Post-Quantum Insecurity from LWE

Alex Lombardi* Ethan Mook† Willy Quach‡ Daniel Wichs§

July 3, 2022

Abstract

We show that for many fundamental cryptographic primitives, proving classical security under the learning-with-errors (LWE) assumption, does *not* imply post-quantum security. This is despite the fact that LWE is widely believed to be post-quantum secure, and our work does not give any evidence otherwise. Instead, it shows that post-quantum insecurity can arise inside cryptographic constructions, even if the assumptions are post-quantum secure.

Concretely, our work provides (contrived) constructions of pseudorandom functions, CPA-secure symmetric-key encryption, message-authentication codes, signatures, and CCA-secure public-key encryption schemes, all of which are proven to be classically secure under LWE via black-box reductions, but demonstrably fail to be post-quantum secure. All of these cryptosystems are stateless and non-interactive, but their security is defined via an interactive game that allows the attacker to make oracle queries to the cryptosystem. The polynomial-time quantum attacker can break these schemes by only making a few *classical* queries to the cryptosystem, and in some cases, a single query suffices.

Previously, we only had examples of post-quantum insecurity under post-quantum assumptions for stateful/interactive protocols. Moreover, there appears to be a folklore intuition that for stateless/non-interactive cryptosystems with black-box proofs of security, a quantum attack against the scheme should translate into a quantum attack on the assumption. This work shows otherwise. Our main technique is to carefully embed interactive protocols inside the interactive security games of the above primitives.

As a result of independent interest, we also show a 3-round *quantum disclosure of secrets (QDS)* protocol between a classical sender and a receiver, where a quantum receiver learns a secret message in the third round but, assuming LWE, a classical receiver does not.

*MIT. E-mail: alexjl@mit.edu. Supported in part by DARPA under Agreement No. HR00112020023, a grant from MIT-IBM Watson AI, a grant from Analog Devices, a Microsoft Trustworthy AI grant, the Thornton Family Faculty Research Innovation Fellowship and a Charles M. Vest fellowship. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

†Northeastern. E-mail: mook.e@northeastern.edu.

‡Northeastern. E-mail: quach.w@northeastern.edu.

§Northeastern and NTT Research. E-mail: wichs@ccs.neu.edu. Research supported by NSF grant CNS-1750795, CNS-2055510 and the Alfred P. Sloan Research Fellowship.

Contents

1 Introduction

Recent years have seen tremendous investment and progress in quantum computing (e.g., [AAB+19]), raising our hopes and fears that quantum computing may one day become a reality. The fear is due to the fact that the public-key cryptosystems in use today, based on the hardness of factoring and discrete-logarithms, are known to be efficiently breakable by quantum computers. This brought about the search for *post-quantum secure* cryptosystems that would remain unbreakable even by quantum computers, and there is an ongoing NIST competition to standardize such cryptosystems [NIS]. While there are several candidates, arguably the most appealing ones are based on the *learning with errors (LWE)* assumption [Reg05], which is widely believed to be post-quantum secure. The LWE assumption is also extremely versatile and enables us to construct many types of advanced cryptosystems, such as fully homomorphic encryption [Gen09, BV11], attribute-based encryption [GVW13], and more.

Post-Quantum Security of Cryptosystems? While the post-quantum security of LWE itself has been well studied, the post-quantum security of the various cryptosystems based on LWE has been given considerably less scrutiny. In general, one can ask:

> When does classical security under a post-quantum assumption imply post-quantum security?

For example, is it the case that cryptosystems (encryption, signatures, PRFs, etc.) with classical black-box proofs of security under LWE[1] are also guaranteed to be post-quantum[2] secure? At first glance, it may seem that this should generally hold, based on the following reasoning: black-box reductions should be oblivious to the computational model and should therefore work equally well for classical attackers and quantum attackers. In particular, a black-box reduction should convert any attack on the cryptosystem, whether classical or quantum, into an equivalent attack on the underlying assumption.

Post-Quantum Insecurity for Protocols. Unfortunately, the above intuition is not rigorous and fails on closer inspection. The most glaring reason for this is due to *rewinding* in the context of interactive protocols.

