-sets of A_{z_i}).

2. For every $i \in [r]$, we generate the $5\lambda \cdot T$-qubit state,

$$|\pi_i\rangle := \sum_{\sigma \in S_T} |a_{i,\sigma(1)}, \cdots, a_{i,\sigma(T)}\rangle \enspace ,$$

where S_T is the set of all permutations on T elements.

The proof verification algorithm $\mathcal{Ver}(\cdot, \cdot)$ stays the same. The state generation algorithm $\mathcal{Gen}(\cdot)$, given $x \in \{0,1\}^d$ for the c-th query (for $c \in [T]$), outputs the c-th sub-register of the state $|\pi_{i_x}\rangle$.

- Hyb_7 : Using fewer elements for the verification of classical proofs. Note two things: (1) In the previous hybrid we ignore the set B_{z_i} altogether, and (2) Regarding the set A_{z_i}, the only place where we use the *full* information of its elements is to verify proofs in the procedure $\mathcal{Ver}(\cdot, \cdot)$ (where we check if the given proof is in the set A_{z_i}). In the current hybrid, there are two changes: (1) We do not sample the set B_{z_i}, and (2) We do sample all 2^λ elements of the set A_{z_i}, but then for every $i \in [r]$, we sample a small set: a uniformly random T-sized multi-set $M_i = (a_{i,1}, \cdots, a_{i,T})$ of A_{z_i}. The generation algorithm $\mathcal{Gen}(\cdot)$ stays the same as before and the multi-set which it uses for the state generation is M_i, but the verification changes: given input $(x, q) \in \left(\{0,1\}^d \times \{0,1\}^{5\lambda}\right)$, the verification now just checks whether $q \in \{a_{i_x,1}, \cdots, a_{i_x,T}\}$.

- Hyb_8 : **Sampling** fewer elements for the verification of classical proofs. This hybrid is identical to the previous, only that for every $i \in [r]$, instead of sampling the entire, 2^λ-sized set A_{z_i} and then sampling the multi-set M_i, the multi-set M_i is just sampled as a uniformly random T-sized multi-set of $\{0,1\}^{5\lambda}$ (i.e. uniformly random overall T-sized multi-sets of $\{0,1\}^{5\lambda}$).

- Hyb_9 : Moving to the full-range distribution of the new functions. The only change between this process and the previous is that in this process, instead of sampling only r small multi-sets $\{M_i\}_{i \in [r]}$, we sample an i.i.d. uniformly random T-sized multi-set of $\{0,1\}^{5\lambda}$ for every $x \in \{0,1\}^d$: $\{M_x\}_{x \in \{0,1\}^d}$. The algorithms $\mathcal{Gen}(\cdot)$ and $\mathcal{Ver}(\cdot, \cdot)$ function the same as before, only that they now do not do the mapping from x to its small-range element z_{i_x}, that is: For every x the generation algorithm now generates the state,

$$|\pi_x\rangle := \sum_{\sigma \in S_T} |a_{x,\sigma(1)}, \cdots, a_{x,\sigma(T)}\rangle \enspace ,$$

and returns the c-th sub-register of $|\pi_x\rangle$ on the c-th query. The verification algorithm, given query (x, q) checks whether $q \in \{a_{x,1}, \cdots, a_{x,T}\}$. Note that this is exactly the distribution D_1.

We next prove a bound on the computational indistinguishability between the hybrids.

- $\mathsf{Hyb}_0 \approx_c \mathsf{Hyb}_1$: The hybrids are computationally indistinguishable by the security of the pseudorandom permutation PRP_k.

- $\mathsf{Hyb}_1 \equiv \mathsf{Hyb}_2$: One can observe the following property of a uniformly random permutation $P \leftarrow S_{2^{5\lambda}}$: for any set of disjoint sets $\{E_i\}_{i\in[m]}$ such that $\forall i \in [m] : E_i \subseteq \{0,1\}^{5\lambda}$, the set of sets $\{P(E_i)\}^{i\in[m]}$ is a set of uniformly random subsets of $\{0,1\}^{5\lambda}$, conditioned on that all subsets are disjoint to each other. One can also verify that this immediately implies that $\mathsf{Hyb}_1 \equiv \mathsf{Hyb}_2$.

- $\mathsf{Hyb}_2 \approx_s \mathsf{Hyb}_3$: Conditioned on the probabilistic event that all sets $\{A_x, B_x\}_{x\in\{0,1\}^d}$ are disjoint, the hybrids Hyb_2, Hyb_3 distribute exactly the same. Thus, it is sufficient to show that this probabilistic event happens with an overwhelming probability (or alternatively, as we will do, that its negation happens with a negligible probability). Since $d = \lambda$, we are sampling $2 \cdot 2^{\lambda}$ sets, each of size 2^{λ}, which amounts to $2^{2\lambda+1}$ elements. The size of the set we are sampling from is $2^{5\lambda}$, and thus by union bound the probability to have a repeating element is bounded by

$$2^{-5\lambda} \cdot \sum_{\ell\in[2^{2\lambda}]} \ell = 2^{-5\lambda} \cdot \frac{2^{4\lambda} - 2^{2\lambda}}{2} \le \frac{2^{4\lambda}}{2^{5\lambda}} = 2^{-\lambda} \quad ,$$

  which is negligible.

- $\mathsf{TD}(\mathsf{Hyb}_3, \mathsf{Hyb}_4) \le \frac{\varepsilon}{4}$: The only difference between Hyb_3 and Hyb_4 is how we query the function f: In Hyb_3, for every input $x \in \{0,1\}^d$ the output is (A_x, B_x), while in the next Hyb_4, for input x the output is $f(z_{i_x})$. In that sense, Hyb_4 produces the small-range distribution version of Hyb_3, and in Hyb_4 we set $r = \frac{300\cdot t(\mathcal{A})^3\cdot 4}{\varepsilon}$ where $t(\mathcal{A})$ is the running time of $\mathcal{A}$ and thus an upper bound on the number of (quantum) queries made by $\mathcal{A}$ to f. It follows by Theorem A.6 from [AGQY22], that the statistical distance between the two hybrid processes are bounded by $\frac{300\cdot t(\mathcal{A})^3}{r} = \frac{\varepsilon}{4}$.

- $\mathsf{Hyb}_4 \equiv \mathsf{Hyb}_5$: The change between these two processes is only semantic. The sets from $\{A_x, B_x\}_{x\in\{0,1\}^d}$ that are sampled but not included in $\{z_1, \cdots, z_r\}$, are never used - not in the generation nor the classical proof verification algorithm. The oracles are identical and so are the processes.

- $\mathsf{Hyb}_5 \approx_s \mathsf{Hyb}_6$: We can think of $r+1$ sub-hybrid processes $\mathsf{Hyb}_{5,0}, \mathsf{Hyb}_{5,1}, \cdots, \mathsf{Hyb}_{5,r}$, where $\mathsf{Hyb}_{5,0} = \mathsf{Hyb}_5$ and for each $i \in [r]$ we change $\mathcal{Gen}(z_i)$ from its behavior in Hyb_5 to Hyb_6. This also means that $\mathsf{Hyb}_{5,r} = \mathsf{Hyb}_6$. Lemma

8 implies that for each $i \in [r]$, $\mathsf{td}\left(\mathsf{Hyb}_{5,i-1}, \mathsf{Hyb}_{5,i}\right) \le \mathsf{negl}(\lambda)$ for a negligible function $\mathsf{negl}(\lambda)$[14]. It follows that the trace distance between Hyb_5 and Hyb_6 is bounded by $r \cdot \mathsf{negl}(\lambda)$, which is negligible in λ, because r is polynomial in λ.