A classical black-box security reduction for interactive protocols can (and typically does) rewind the adversary and restore its state to some earlier point in the execution. While this is a valid form of analysis for classical adversaries, we cannot always rewind and restore the state of a quantum adversary. In particular, if the adversary performs some measurements on its internal quantum state during the protocol execution, then this can destroy the state in a way that makes it impossible to restore.

The issue of rewinding has been known for some time in the context of establishing zero knowledge [vdG97, Wat06] and computational soundness [Unr12, ARU14, Unr16] for interactive proofs/arguments. For example, it was recognized that classical black-box security proofs of zero-knowledge do not appear to generically translate to the post-quantum setting; instead, there has been much recent work trying to understand and prove the security of specific interactive protocols [Wat06, BS20, CCY21, CMSZ21, LMS21] by relying on substantially more complex techniques.

We highlight that this issue is not merely a limitation of our security analysis; we can also provide explicit examples of interactive protocols that are classically secure under LWE, but are demonstrably not post-quantum secure. One way to see this is by considering "interactive proofs of quantumness" (IPQs) [BCM+18]. An IPQ is an interactive protocol consisting of classical communication between a (potentially quantum) prover and a classical verifier, such that there is an efficient *quantum prover* that causes the verifier to accept at the end of the protocol, but no efficient *classical prover* should be able to do so with better

[1] The same question could also be asked for cryptosystems based on any of the other candidate post-quantum assumptions such as isogenies or even post-quantum secure one-way functions or collision-resistant hashing. We frame our discussion in terms of LWE for concreteness and because our eventual results specifically rely on LWE.

[2] We focus on "post-quantum security", where only the adversary is quantum, but all interaction with the cryptosystem is classical. We distinguish this from what is sometimes called "quantum security" [Zha12a], where the cryptosystem needs to also accept quantum inputs. For the latter, it is already known that, e.g., allowing an adversary quantum query access to a PRF may compromise security. We discuss this in detail in Section 1.2.

than negligible probability. In other words, an IPQ is precisely an example of an interactive protocol that is classically computationally sound but quantumly unsound. We have constructions of IPQs from LWE with 4 rounds of interaction [BCM+18, KLVY22], where classical soundness is proved via a black-box reduction from LWE using rewinding. It is easy to embed such IPQs inside other interactive cryptosystems, such as zero-knowledge proofs or multi-party computation protocols, to get constructions that are classically secure under LWE, but are demonstrably post-quantum insecure.

What about non-interactive cryptography? So far, we have seen that rewinding poses a problem for post-quantum security of interactive protocols. However, it may appear that such examples of post-quantum insecurity under post-quantum assumptions are limited to the interactive setting. Can this phenomenon also occur in non-interactive cryptographic primitives such as pseudorandom functions, encryption, signatures etc.? One might expect that this should not be possible. After all, the only reason we have seen primitives fail to inherit post-quantum security is due to rewinding, and rewinding does not appear to come up for non-interactive primitives.

1.1 Our Results

In this work, we show that the above intuition is wrong! We provide explicit (contrived) examples of many of the most fundamental cryptographic primitives, including pseudorandom functions (PRFs), CPA-secure symmetric-key encryption, message-authentication codes (MACs), signatures, and CCA-secure public-key encryption schemes, all of which are proven to be classically secure under LWE via a black-box reduction, but demonstrably fail to be post-quantum secure.

These primitives are qualitatively different from interactive protocols such as zero-knowledge proof systems. First of all, the primitives are stateless – they maintain a secret key, but do not keep any other state between operations. Second of all, the basic operations (e.g., PRF evaluation, encryption, decryption, signing, verifying) are non-interactive. However, the security of these primitives is defined via an interactive game that allows the attacker to make oracle queries to the cryptosystem (e.g., PRF queries, encryption queries, decryption queries, signing queries). The quantum attacker can keep internal quantum state, but can only query the cryptosystem on classical inputs. We show that even these cryptosystems may be insecure against quantum attacks, despite having provable classical security under LWE.