- $\mathsf{Hyb}_6 \approx_s \mathsf{Hyb}_7$: The only difference between these two hybrid processes is that in Hyb_6, for an input (x, q) to the classical proof verification function $\mathcal{Ver}(\cdot, \cdot)$, the proof is accepted if $q \in A_{z_{i_x}}$, but in Hyb_7 the proof is accepted only if q is in the smaller, at-most-T-sized set $\{a_{i_x,1}, a_{i_x,2}, \cdots, a_{i_x,T}\}$. The point is, that in Hyb_6 the adversary has no information on the rest of the set $A_{z_{i_x}}$, i.e. it does not receive any information on $A_{z_{i_x}} \setminus \{a_{i_x,1}, \cdots, a_{i_x,T}\}$. The probability for $\mathcal{A}$ to output an element in this gap is thus bounded by $|A_{z_{i_x}}|/|\{0,1\}^{5\lambda}| = 2^{\lambda}/2^{5\lambda} = 2^{-4\lambda}$. If there would be a non-negligible advantage to distinguish these two processes it is necessarily the case that in one of the queries that the adversary sent, there is an element in $A_{z_{i_x}} \setminus \{a_{i_x,1}, \cdots, a_{i_x,T}\}$ with a non-negligible amplitude, for some $x \in \{0,1\}^d$. We could guess the position of that query with a noticeable probability and find such an element with a non-negligible probability, in contradiction to the fact that the probability is bounded by 2^{-4n}.

- $\mathsf{Hyb}_7 \equiv \mathsf{Hyb}_8$: For each $i \in [r]$, the distributions over the multi-set M_i are the same: In Hyb_7 we first sample uniformly at random A_{z_i}, a subset of $\{0,1\}^{5\lambda}$, and then M_i is a uniformly random T-sized multi-set of it. In Hyb_8, we just sample M_i as a uniformly random T-sized multi-set of $\{0,1\}^{5\lambda}$. These distributions over multi-sets have statistical distance 0 and thus the processes are equivalent.

- $\mathsf{TD}\left(\mathsf{Hyb}_8, \mathsf{Hyb}_9\right) \le \frac{\varepsilon}{4}$: One can verify that just like process Hyb_4 is the small-range distribution version of Hyb_3 (by thinking of a function f that outputs a classical description of the sets used to generate states), the process Hyb_8 is the small-range distribution of the process Hyb_9, by thinking of a new classical function f' that for input $x \in \{0,1\}^d$, outputs the classical description of the uniformly random T-sized multi-set M_x which blends in the following way: In the beginning of Hyb_8 we sample $z_1, \cdots, z_r \leftarrow \{0,1\}^d$, $\forall x \in \{0,1\}^d : i_x \leftarrow [r]$, and then define $f'(x) := M_{z_{i_x}}$, while in Hyb_9 we just output $f(x) := M_x$. By the exact same argument for the statistical closeness $\mathsf{TD}\left(\mathsf{Hyb}_3, \mathsf{Hyb}_4\right) \le \frac{\varepsilon}{4}$, the statistical closeness $\mathsf{TD}\left(\mathsf{Hyb}_8, \mathsf{Hyb}_9\right) \le \frac{\varepsilon}{4}$ follows.

[14] In the formulation of Lemma 8, the state's phases are given by a random function f and here they are given by a random set B. note that these distributions are the same up to no collisions in f: As long as there are no two elements in A that collide in f, the distributions are identical. Since f is a random function on 5λ bits, and A is of size 2^{λ}, this will happen with an exponentially small probability, thus Lemma 8 is applicable.

From the above statements it follows that the adversary $\mathcal{A}$ can distinguish between $\mathsf{Hyb}_0 := D_0$ and $\mathsf{Hyb}_9 := D_1$ with at most advantage $\frac{\varepsilon}{2} + \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\lambda)$. This is in contradiction to the assumption that the distinguishing advantage of $\mathcal{A}$ is ε, and that ε is non-negligible. □

Lemma 8. *Let $A \subseteq \{0,1\}^{5\lambda}$ a set of size 2^λ and a polynomial $T := T(\lambda)$. Then, the following two distributions on quantum states have trace distance negligible in λ.*

- Q_0 *: Sample a random function $f \leftarrow \left(\{0,1\}^{5\lambda}\right)^{\{0,1\}^{5\lambda}}$ and output T identical copies of the state*

$$|\psi_{f,A}\rangle := 2^{-\frac{\lambda}{2}} \sum_{a \in A} \omega_{2^{5\lambda}}^{f(a)} \cdot |a\rangle \ .$$

- Q_1 *: Sample a uniformly random T-sized multi-set of A: $(a_1, \cdots, a_T)$. Let S_T be the set of all permutations on T elements. Output the state*

$$\frac{1}{\sqrt{T!}} \sum_{\sigma \in S_T} |a_{\sigma(1)}, \cdots, a_{\sigma(T)}\rangle \ .$$

Proof. We ignore normalizations throughout the proof. Let $A_d^T \subseteq A^T$ the subset of A^T such that the T elements in the sequence $(x_1, \cdots, x_T) \in A^T$ are all distinct. We can consider T copies of the state $|\psi_{f,A}\rangle$:

$$(|\psi_{f,A}\rangle)^{\otimes T} := \left(\sum_{a \in A} \omega_{2^{5\lambda}}^{f(a)} \cdot |a\rangle\right)^{\otimes T} = \sum_{(a_1,\cdots,a_T) \in A^T} \omega_{2^{5\lambda}}^{\sum_{i\in[T]} f(a_i)} \cdot |a_1, \cdots, a_T\rangle \ .$$

Let Π_d be the $5\lambda{\cdot}T$-qubit projection that checks whether a classical $5\lambda{\cdot}T$-bit string is in A_d^T, and one can verify that (1) the probability for this projection to succeed for $(|\psi_{f,A}\rangle)^{\otimes T}$ is $\frac{T^2}{|A|} = \frac{T^2}{2^\lambda}$ and that (2) after a successful projection the state is,

$$|\psi_{f,A_d^T}\rangle := \sum_{(a_1,\cdots,a_T) \in A_d^T} \omega_{2^{5\lambda}}^{\sum_{i\in[T]} f(a_i)} \cdot |a_1, \cdots, a_T\rangle \ ,$$

which means that for every f, A, the trace distance between the two states is bounded by $\frac{T^2}{2^\lambda}$, which in turn implies that for random f, the distributions for T copies of $|\psi_{f,A}\rangle$ or one copy of the state $|\psi_{f,A_d^T}\rangle$ have trace distance $\leq \frac{T^2}{2^\lambda}$.

For a given subset $A \subseteq \{0,1\}^{5\lambda}$, $|A| = 2^\lambda$, consider the mixed state that corresponds to the distribution of $|\psi_{f,A_d^T}\rangle$ over a random f:

$$\mathbb{E}_f \left[|\psi_{f,A_d^T}\rangle\langle\psi_{f,A_d^T}| \right]$$

$$
\begin{aligned}
&:= \mathbb{E}_f \Bigg[\Bigg(\sum_{(a_1,\cdots,a_T)\in A_d^T} \omega_{2^{5\lambda}}^{\sum_{i\in[T]} f(a_i)} \cdot |a_1,\cdots,a_T\rangle \Bigg) \\
&\qquad \cdot \Bigg(\sum_{(b_1,\cdots,b_T)\in A_d^T} \omega_{2^{5\lambda}}^{\sum_{i\in[T]} -f(b_i)} \cdot \langle b_1,\cdots,b_T| \Bigg) \Bigg] \\
&= \mathbb{E}_f \Bigg[\sum_{\substack{(a_1,\cdots,a_T),\\(b_1,\cdots,b_T)\in A_d^T}} \omega_{2^{5\lambda}}^{\sum_{i\in[T]}(f(a_i)-f(b_i))} \cdot |a_1,\cdots,a_T\rangle\langle b_1,\cdots,b_T| \Bigg] \\
&= \sum_{\substack{(a_1,\cdots,a_T),\\(b_1,\cdots,b_T)\in A_d^T}} |a_1,\cdots,a_T\rangle\langle b_1,\cdots,b_T| \cdot \mathbb{E}_f \left[\omega_{2^{5\lambda}}^{\sum_{i\in[T]}(f(a_i)-f(b_i))} \right] \ .
\end{aligned}
$$

A useful property of the above expression is that for $\mathbf{a} := (a_1, \cdots, a_T)$, $\mathbf{b} := (b_1, \cdots, b_T)$, $\mathbf{a}, \mathbf{b} \in A_d^T$, if $\mathbf{a}$ and $\mathbf{b}$ are permutations of each other (that is, as T-element strings, with elements over the set $\{0,1\}^{5\lambda}$), then the expectation $\mathbb{E}_f \left[\omega_{2^{5\lambda}}^{\sum_{i\in[T]}(f(a_i)-f(b_i))} \right]$ is one, because $\sum_{i\in[T]} (f(a_i) - f(b_i)) = 0$.