Concretely, we give the following constructions under the LWE assumption:

- A PRF scheme that is classically secure in the standard sense, but broken by a quantum adversary making 3 classical PRF queries. If we consider a PRF with *public parameters* (e.g., the adversary gets some public parameters that depend on the secret key at the beginning of the game) then we get a scheme that can be quantumly broken with only 2 queries.[3]

- A symmetric-key encryption scheme that is classically CPA-secure in the standard sense, but broken by a quantum adversary making 2 encryption queries before seeing the challenge ciphertext. If we consider symmetric-key encryption with public parameters, then we get a scheme that is broken by a quantum adversary making just 1 encryption query before seeing the challenge ciphertext.

- A MAC that is classically secure in the standard sense, but broken by a quantum adversary making 2 authentication queries. If we consider a MAC with public parameters, then we get a scheme that is quantumly broken with just 1 authentication query.

- A signature scheme that is classically secure in the standard sense, but broken by a quantum adversary making 2 signing queries.

- A public-key encryption scheme that is classically CCA-2 secure in the standard sense, but is broken by a quantum adversary making 2 decryption queries before seeing the challenge ciphertext.

[3]Note that PRFs (and other symmetric-key primitives) with public parameters are natural to consider; for instance, the group-based PRFs (e.g., [NR97]) would naturally have public parameters that include a description of the group.

Additional Counterexamples for *one-time* cryptography. Using a modified technique, we construct further examples of schemes that are quantumly broken using even *a single classical query*, but are also only classically secure for a single query:

- A PRF scheme with public parameters that is classically but not post-quantum secure against an adversary making a single query.
- A one-time symmetric-key encryption scheme (i.e., the adversary only gets a single challenge ciphertext) with public parameters that is classically but not post-quantum secure.
- A one-time signature scheme that is classically but not post-quantum secure.
- A bounded-CCA public-key encryption scheme that is classically but not post-quantum secure against an adversary making a single decryption query.

These examples are incomparable to the previous ones, since they give a more dramatic demonstration of post-quantum insecurity with minimal interaction, but they also only satisfy a limited form of classical security against a bounded number of queries. We view these examples as particularly surprising: a one-time signature scheme seems *very* non-interactive, so how can we distinguish between classical and quantum attacks?

Our Techniques. All of our examples are constructed by carefully embedding instances of interactive quantum advantage — either an IPQ or a new protocol that we call "quantum disclosure of secrets" (QDS) — into stateless/non-interactive cryptographic primitives. The key conceptual insight is that although the primitives we consider are non-interactive, the corresponding security games are interactive, allowing us to use a quantum attacker that wins an IPQ to also win in the security game of the given primitive. The classical security of our constructions follows via a black-box reduction that rewinds the adversary, which is the underlying reason that it fails to translate into the quantum setting.

Towards showing the above results, we also develop new ways of demonstrating quantumness that may be of independent interest. Firstly, we observe that the known 4-round IPQs also satisfy *resettable soundness* against classical provers that can arbitrarily rewind the verifier to earlier points in the execution. Using this observation, we construct a stateless/deterministic *quantum advantage function* F_{sk} keyed by some secret key sk that is generated together with some public parameters pp: an efficient classical attacker given pp and oracle access to F_{sk} cannot cause it to ever output a special "accept" symbol (in fact, cannot even distinguish it from a random function), while a quantum attacker can do so by only making 2 classical queries.

Secondly, we construct a 3-round *quantum disclosure of secrets (QDS)* protocol between a classical sender that has some message m and a receiver, where a classical receiver does not learn anything about m during the protocol (assuming LWE), while a quantum receiver learns m at the end of the protocol. This gives a kind of interactive quantum advantage *in three rounds*, despite the fact that interactive proofs of quantumness in three rounds are not known under post-quantum assumptions (e.g., LWE) in the plain model. This primitive is used to prove our second slate of results. Our QDS protocol makes essential use of the recent quantum advantage technique of [KLVY22].

We give a more detailed description of our techniques in Section 2.