Also, note the following: Because f is a random function from $\{0,1\}^{5\lambda}$ to $\{0,1\}^{5\lambda}$, then for every element $a \in \{0,1\}^{5\lambda}$

$$
\mathbb{E}_f \left[\omega_{2^{5\lambda}}^{f(a)} \right] = 0 \ ,
$$

which follows from the standard fact: $\forall N \in \mathbb{N} : \sum_{i\in[N]} \omega_N^i = 0$. Now, as an implication of the above, because $\mathbf{a}$ and $\mathbf{b}$ are strings in A_d^T rather than A^T (i.e. in each of them, their T elements are pairwise distinct), whenever the strings are not permutations of each other, it means there is a free element (say, in the set $\{a_1, \cdots, a_T\}$) that is not in the other set $\{b_1, \cdots, b_T\}$, which makes the expectation $\mathbb{E}_f \left[\omega_{2^{5\lambda}}^{\sum_{i\in[T]}(f(a_i)-f(b_i))} \right]$ zero.

Since only $\mathbf{a} \in A_d^T$ and its permutations stay in the sum, it follows that for every $A \subseteq \{0,1\}^{5n}$,

$$
\mathbb{E}_f \left[|\psi_{f,A_d^T}\rangle\langle\psi_{f,A_d^T}| \right] = \sum_{(a_1,\cdots,a_T)\in A_d^T, \sigma\in S_T} |a_{\sigma(1)}, \cdots, a_{\sigma(T)}\rangle\langle a_1, \cdots, a_T| \ ,
$$

which is in turn equal to,

$$
\rho := \sum_{(a_1,\cdots,a_T)\in A_d^T} \left(\sum_{\sigma\in S_T} |a_{\sigma(1)}, \cdots, a_{\sigma(T)}\rangle \right) \cdot \left(\sum_{\sigma\in S_T} \langle a_{\sigma(1)}, \cdots, a_{\sigma(T)}| \right) \ .
$$

The last mixed state ρ corresponds to the distribution of sampling a T-sized set $\{a_1, \cdots, a_T\}$ at random from A, and outputting a superposition of all of its permutations. One can think of a tweak of this distribution: instead of sampling a T-sized set, sample a T-sized multi-set of A (i.e. with repetitions) and then output the uniform superposition over all of its permutations - note that this is exactly the distribution Q_1. Finally, it is a known fact that the number of such subsets of A is $\binom{|A|}{T}$ and the number of such multi-sets is $\binom{|A|+T-1}{T}$. The probability to sample a set out of all of the multi-sets is overwhelming:

$$\frac{\binom{|A|}{T}}{\binom{|A|+T-1}{T}} = \frac{\frac{2^\lambda!}{T!\cdot(2^\lambda-T)!}}{\frac{(2^\lambda+T-1)!}{T!\cdot(2^\lambda-1)!}} = \frac{2^\lambda!\cdot T!\cdot(2^\lambda-1)!}{T!\cdot(2^\lambda-T)!\cdot(2^\lambda+T-1)!}$$

$$= \frac{2^\lambda!\cdot(2^\lambda-1)!}{(2^\lambda-T)!\cdot(2^\lambda+T-1)!} = \frac{\Big(2^\lambda-(T-1)\Big)\cdots\Big(2^\lambda-(1)\Big)}{(2^\lambda+(T-1))\cdots(2^\lambda+1)}$$

$$\prod_{i\in[T]}\left(1-\frac{2\cdot i}{2^\lambda}\right) \geq \left(1-\frac{2\cdot T}{2^\lambda}\right)^T \quad ,$$

which is $1-\mathsf{negl}(\lambda)$ whenever T is any polynomial in λ. It follows that the state ρ has statistical distance $\leq \mathsf{negl}(\lambda)$ from Q_1, and also has statistical distance bounded by $\frac{T^2}{2^\lambda} = \mathsf{negl}(\lambda)'$ from Q_0, which impies that the statistical distance between Q_0 and Q_1 is negligible. □

D Point and Permute Garbled Circuits [BMR90, AGQY22] for P/Poly from Pseudo One-time Pad

In this section, we will construct garbling schemes for $\mathsf{P/poly}$ based on PRFSPD. First, we recall the definition of a garbling scheme.

Definition 14 (Garbled circuits with classical encodings (adapted from [AGQY22]))**.** *A Garbling Scheme for a class of circuits $\mathcal{C}$ with classical encodings is a triplet of QPTalgorithms (Garble, $\mathit{InputEncode}$, Decode) with the following syntax:*

- $(\mathsf{GC},\mathsf{sk}) \leftarrow \mathit{Garble}(1^\lambda, C)$*: takes as input a security parameter λ, the classical description of a circuit $C\in\mathcal{C}$, and outputs a classical state* GC *called the garbled circuit, and a classical secret key* sk*.*
- $\sigma_x \leftarrow \mathit{InputEncode}(\mathsf{sk}, x)$*: takes as input a classical secret key* sk*, and a classical input x, and outputs a classical encoding σ_x.*
- $\chi \leftarrow \mathit{Decode}(\mathsf{GC}, \sigma_x)$*: takes as input a garbled circuit* GC*, an input encoding σ_x, and a classical output χ.*

Correctness *: For every circuit $C \in \mathcal{C}$, and input x, $C(x) = \mathcal{Decode}(\mathsf{GC}, \sigma_x)$, where $(\mathsf{GC}, \mathsf{sk}) \leftarrow \mathcal{Garble}(1^\lambda, C)$ and $\sigma_x \leftarrow \mathcal{InputEncode}(\mathsf{sk}, x)$.*

Security *: There exists a QPT simulator $\mathcal{Sim}$ such that for every $C \in \mathcal{C}$, input x, and QPT distinguisher D, there exists a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\Big|\Pr[1 \leftarrow D(\mathsf{GC}, \sigma_x) : \sigma_x \leftarrow \mathcal{InputEncode}(\mathsf{sk}, x), (\mathsf{GC}, \mathsf{sk}) \leftarrow \mathcal{Garble}(1^\lambda, C)] \tag{85}$$

$$- \Pr[1 \leftarrow D(\mathsf{GC}, \sigma_x) : (\mathsf{GC}, \sigma_x) \leftarrow \mathcal{Sim}(1^\lambda, C, C(x))]\Big| \leq \mathsf{negl}(\lambda). \tag{86}$$

Construction We construct a garbling scheme for $\mathsf{P/poly}$. We assume that each gate in the circuit has a fan-in of 2 and a fan-out of 1. Let $\mathcal{E} = (\mathcal{Enc}, \mathcal{Dec})$ be a CPA-secure encryption scheme with message length strictly greater than the key length[15]. By Remark 23, we know that such encryption schemes can be constructed from PRFSPD. It is known that garbled circuits can be constructed from any CPA-secure encryption scheme with the appropriate parameters mentioned above (and actually, a pseudo-encryption scheme [AGQY22] suffices) [BMR90]. We repeat the construction for completeness, without providing the completeness and security analysis. This observation was already made in the case of quantum garbled circuits by [AQY21].