Conclusion: Counterexamples in Cryptography. This paper provides counterexamples to the folklore belief that classical proofs of security under post-quantum assumptions (e.g., LWE) imply post-quantum security for basic cryptographic primitives, including PRFs, symmetric/public-key encryption, and signatures. To do so, we construct schemes that are classically secure under LWE but demonstrably fail to be post-quantum secure. Why are we putting effort into constructing schemes that *fail* to be post-quantum secure? This result fits into a broader and important area of cryptography that provides demonstrable counterexamples to intuitive but incorrect beliefs that certain forms of security should generically hold. Other examples of such results include counterexamples for the random-oracle heuristics [CGH98, Bar01, GK03], circular security [Rot13, KRW15, GKW17, WZ17], selective-opening attacks [DNRS99, HRW16], hardness

amplification [BIN97, DJMW12, BIK+22], security composition [GK96, DNRS99], etc. Such counterexamples are extremely important and serve as a warning that can hopefully prevent us from making such mistakes in the future. Having a demonstrable counterexample is much more convincing than just pointing out that our intuition for why security should hold is flawed. Counterexamples also point to specific pitfalls that need to be avoided if we want to prove security. They enhance our understanding of otherwise elusive topics. Lastly, they often lead to new techniques that tend to find positive applications down the line.

1.2 Related Work

One of the primary goals of the study of quantum computation is to understand which tasks can be solved efficiently by quantum computers but not by classical ones. This is informally referred to as a *quantum advantage*. Many instances of quantum advantage have implications for the security of classical cryptography; the implications will typically hold in the particular computational model specified by the kind of quantum advantage obtained. We list a few examples below.

Shor's Algorithm. [Sho94] gives a quantum polynomial-time algorithm for factoring integers and computing discrete logarithms in finite cyclic groups with computationally efficient group operations. This renders typical cryptosystems based on discrete logarithms, factoring, or RSA-type assumptions broken in quantum polynomial time.

Interactive Proofs of Quantumness. As discussed above, [BCM+18, KMCVY21, KLVY22] give surprising examples of interactive quantum advantage under LWE, *despite* the fact that LWE is believed to be hard for efficient quantum algorithms. They construct interactive protocols where an honest quantum prover causes the verifier to accept, but any efficient classical prover cannot cause the verifier to accept assuming the hardness of LWE. This immediately implies that certain interactive protocols can be classically secure under LWE but quantumly insecure.

Counterexamples in the Random Oracle Model. Many cryptosystems are built using a generic "unstructured" hash function H; security is argued in the *random oracle model* [BR94], a model in which the adversary can make only polynomially many *queries* to H (and H is treated as a uniformly random function).

For these schemes, the random oracle model serves as a heuristic indicating that the scheme *might* be secure when instantiated with a good concrete hash function. However, when quantum attacks on the scheme are considered, a serious problem arises [BDF+11]: given a concrete hash function H, a quantum algorithm can query H *in superposition* (that is, compute the unitary map $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle$ on an arbitrary input state). Thus, to heuristically capture security of these schemes against quantum attacks, one should prove security in the *quantum random oracle model* (QROM), in which the adversary can make polynomially many superposition queries (rather than classical queries).

Prior work [BDF+11, YZ21, ZYF+20, YZ22] has constructed examples of cryptosystems, defined relative to an arbitrary hash function H, that are secure in the classical random oracle model (possibly under an additional computational assumption) but insecure in the QROM. For example, [YZ21] construct encryption and signature schemes that are secure in the ROM but not the QROM, while [YZ22] even constructs such examples for one-way functions!

We note that counterexamples for ROM cryptosystems are fundamentally different from what we are asking in this work. ROM vs. QROM separations highlight the insufficiency of the classical ROM for accurately describing the security of hash function-based cryptosystems against quantum attacks. And at the technical level, the ROM "has room" for counterexamples by embedding an *oracle separation* between classical and quantum computation, which may even be unconditional. Of course, ROM based examples also translate into plain model examples that are quantum insecure and heuristically classically secure when instantiated with a good hash function. For example, [YZ22] gives a construction of a one-way function with this property. However, the classical security of the resulting one-way function is only heuristic and does not

appear to be provable under any standard post-quantum assumption such as LWE. Indeed, since one-wayness is defined via a completely non-interactive security game with no room for rewinding, if one had a black-box reduction showing one-wayness under LWE, then it would also imply the post-quantum insecurity of LWE (at least in the uniform setting without [quantum] auxiliary input, see discussion on [BBK22] below). In contrast, our work shows quantum insecurity for primitives whose classical security is proved under LWE using a black-box reduction.

Quantum Oracle Queries in the Security Game. When the security game underlying a cryptographic primitive involves giving an adversary *oracle access* to some functionality (such as a PRF), the natural definition of post-quantum security is to consider a quantum attacker breaking a cryptosystem used by classical honest users who perform operations on classical inputs. Modeling this corresponds to a security game where the attacker is restricted to querying the oracle on classical inputs. However, one could imagine a stronger notion of "quantum security" [Zha12a], where even the honest users want to perform cryptographic operations on quantum inputs, in which case we need to give the adversary quantum oracle access.

In these situations, classical security proofs do not generically carry over to the quantum query setting, and there often exist counterexample protocols that are secure against adversaries that make classical queries but *insecure* in the presence of quantum queries [Zha12b, Zha12a, BZ13a, BZ13b].

On the other hand, in this work we are interested in understanding whether there are quantum attacks on *classical* cryptosystems that only operate on classical inputs, and therefore the above counterexamples do not apply.

Quantum Auxiliary Input. The recent work of [BBK22] noticed that rewinding may be an issue even for completely non-interactive security games (e.g., one-way functions or pseudorandom generators), if one considers a setting where a non-uniform adversary may have quantum auxiliary input. They provide techniques for showing that certain (but not all) forms of classical rewinding-based reductions do in fact carry over to the quantum setting. While they provide some examples were their techniques fail, it does not translate into an overall example showing insecurity. It would be extremely interesting to see if one can come up with examples of (e.g.,) one-way functions that are proven secure classically via a black-box reduction under a post-quantum assumption, but are not secure in the quantum setting with quantum auxiliary input.

2 Technical Overview

Our main technique in constructing cryptographic primitives that are classically secure but post-quantum insecure is to embed interactive proofs of quantumness (IPQs) [BCM+18, KMCVY21, KLVY22] based on LWE inside these primitives. Such IPQs consist of 4-message interactive protocols, where the verifier sends the first message and the prover sends the last message. The main difficulty is that IPQs are stateful/interactive protocols, while the primitives we consider are stateless/non-interactive.

For concreteness, let's start with *signature schemes* as an illustrative example, but we will later explain how to extend the ideas all the other primitives as well.

Stateful Signatures. As a start, let's relax the standard notion of signatures to allow the signing algorithm to be stateful. Then we can take any standard signature scheme (under LWE) and easily augment it to incorporate an IPQ as follows. In addition to signing the messages with the standard signature scheme, our augmented signing algorithm also runs the verifier of an IPQ on the side. It interprets any messages to be signed as prover message in an IPQ and appends the appropriate verifier responses to the signatures (the verification algorithm of the augmented signature scheme simply ignores these appended values). Since the IPQ verifier is stateful, this also requires the signing algorithm to be stateful. If at any point in time the IPQ verifier accepts, then the signing algorithm simply appends the secret key of the signature scheme to the signature.

It is easy to see that the above augmented signature scheme is classically secure under LWE, since a classical adversary making signing queries will be unable to get the IPQ verifier to accept. It is also easy

to see that the scheme is insecure against a quantum attacker who acts as the quantum prover in an IPQ, causes it to accept, and recovers the secret key of the signing algorithm, which it then uses to construct its forgery. If we use a 4-message IPQ and append the initial verifier message to the verification key of the signature, then the above attack corresponds to making 2 signing queries.

Stateless signatures. Unfortunately, the above idea seems to crucially rely on having a stateful signing algorithm, and our goal is to extend it to the stateless setting. To do so, we essentially construct an IPQ with a stateless verifier and resettable security: even if the classical prover can reset the verifier and run it many times with different prover messages, it cannot cause the verifier to accept.