The correctness and security guarantees follow from arguments identical to the original work of [BMR90]. Hence we omit the proofs for brevity.

E Proofs of the results in Section 4

E.1 Proofs of the theorems in Section 4.1

Proof of Theorem 8. Fix a message $m \in \{0,1\}$ arbitrarily. Let $\mathsf{sig} \leftarrow \mathcal{Sign}(sk, m)$ where $(sk, pk) \leftarrow \mathcal{Keygen}(1^\lambda)$.

By the $\mathsf{Unforgeability\text{-}of\text{-}proofs}$ of $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$, there exists a negligible function $\mathsf{negl}(\lambda)$, such that

$$\Pr_{\substack{k,\tilde{k} \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)};\\ r \leftarrow \mathcal{Destruct}(\mathcal{Gen}(\tilde{k}))}} [\mathcal{Ver}(k, r) = 1] = \mathsf{negl}(\lambda). \tag{87}$$

Hence, the probability that the first step Item 1 of $\mathcal{Verify}(\mathsf{pk}, m, \mathsf{sig})$ (see

[15] This is sufficient because we assume the fan-out of each gate to be 1. For a circuit with fan-out r, we would need $n(\lambda) = r \cdot (w(\lambda) + 1)$

Assumes: CPA encryption scheme, $(\mathcal{Enc}, \mathcal{Dec})$ with key length $k(\lambda)$, and message length $k(\lambda)+1$.

$\mathcal{Garble}(1^\lambda, C)$

1. For every wire $w \in C$, sample two strings $k_w^0, k_w^1 \leftarrow \{0,1\}^{k(\lambda)}$, and $r_w \leftarrow \{0,1\}$.

2. For every gate $G \in C$, compute a garbled gate consisting of four entries, indexed by elements $\{0,1\}^2$. Let w_1, w_2 be the input wires of G and w_3 be the output wire. For every $(b_1,b_2) \in \{0,1\}^2$, the $(b_1,b_2)^{th}$ entry is $\rho_G^{b_1,b_2} = \mathcal{Enc}(k_{w_1}^{b_1} \oplus k_{w_2}^{b_2}, \theta_{G,b_1,b_2})$, where
$$\theta_{G,b_1,b_2} = \left(k_{w_3}^{G(b_1 \oplus r_{w_1}, b_2 \oplus r_{w_2}) \oplus r_{w_3}} \| G(b_1 \oplus r_{w_1}, b_2 \oplus r_{w_2}) \oplus r_{w_3}\right).$$
Let the concatenation of all the ciphers in the corresponding order be $\mathcal{T}_G$.

3. Let $\mathcal{W}_{out}$ be the set of all output wires of the circuit C. Compute the translation table $\{\mathcal{O}_w\}_{w \in \mathcal{W}_{out}}$ where for each $w \in \mathcal{W}_{out}$, $\mathcal{O}_w(k_w^{b+r_w}) = b$.
4. Output $(\{\mathcal{T}_G\}_{G \in C}, \{\mathcal{O}_w\}_{w \in \mathcal{W}_{out}})$ as GC, the garbled table and $\{k_w^b, r_w\}_{b \in \{0,1\}^w, w \in \mathcal{W}_{in}}$, where $\mathcal{W}_{in}$ is the set of input wires.

$\mathcal{InputEncode}(sk, x)$

1. Let $d(\lambda)$ be the input length of C, and $\pi : \mathcal{W}_{in} \to [d]$, be the function assigning the input wire to the bits of the input.
2. Output $\sigma_x = \left\{\left(k^{r_w \oplus x_{\pi(w)}}, r_w \oplus x_{\pi(w)}\right)\right\}_w$.

$\mathcal{Decode}(\mathsf{GC}, \sigma_x)$

1. For every $G \in C$, with input wire w_1, w_2 and output wire w_3,
   (a) let k'_{w_1}, k'_{w_2} be the keys recovered for the gate G.
   (b) Compute $k'_{w_3} \| r_{w_3} \leftarrow \mathcal{Dec}(k'_{w_1} \oplus k'_{w_2}, \rho_G^{r_{w_1}, r_{w_2}})$.
2. Continue layer-by-layer till the output wires are reached. Finally output $\{\mathcal{O}_w(k'_w)\}_{w \in \mathcal{W}_{out}}$.

Figure 9: Garbled scheme $\mathcal{G}$.

Fig. 4) would reject,

$$\Pr_{\substack{k \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}; \\ \forall j \in [w^2], k_j \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}, \\ r_j \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k_j))}} [\exists j \in [w^2]\ \mathcal{Ver}(k, r_j) = 1] \tag{88}$$

$$\leq \sum_{j \in [w^2]} \Pr_{\substack{k, k_j \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}; \\ r_j \leftarrow \mathcal{Destruct}(\mathcal{Gen}(k_j))}} [\mathcal{Ver}(k, r_j) = 1] \tag{89}$$

$$= \sum_{j \in [w^2]} \Pr_{\substack{k, \tilde{k} \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}; \\ r \leftarrow \mathcal{Destruct}(\mathcal{Gen}(\tilde{k}))}} [\mathcal{Ver}(k, r) = 1] \tag{90}$$

$$= \sum_{j \in [w^2]} \mathsf{negl}(\lambda) \qquad \text{By Eq. (87)} \tag{91}$$

$$= w^2 \cdot \mathsf{negl}(\lambda), \tag{92}$$

which is negligible, since $w(\lambda) \in \mathsf{poly}(\lambda)$.

Next, by the correctness of $(\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$ (see Definition 5), the second step of $\mathcal{Verify}(\mathsf{pk}, m, \mathsf{sig})$ (see Item 2) with probability 1, i.e., it would fail with probability 0.

Therefore, the probability that $\mathcal{Verify}(\mathsf{pk}, m, \mathsf{sig})$ (see Fig. 4) would reject, i.e. either one of the two steps (see Items 1 and 2) will fail, is negligible. □

Proof of Lemma 7. The main ingredient in the proof is to use Lemma 2 to bound the probabilities of events concerning proofs sampled from $\mathsf{Correlated\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}$ using the probability for the analogous events with respect to $\mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}$. In particular, for every $s \in \{0,1\}^w$,

$$\Pr[\wedge_{j\in[w]}\mathsf{Success}(s, q_j)] \tag{93}$$

$$= \Pr_{(q_1,\ldots,q_w)\sim\mathsf{Correlated\text{-}Destruction}_w^{\mathcal{Haar},\mathcal{Destruct}}}[\wedge_{j\in[w]}\mathsf{Success}(s, q_j)] \tag{94}$$

$$\leq \frac{N^w}{\binom{N+w-1}{w}} \Pr_{(q_1,\ldots,q_w)\sim\mathsf{Product\text{-}Destruction}_w^{\mathcal{Haar},\mathcal{Destruct}}}[\wedge_{j\in[w]}\mathsf{Success}(s, q_j)] \quad \text{By Lemma 2.} \tag{95}$$

Since, $\mathsf{Product\text{-}Destruction}_w^{\mathcal{Haar},\mathcal{Destruct}}$ is the w-fold product of $\mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}$, we conclude from Eq. (95) that for every $s \in \{0,1\}^w$,

$$\Pr[\wedge_{j\in[w]}\mathsf{Success}(s, q_j)] \tag{96}$$

$$\leq \frac{N^w}{\binom{N+w-1}{w}} \Pr_{(q_1,\ldots,q_w)\sim\mathsf{Product\text{-}Destruction}_w^{\mathcal{Haar},\mathcal{Destruct}}}[\wedge_{j\in[w]}\mathsf{Success}(s, q_j)] \tag{97}$$

$$= \frac{N^w}{\binom{N+w-1}{w}} \sum_{s\in\mathsf{Success}_{<\frac{1}{w}}} \left(\Pr_{f\sim\mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}}[\mathsf{Success}(s, f)] \right)^w . \tag{98}$$

Hence, the first term in Lemma 7,

$$
\begin{aligned}
&\Pr\left[\left(\wedge_{j\in[w]}\mathsf{Success}(S,q_j)\right) \wedge S \in \mathsf{Success}_{<\frac{1}{w}}\right] && (99)\\
&= \sum_{s\in\mathsf{Success}_{<\frac{1}{w}}} \Pr[\wedge_{j\in[w]}\mathsf{Success}(s,q_j)]\Pr[S=s] && (100)\\
&\leq \sum_{s\in\mathsf{Success}_{<\frac{1}{w}}} \Pr[\wedge_{j\in[w]}\mathsf{Success}(s,q_j)] && (101)\\
&\leq \frac{N^w}{\binom{N+w-1}{w}} \sum_{s\in\mathsf{Success}_{<\frac{1}{w}}} \left(\Pr_{f\sim\mathsf{Product\text{-}Destruction}^{\mathcal{Haar},\mathcal{Destruct}}}[\mathsf{Success}(s,f)]\right)^w && \text{By Eq. (98)} \quad (102)\\
&\leq \frac{N^w}{\binom{N+w-1}{w}} \sum_{s\in\mathsf{Success}_{<\frac{1}{w}}} \left(\frac{1}{w}\right)^w && \text{By definition of } \mathsf{Success}_{<\frac{1}{w}}. \quad (103)\\
&= \frac{N^w}{\binom{N+w-1}{w}} \frac{|\mathsf{Success}_{<\frac{1}{w}}|}{w^w} && (104)\\
&\leq \frac{N^w}{\binom{N+w-1}{w}} \frac{2^w}{w^w} && (105)\\
&= \frac{N^w N! w!}{(N+w-1)!} \frac{2^w}{w^w} && (106)\\
&= w! \frac{N^w}{\prod_{r\in[w](N+r-1)}} \frac{2^w}{w^w} = \frac{N^w}{\prod_{r\in[w](N-1+r)}} \frac{(w!)2^w}{w^w}, && (107)
\end{aligned}
$$

which is negligible considering $w \in \mathsf{poly}(\lambda)$ and $N \in \exp(\lambda)$, because $\frac{N^w}{\prod_{r\in[w](N-1+r)}} \to 1$[16] as $\lambda \to \infty$, and by Stirling's upper bound,

$$
\frac{(w!)2^w}{w^w} \leq \frac{\sqrt{2\pi w}\left(\frac{w}{e}\right)^w e^{\frac{1}{12w}} 2^w}{w^w} = \frac{(\sqrt{2\pi w})e^{\frac{1}{12w}}}{\left(\frac{e}{2}\right)^w},
$$

which is negligible.

Next, for the second term in Lemma 7, we use the fact that $(r_1, \ldots, r_{w^2})$ are chosen independently of the key S submitted by the adversary. Hence, for every $s \in \mathsf{Success}_{\geq\frac{1}{w}}$,

$$
\begin{aligned}
&= \Pr[\wedge_{j\in[w^2]}\mathsf{Fail}(s,r_j)] && (108)\\
&= \prod_{j\in[w^2]} \Pr[\mathsf{Fail}(s,r_j)] = \prod_{j\in[w^2]} (1 - \Pr[\mathsf{Success}(s,r_j)]) \leq \left(1-\frac{1}{w}\right)^{w^2} \leq (e^{-1})^w = e^{-w}. && (109)
\end{aligned}
$$

[16] One way to see it is: $1 \leq \frac{N^w}{\prod_{r\in[w](N-1+r)}} \leq \left(\frac{1}{\left(1-\frac{1}{N}\right)}\right)^w = \left(1+\frac{1}{N-1}\right)^w$, which converges to 1 since $w \in o(N-1)$.

Hence,

$$
\begin{aligned}
&\Pr\left[\left(\wedge_{j\in[w]}\mathsf{Fail}(S,r_j)\right)=1\wedge S\in\mathsf{Success}_{\geq\frac{1}{w}}\right] && (110)\\
&=\sum_{s\in\mathsf{Success}_{\geq\frac{1}{w}}}\Pr[\wedge_{j\in[w]}\mathsf{Fail}(s,r_j)]\Pr[S=s] && (111)\\
&\leq\sum_{s\in\mathsf{Success}_{\geq\frac{1}{w}}}e^{-w}\Pr[S=s] \qquad\qquad \text{By Eq. (109)} && (112)\\
&\leq e^{-w}, && (113)
\end{aligned}
$$

which is negligible.

□

E.2 Proofs of the theorems in Section 4.3

Proof of Theorem 15. Let $\mathcal{R}^*_\lambda$ be the malicious receiver, and $\sigma^b_{\mathcal{R}^*}$ be the state held by $\mathcal{R}^*$ at the end of the commit phase. We will show that $\sigma^0_{\mathcal{R}^*}\approx_c\sigma^1_{\mathcal{R}^*}$. Note that,

$$\sigma^0_{\mathcal{R}^*}=(p_1,\ldots,p_{\tilde{r}},c_1,\ldots,c_{\tilde{r}})\quad\sigma^1_{\mathcal{R}^*}=(c_1\oplus p_1,\ldots,c_{\tilde{r}}\oplus p_{\tilde{r}},c_1,\ldots,c_{\tilde{r}})$$

where $(p_1,\ldots,p_{\tilde{r}})\leftarrow\mathcal{Destruct}^{\otimes\tilde{r}}\left((\mathcal{Gen}(k))^{\otimes}\tilde{r}\right)$.

Since the key k is not revealed to the $\mathcal{R}^*$ during the *Commit* phase, by the $\mathsf{Pseudorandom\text{-}proofs}$ property of $(\mathcal{Gen},\mathcal{Destruct},\mathcal{Verify})$,

$$\sigma^0_{\mathcal{R}^*}\approx_c u^0_{\mathcal{R}^*},\text{ and }\sigma^1_{\mathcal{R}^*}\approx_c u^1_{\mathcal{R}^*},$$

where $u^0_{\mathcal{R}^*}=\{(u_1,\ldots,u_{\tilde{r}},c_1,\ldots,c_{\tilde{r}})\}_{u_i\xleftarrow{u}\{0,1\}^c}$ and $u^1_{\mathcal{R}^*}=\{(c_1\oplus u_1,\ldots,c_{\tilde{r}}\oplus u_{\tilde{r}},c_1,\ldots,c_{\tilde{r}})\}_{u_i\xleftarrow{u}\{0,1\}^c}$.

Finally, note that for any choice of $(c_1,\ldots,c_{\tilde{r}})$, $u^0_{\mathcal{R}^*}$ and $u^1_{\mathcal{R}^*}$ are the same distribution. Hence, putting it all together,

$$\sigma^0_{\mathcal{R}^*}\approx_c u^0_{\mathcal{R}^*}=u^1_{\mathcal{R}^*}\approx_c\sigma^1_{\mathcal{R}^*}.$$

□

Proof of Theorem 16. Let $\mathcal{C}^*_\lambda$ be the malicious committer and $\mathcal{R}_\lambda$ be the honest receiver. Let $\mathsf{Bad\text{-}String}\subset\{0,1\}^c$ be the set of all strings e such that there exists $k_1,k_2\in\{0,1\}^w$, and f_1,f_2 such that $f_2=e\oplus f_1$, i.e., $f_1\oplus f_2=c$ and

$$\Pr[\mathcal{Ver}(k_1,f_1)=1]\geq\left(1-\frac{1}{r(\lambda)}\right),\quad\Pr[\mathcal{Ver}(k_2,f_2)=1]\geq\left(1-\frac{1}{r(\lambda)}\right).$$

Let $\mathsf{Good\text{-}String}=\overline{\mathsf{Bad\text{-}String}}$.