We rely on the fact that the 4-message IPQs of [BCM+18, KLVY22] have special structure. The first round is secret-coin and the verifier generates an initial message v_1 together with some secret state st and sends v_1. The prover responds with p_1. The verifier then uses public-coins to send a uniformly random message v_2 and the prover responds with p_2. At the end of the 4th round, the verifier uses the secret state st to decide if the transcript (v_1, p_1, v_2, p_2) is accepting or rejecting. We observe that we can convert the verifier of such an IPQ (as long as it has negligible soundness error) into a deterministic/stateless IPQ verifier V_{sk} that just maintains a secret key $\mathsf{sk} = (v_1, \mathsf{st}, k)$ consisting of the first round verifier message v_1 of the original IPQ, the secret st, and a key k for a PRF f_k. We define the function V_{sk} as follows:

- On input the empty string, output v_1.
- On input p_1, output $v_2 = f_k(p_1)$.
- On input p_1, p_2, compute $v_2 = f_k(p_1)$ and use st to check if (v_1, p_1, v_2, p_2) is an accepting transcript: if so accept, else reject.

An efficient quantum prover with oracle access to V_{sk} can cause it to accept, using the same strategy as in the original IPQ.[4] However, an efficient classical prover with oracle access to V_{sk} cannot cause it to accept, even if it can make arbitrarily many queries on arbitrary inputs, effectively being able to run many executions of the original interactive protocol with rewinding. We show this via a simple reduction where we convert any adversary that causes the stateless IPQ verifier V_{sk} to accept into an adversary on the original stateful IPQ.

We use the above stateless IPQ to derive our counterexample for stateless signatures. We start with any standard signature scheme (secure under LWE) and augment it by incorporating the stateless IPQ as follows. Firstly, we generate the secret key sk of the stateless IPQ verifier V_{sk} as above, and append sk to the original signature secret key $\mathsf{sk}_{\mathsf{Sig}}$. We also append v_1 to the original verification key. We then modify the signing algorithm: we append the output of $V_{\mathsf{sk}}(m)$ to any signature of m, and, if at any point $V_{\mathsf{sk}}(m)$ accepts, then we append the original signature signing key $\mathsf{sk}_{\mathsf{Sig}}$ to the signature. The verification algorithm ignores these appended components.

We have an efficient quantum adversary on this signature scheme by running the quantum prover of the IPQ: the adversary gets v_1 from the verification key and queries the signing algorithm twice, once on p_1 to get v_2 and once on p_1, p_2 to cause the IPQ verifier to accept and recover $\mathsf{sk}_{\mathsf{Sig}}$. At this point, the adversary can forge a signature on any message of its choosing. On the other hand, an efficient classical adversary cannot cause V_{sk} to accept and hence does not learn any additional information about $\mathsf{sk}_{\mathsf{Sig}}$ beyond what it would get in the original signature game. Therefore the above signature scheme is classically secure under LWE, but quantumly broken with just 2 signing queries.

Generalizing: Quantum Advantage Function. We abstract out the above idea of stateless IPQs via a *quantum advantage function* (QAF). A QAF is a deterministic/stateless function F_{sk}, indexed by a secret key sk. A classical polynomial-time adversary with oracle access to F_{sk} can never cause it to output a special accept value (except with negligible probability), while a quantum polynomial-time adversary can cause it to do so by only making 3 classical oracle queries. We can set the QAF $F_{\mathsf{sk}} = V_{\mathsf{sk}}$ to be the stateless IPQ verifier defined above.

[4] Technically, it may be possible that the completeness error of the IPQ increases non-negligibly if the PRF is only classically secure but not post-quantum secure. But it is easy to solve this by relying on a PRF that is one-wise independent.

Alternatively, we can define a QAF with public parameters pp that depend on sk: even given pp a classical polynomial-time adversary with oracle access to F_{sk} can never cause it to output accept, while a quantum polynomial-time adversary given pp can do so by only making 2 classical oracle queries. We can construct such a QAF by setting the public parameters $\mathsf{pp} = v_1$ to be the first verifier message and setting $F_{\mathsf{sk}} = V_{\mathsf{sk}}$ to be the stateless IPQ verifier above.[5]

We can embed our QAF inside various stateless/non-interactive cryptosystems to get our remaining counterexamples:

- Symmetric-key message authentication codes (MAC): Take any existing secure MAC and augment it by running a QAF on the side. The QAF outputs are appended to the tags of the original scheme, and the verification procedure is augmented to automatically accept any message on which the QAF accepts. This gives a classically secure MAC that can be quantumly broken using 2 authentication queries, or alternately, even just 1 authentication query in the setting with public parameters.[6] In particular, the quantum attacker uses the k queries needed to get the QAF to accept ($k = 3$ or $k = 2$ depending on public parameter) as $k - 1$ authentication queries and a forgery.