We start with bounding the size of $\mathsf{Bad\text{-}String}$. For every $k \in \{0,1\}^w$, let $\mathsf{Proof\text{-}Challenge}_k \subset \{0,1\}^w$ be the set of all strings f such that $\Pr[\mathcal{Ver}(k_1, d) = 1] \geq \left(1 - \frac{1}{r(\lambda)}\right)$. For any (possibly non-distinct) pair of keys k_1, k_2 let $\mathsf{Proof\text{-}Challenge}_{k_1,k_2}$ be the set

$\mathsf{Proof\text{-}Challenge}_{k_1,k_2} \equiv \{f_1 \oplus f_2 \mid f_1 \in \mathsf{Proof\text{-}Challenge}_{k_1} \wedge f_2 \in \mathsf{Proof\text{-}Challenge}_{k_2}\}$.

By the $\left(1 - \frac{1}{r(\lambda)}, 2^{m(\lambda)}\right)$-bounded proofs property of $\mathcal{Gen}, \mathcal{Destruct}, \mathcal{Ver})$, for every $k \in \{0,1\}^w$, $|\mathsf{Proof\text{-}Challenge}_k| \leq 2^m$. Hence for every $k_1, k_2 \in \{0,1\}^w$, $|\mathsf{Proof\text{-}Challenge}_{k_1,k_2}| \leq 2^{2m}$. Note that, $\mathsf{Bad\text{-}String} = \bigcup_{k_1,k_2 \in \{0,1\}^w} \mathsf{Proof\text{-}Challenge}_{k_1,k_2}$. Hence

$$|\mathsf{Bad\text{-}String}| \leq \sum_{k_1,k_2 \in \{0,1\}^w} \mathsf{Proof\text{-}Challenge}_{k_1,k_2} \tag{114}$$

$$\leq \sum_{k_1,k_2 \in \{0,1\}^w} 2^{2m} = 2^{2w+2m}. \tag{115}$$

Let $C_1, \ldots, C_{\tilde{r}}$ be the random messages sent by $\mathcal{R}_\lambda$, $P_1, \ldots, P_{\tilde{r}}$ be the response of $\mathcal{C}^*_\lambda$ in the *Commit* phase, and Good be the event that $C_1, \ldots, C_{\tilde{r}} \in \mathsf{Good\text{-}String}$. Let $\mathsf{bad} = \mathsf{Good}^c$. Note that,

$$\Pr[Bad] \leq \sum_{i \in [\tilde{r}]} \Pr[C_i \in \mathsf{Bad\text{-}String}] = \sum_{i \in [\tilde{r}]} \frac{\mathsf{Bad\text{-}String}}{2^c} \leq \sum_{i \in [\tilde{r}]} \frac{2^{2w+2m}}{2^c} = \frac{\tilde{r}}{2^{c-2w-2m}}, \tag{116}$$

which is negligible since $c - 2w - 2m \in \omega(\log(\lambda))$.

We define the extractor $\mathcal{E}$ as follows. If $C_1, \ldots, C_{\tilde{r}} \in \mathsf{Bad\text{-}String}$, $\mathcal{E}$ outputs $\perp$. Else, $\mathcal{E}$ checks if $P_1, \ldots, P_{\tilde{r}} \in \bigcup_k \mathsf{Proof\text{-}Challenge}_k$ and outputs 0 if that's the case, else checks if $C_1 \oplus P_1, \ldots, C_{\tilde{r}} \oplus P_{\tilde{r}} \in \bigcup_k \mathsf{Proof\text{-}Challenge}_k$, and outputs 1 if that is the case, else outputs $\perp$. . Note that, since $C \in \mathsf{Good\text{-}String}$, D and $D \oplus C$ cannot both be in $\bigcup_k \mathsf{Proof\text{-}Challenge}_k$. Hence, conditioned on the event Good and that $\mathcal{E}(C_1, \ldots, C_{\tilde{r}}, P_1, \ldots, P_{\tilde{r}}) = 0$, then there cannot exist $k \in \{0,1\}^w$ such that $\Pr[\mathcal{Ver}(k, C_i \oplus P_i) = 1] \geq 1 - \frac{1}{r}$, for some $i \in [\tilde{r}]$. Hence for every $k \in \{0,1\}^w$,

$$\Pr[\mathcal{Ver}(k, C_i \oplus P_i) = 1 \quad \forall i \in [\tilde{r}]] \tag{117}$$

$$= \prod_{i=1}^{\tilde{r}} \Pr[\mathcal{Ver}(k, C_i \oplus P_i)] \qquad \text{For any fixed key, verification is independent.} \tag{118}$$

$$\leq \left(1 - \frac{1}{r}\right)^{\tilde{r}} \tag{119}$$

$$\leq e^{-\frac{\tilde{r}}{r}} = e^{-\lambda}. \tag{120}$$

Let S be the random variable representing the key that $\mathcal{C}^*_\lambda$ sends in the *Reveal* phase. Hence for every $k \in \{0,1\}^w$

$$\Pr[Reveal(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 1 \mid \mathcal{E}(\tau) = 0 \wedge \mathsf{Good} \wedge S = k] \leq e^{-\lambda}.$$

Therefore,

$$\begin{aligned}
&\Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 1 \mid \mathcal{E}(\tau) = 0 \wedge \mathsf{Good}] && (121)\\
&= \sum_{k \in \{0,1\}^w} \Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 1 \mid \mathcal{E}(\tau) = 0 \wedge \mathsf{Good} \wedge S = k] \Pr[S = k \mid \mathcal{E}(\tau) = 0 \wedge \mathsf{Good}] && (122)\\
&\le e^{-\lambda} \sum_{k \in \{0,1\}^w} \Pr[S = k \mid \mathcal{E}(\tau) = 0 \wedge \mathsf{Good}] && (123)\\
&\le e^{-\lambda}. && (124)
\end{aligned}$$

Hence,

$$\begin{aligned}
&\Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 1 \mid \mathcal{E}(\tau) = 0] && (125)\\
&\le \Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 1 \wedge \mathsf{Good} \mid \mathcal{E}(\tau) = 0] + \Pr[\mathsf{bad}] && (126)\\
&\le \Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 1 \wedge \mathsf{Good} \mid \mathcal{E}(\tau) = 0 \wedge \mathsf{Good}] + \Pr[\mathsf{bad}] && (127)\\
&\le e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}} && \text{By Eqs. (116) and (124)} \quad (128)
\end{aligned}$$

Similarly, we can bound

$$\Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 0 \mid \mathcal{E}(\tau) = 1] \le e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}}. \tag{129}$$

For the case when $\mathcal{E}(\tau)$ outputs $\perp$, we only need to consider conditioned on the event Good as demonstrated in the previous cases. Note that conditioned on Good and $\mathcal{E}(\tau) = \perp$, it must hold that neither $P_1, \ldots, P_{\tilde{r}} \in \bigcup_k \mathsf{Proof\text{-}Challenge}_k$ nor $C_1 \oplus P_1, \ldots, C_{\tilde{r}} \oplus P_{\tilde{r}} \in \bigcup_k \mathsf{Proof\text{-}Challenge}_k$. Hence following the same analysis as above, we can bound

$$\Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 0 \mid \mathcal{E}(\tau) = \perp] \le e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}},$$

and

$$\Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) = 1 \mid \mathcal{E}(\tau) = \perp] \le e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}}.$$

Combining the last two equations, we get

$$\Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) \neq \perp \mid \mathcal{E}(\tau) = \perp] \le 2\left[e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}}\right]. \tag{130}$$

Combining all the last-four equations, we get

$$\begin{aligned}
&\Pr[\mu \neq \{b^*\} \cup \{\perp\} \mid b^* \leftarrow \mathcal{E}(\tau), \mu \leftarrow \mathit{Reveal}(\mathcal{C}^*, \mathcal{R}, \sigma_{C^*,\mathcal{R}})]\\
&= \sum_{b^* \in \{0,1,\perp\}} \Pr[\mathit{Reveal}(\mathcal{C}^*_\lambda, \mathcal{R}_\lambda, \sigma_{C^*,\mathcal{R}}) \notin \{b^*\} \cup \{\perp\} \mid \mathcal{E}(\tau) = b^*]\\
&\le \left(e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}}\right) + \left(e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}}\right) + \left(2\left[e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}}\right]\right) && \text{By Eqs. (128) to (130)}\\
&= 4\left(e^{-\lambda} + \frac{\tilde{r}}{2^{c-2w-2m}}\right),
\end{aligned}$$

which is negligible. □

F Lower bounds for the applications

In this section, we discuss why computational assumptions are necessary for the cryptographic primitives we discussed in the applications. Since most of the primitives involve classical communication, the primary technique that we use for most of these impossibility results is a classical analogue of shadow tomography [Aar18, Proposition 17]. We note that similar results for the analogous primitive with quantum communication, via shadow tomography [Aar18, Theorem 2].

Proposition 5 (Impossiblity result for unconditionally secure CMA MAC)**.** *There cannot exist an* CMA *secure* MAC *scheme (see Definition 11) that is secure against unbounded adversaries.*

Proof sketch. Let $\mathcal{M}$ be an arbitrary MAC scheme with key length $w(\lambda)$. We will construct an algorithm $\mathcal{A}$ that makes only polynomially many queries to the $\mathcal{Sign}$ oracle in the forging game $\mathsf{Strong\text{-}CMA\text{-}Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathcal{M}}$. $\mathcal{A}$ fixes $w(\lambda)+1$ distinct strings $x_1, x_2, \dots, x_{w+1}$. Let $\epsilon = \frac{1}{\mathsf{poly}(\lambda)}$ be a small enough error precision. Let k denote the key sampled by the challenger, and for each $i \in [w+1]$, let T_i denote the distribution of the tags $\mathcal{Sign}(k, x_i)$. $\mathcal{A}$ initializes a set S_0 to the full keyspace $\{0,1\}^w$, and samples an index $J \overset{u}{\leftarrow} [w+1]$.

Next for $i = 1$ to J, she does the following. She makes q_ϵ queries to the signing oracle with message x_i, i.e., to the distribution T_i and computes $\mathsf{prob}_{\tilde{k},i} \equiv \Pr_{\mathsf{sig}\sim T_i}[\mathcal{Verify}(\tilde{k}, \mathsf{sig}) = 1]$ for every $\tilde{k} \in S_{i-1}$, upto error ϵ with probability $1 - 2^{-\lambda}$. By [Aar18, Proposition 17], this can be done with $q_\epsilon \in \mathsf{poly}(\lambda, \epsilon)$. After this, $\mathcal{A}$ computes the set $S_i \in S_{i-1}$ the set of all $\tilde{k} \in S_i$, such that $\mathsf{prob}_{\tilde{k},i} \geq 1-\epsilon$. If the estimation is correct (which happens with probability at least $1 - \frac{1}{2^\lambda}$), S_i will be non-empty since $k \in S_i$. $\mathcal{A}$ samples a key $k_i \overset{u}{\leftarrow} S_i$, and computes $\mathsf{sig}_i = \mathcal{Sign}(k_i, x_{i+1})$. If $i = J$, she ends the game and submits sig_i to the challenger as the alleged new tagged message, other moves to $i+1$.

Clearly, if $\mathcal{A}$ terminates at any stage $i \in [w+1]$, then she wins the game if $J = i$, and $\mathcal{Verify}(k, \mathsf{sig}_i)$, passes. Since $w \in \mathsf{poly}(\lambda)$, $\Pr[J = i] = \frac{1}{w+1}$ is non-negligible for every i. Hence, if $\Pr[\mathcal{Verify}(k, \mathsf{sig}_i) = 1]$ is non-negligible for any $i \in [w+1]$, then $\mathcal{A}$ would win the game with a non-negligible probability, too. Hence, it is enough to show that with overwhelming probability, $\exists i \in [w+1]$, such that $\Pr[\mathcal{Verify}(k, \mathsf{sig}_i) = 1]$ is non-negligible.

Note that if for any $i \in [w+1]$, $\Pr[\mathcal{Verify}(k, \mathsf{sig}_i) = 1]$ is negligible, then S_{i-1} must be a negligible fraction of S_i, and hence $\frac{|S_{i-1}|}{|S_i|}$. Therefore, if $\Pr[\mathcal{Verify}(k, \mathsf{sig}_i) = 1]$ is negligible for every i, and we assume that shadow tomography estimations were correct upto ϵ-precision (which happens with probability at least $1 - \frac{w+1}{2^\lambda}$), then

$$S_{w+1} \leq \frac{|S_0|}{2^{w+1}},$$

which is strictly less than 1. Hence, with probability $1 - \frac{w+1}{2^\lambda}$, there must exist $i \in [w+1]$, such that $\Pr[\mathcal{Verify}(k, \mathsf{sig}_i) = 1]$ is non-negligible. These arguments can be formalized by choosing $\epsilon \in \frac{1}{\mathsf{poly}(\lambda)}$ appropriately. □

Proposition 6 (Impossibility of classically-verifiable private quantum coins with unconditional security)**.** *There cannot exist a* classically-verifiable private quantum coins *scheme (see Definition 6) that is secure against unbounded adversaries.*

Proof sketch. Let NCV-Coin be an arbitrary classically-verifiable private quantum coins scheme. We will construct an algorithm $\mathcal{A}$ that makes only polynomially many queries to the $\mathcal{Mint}$ oracle in the forging game $\mathsf{Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathsf{NCV\text{-}Coin}}$. Let k denote the key sampled by the challenger, and $|\psi_k\rangle$ the corresponding coin state, and for each $i \in [w+1]$, let T_k denote the distribution of the certificate $\mathcal{Cert\text{-}Gen}(|\psi_k\rangle)$. Let $\epsilon \in \frac{1}{\mathsf{poly}(\lambda)}$.$\mathcal{A}$ computes $\mathsf{prob}_{\tilde{k}} \equiv \Pr_{\mathsf{cert} \sim T_k}[\mathcal{Cert\text{-}Verify}(\tilde{k}, \mathsf{cert}) = 1]$ for every $\tilde{k} \in \{0,1\}^w$, upto error ϵ with probability $1 - 2^{-\lambda}$, using q_ϵ samples from T_k, where $q_\epsilon \in \mathsf{poly}(\lambda, \epsilon)$, by [Aar18, Proposition 17]. She generates the q_ϵ samples from T_k, $\mathsf{cert}_1, \ldots, \mathsf{cert}_{q_\epsilon}$ by querying $\mathcal{Mint}$ oracle q_ϵ times to get the state $|\psi_k\rangle^{\otimes q_\epsilon}$ and then running $\mathcal{Cert\text{-}Gen}^{\otimes q_\epsilon}$ on them, to get $\mathsf{cert}_1, \ldots, \mathsf{cert}_{q_\epsilon}$. If $\mathsf{cert}_1, \ldots, \mathsf{cert}_{q_\epsilon}$ are not distinct, she aborts. By Item 2 of the correctness of NCV-Coin, she would abort at this step only with negligible probability. She then computes the set $S \subset \{0,1\}^w$, which is the set of all $\tilde{k}$ such that $\mathsf{prob}_{\tilde{k}} \geq 1 - \epsilon$. Note that the if the tomography was correct, then $k \in S$. For each $\tilde{k} \in S$, she runs $\mathcal{Cert\text{-}Gen}(\mathcal{Mint}(\tilde{k}))$ to get $\mathsf{cert}_{\tilde{k}}$. If $\mathsf{cert}_{\tilde{k}} \in \{\mathsf{cert}_1, \ldots, \mathsf{cert}_{q_\epsilon}\}$ for every $\tilde{k} \in S$, then she aborts. Otherwise, she selects a $\mathsf{cert}_{\tilde{k}} \notin \{\mathsf{cert}_1, \ldots, \mathsf{cert}_{q_\epsilon}\}$, and outputs it along with $\mathsf{cert}_1, \ldots, \mathsf{cert}_{q_\epsilon}$. Note that if the tomography estimation was correct, then $k \in S$. Independently, by correctness of NCV-Coin, Item 2), $\mathsf{cert}_k \notin \{\mathsf{cert}_1, \ldots, \mathsf{cert}_{q_\epsilon}\}$ with overwhelming probability. Hence if the tomography estimation was correct, she would not abort at the last step with overwhelming probability, i.e., probability $1 - \mathsf{negl}(\lambda)$ for some negligible probability. Note that if $\mathcal{A}$ did not abort, she would submit $q_\epsilon + 1$ many certificates, $\mathsf{cert}_{\tilde{k}}, \mathsf{cert}_1, \ldots, \mathsf{cert}_{q_\epsilon}$, out of which the last q_ϵ will pass verification with certainty due to Item 1 of the correctness of NCV-Coin, whereas $\mathsf{cert}_{\tilde{k}}$ will pass verification with probability at least $1-2\epsilon$, assuming the tomography estimation was correct. Hence, assuming the tomography estimation was correct which happens with probability $1 - 2^{-\lambda}$, $\mathcal{A}$ wins $\mathsf{Forging\text{-}Exp}_\lambda^{\mathcal{A},\mathsf{NCV\text{-}Coin}}$ with probability at least $(1-2\epsilon)(1 - \mathsf{negl}(\lambda))$, which is non-negligible since $\epsilon \in \frac{1}{\mathsf{poly}(\lambda)}$. □

Proposition 7 (Impossibility result for unconditionally secure One-Time-Signature)**.** *There cannot exist a* One-Time-Signature *scheme (see Definition 5) that is secure against unbounded adversaries.*

Proof sketch. The attack is similar to the one against Public-key quantum money. Let OTS be One-Time-Signature for 1-bit messages, with statistical correctness with precision δ (see Definition 5). Let (sk, pk) be the key pair that the challenger samples in the $\mathsf{Forging\text{-}Exp}_{\lambda}^{\mathcal{A},\mathsf{OTS}}$. Given the public key pk, $\mathcal{A}$ simply brute forces over all possible strings of the appropriate length, to obtain a string sig such that $\mathcal{Verify}(pk, 0, \mathsf{sig}) \geq 1 - 2\delta$. We know such a string exists since $\mathsf{sig}_0 \leftarrow \mathcal{Sign}(sk, 0)$ would pass verification with probability at least $1 - \delta$. □

Next for the encryption schemes, we will argue why Pseudo One-Time Pad or pseudo OTP schemes, which are a weaker primitive than CPA encryption, cannot exist unconditionally. In pseudo-one-time pads, the message length is strictly larger than the key length and the security guarantee is that for any two messages m_0, m_1, the cipher distributions $(C_0, C_1) = (\{\mathcal{Enc}(K, m_0)\}_{K \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}}, \{\mathcal{Enc}(K, m_1)\}_{K \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}})$ are indistinguishable, for any two messages m_0, m_1.

Proposition 8 (Impossibility result for pseudo OTP)**.** *There cannot exist a* pseudo OTP *scheme that is secure against unbounded adversaries. The impossibility result holds even with quantum ciphers.*

Proof sketch. We will sketch a proof for pseudo OTP schemes with classical ciphers. Let $m \in \{0,1\}^{d(\lambda)}$ be arbitrary, and let $C_m = \{\mathcal{Enc}(K, m)\}_{K \overset{u}{\leftarrow} \{0,1\}^{w(\lambda)}}$. By the correctness guarantee of the pseudo OTP scheme, there exists a key k such that $\Pr[\mathcal{Dec}(k, C_m) = m]$ is overwhelming.

Hence, there are at most a $2^{w(\lambda)}$ many messages $\tilde{m} \in \{0,1\}^{d(\lambda)}$, such that there exists $k \in \{0,1\}^{w(\lambda)}$ such that $\Pr[\mathcal{Dec}(k, C_m) = \tilde{m}]$ is an overwhelming function. Note that, the message length is strictly larger than the key length, i.e., $d(\lambda) \geq w(\lambda) + 1$. Hence with probability $1 - \frac{2^{w(\lambda)}}{2^{d(\lambda)}}$, which is at least $\frac{1}{2}$, for $M \overset{u}{\leftarrow} \{0,1\}^{d(\lambda)}$, it holds that

$$\Pr[\mathcal{Dec}(k, C_m) = M] \leq 1 - \frac{1}{p},$$

for every $k \in \{0,1\}$, for some $p \in \mathsf{poly}(\lambda)$. Since m was arbitrary the above also holds for a randomly chosen m.

This gives us a recipe for a distinguishing attack against the pseudo OTP scheme with respect to (m_0, m_1) where $m_0, m_1 \overset{u}{\leftarrow} \{0,1\}^{d(\lambda)}$. Given a challenge cipher distribution C_b, the all-powerful adversary would perform a classical version of shadow tomography on the cipher distribution C_b, with respect to the predicates $\{V_{k,b}\}_{k \in 0,1^{w(\lambda)}, b' \in \{0,1\}}$, where $V_{k,b'}$ outputs 1 if

$\mathcal{Dec}(k, C_b) = m_{b'}$. By [Aar18, Proposition 17], we only need polynomially many samples to estimate $\mathbb{E}_{c \sim C_b} \Pr[V_{k,b'}(c)]$ upto error ϵ for every k and b with probability $1 - \frac{1}{2^\lambda}$ independent of m_0, m_1, where $\epsilon \in \frac{1}{\mathsf{poly}(\lambda)}$ would be chosen later.

As discussed above, for $b' = b$, there would exist a key k such that

$$\mathbb{E}_{c \sim C_b} \Pr[V_{k,b'}(c)] = \Pr[\mathcal{Dec}(k, C_b) = m_{b'}] \geq 1 - \mathsf{negl}(\lambda)$$

where $\mathsf{negl}(\lambda)$ is some negligible function. Similarly, for $b' = 1 - b$, with probability atleast $\frac{1}{2}$,

$$\mathbb{E}_{c \sim C_b} \Pr[V_{k,b'}(c)] = \Pr[\mathcal{Dec}(k, C_b) = m_{b'}] \leq 1 - \frac{1}{p}, \forall k \in \{0,1\}^w.$$

The adversary upon estimating $\mathbb{E}_{c \sim C_b} \Pr[V_{k,b'}(c)]$ for each k, b, outputs $\tilde{b}$ for which there exists a k for which $\mathbb{E}_{c \sim C_b} \Pr[V_{k,\tilde{b}}(c)] \geq 1 - \mathsf{negl}(\lambda)$. Hence, given the tomography succeeded, with probability at least $\frac{1}{2}$, the adversary distinguishes with probability $\frac{1}{p} - 2\epsilon - \mathsf{negl}(\lambda)$. Therefore, the adversary distinguishes with probability at least

$$(1 - 2^{-\lambda})\frac{1}{2} \cdot \left(\frac{1}{p} - 2\epsilon - \mathsf{negl}(\lambda)\right),$$

which is non-negligible if we chose $\epsilon \in \frac{1}{\mathsf{poly}(\lambda)}$ small enough. □

Lastly, for statistically binding bit-commitment schemes, it is known that they cannot exist unconditionally, see [May97, LC97].