- CCA-2 secure public-key encryption: Take any existing secure scheme and augment it with a QAF with public parameters as follows. Append the public parameters to the public key of the scheme. Modify encryption to ensure that all valid ciphertexts start with a 0 bit. Modify the decryption procedure so that, it decrypts valid ciphertexts correctly, but if it gets as an invalid ciphertext it evaluates the QAF on it instead of decrpyting. If the QAF ever accepts, the decryption procedure outputs the secret key of the encryption scheme. The scheme remains correct and classically secure, but can be quantumly broken using just 2 decryption queries (made before receiving the challenge ciphertext) to recover the secret key.

- Pseudorandom functions (PRF): We notice that that the outputs our QAF can be either: (i) v_1 which is pseudorandom for known IPQs, (ii) $v_2 = F_k(p_1)$ which is pseudorandom, or (iii) accept/reject. We can modify the QAF so that instead of rejecting it applies an independent PRF. With this modification, a classical attacker cannot distinguish it from a random function, since it cannot cause the original QAF to ever accept. On the other hand, a quantum attacker can easily distinguish, by causing the original QAF to accept, using just 3 queries, or even 2 queries in the setting with public parameters.

- Symmetric-key encryption: Take any existing secure scheme and augment it with a pseudorandom QAF (as constructed in the previous bullet) as follows. When encrypting a message m, choose some fresh randomness r and append r together with the output of the QAF applied on $m||r$ to the ciphertext. If the QAF accepts, also append the secret key of the original symmetric-key encryption to the ciphertext. The decryption algorithm ignores the appended values.

  For classical adversaries, we can rely on the fact that the QAF is pseudorandom (and cannot be caused to accept) to argue that this modification does not break CPA security. For quantum adversaries, we show that it is possible to cause the QAF to accept using 3 CPA queries, or even just 2 CPA queries in the setting with public parameters. There is a minor difficulty that the quantum adversary only gets to pick the left half m of the QAF inputs, while the right half r is chosen randomly. Nevertheless, by starting with an IPQ protocol where we expand prover messages to contain a dummy "right half" that the verifier ignores, we get a QAF that can be efficiently quantumly attacked even if the right half of the inputs is chosen randomly.

One-Time Security and Quantum Disclosure of Secrets. We also give alternate examples of cryptosystems that are classically "one-time" secure, but are not post-quantum one-time secure. As an example, let's consider one-time signatures. The security game for one-time signatures consists of 4 rounds: the

[5] In this case, we can remove the instruction that V_{sk} outputs v_1 on the empty string, since we already give out v_1 in the public parameters.

[6] For symmetric-key primitives in the public-parameter setting, the secret key of the primitive is generated together with some public parameters that are given to the adversary, but are not otherwise needed for correctness.

challenger sends a verification key, the attacker chooses a message, the challenger sends a signature and the attacker produces a forgery. Therefore, there is hope that we can embed a 4-message IPQ into the 4-message security game of one-time signatures. However, we notice that the one-time signature game has an additional feature that we call *public verifiablity*: just by looking at the transcript of the game, an external observer can tell whether the verifier accepted or rejected. On the other hand, the known 4-message IPQs from LWE do not have public verifiability. Therefore, to give a counterexample for signatures, we at the very least need to construct a 4-message *publicly verifiable* IPQ.[7] Alternately, let's consider one-time symmetric-key encryption with public parameters. There, the security game consists of only 3 rounds: the challenger chooses the secret key with public parameters and sends the latter to the attacker, the attacker chooses two messages m_0, m_1 and gets an encryption of m_b. At the end of the 3 rounds the adversary has to distinguish between $b = 0$ and $b = 1$. Therefore, we would need some sort of a 3 round game with *quantum advantage*, where a quantum adversary can distinguish between two possibilities, but a classical one cannot. Current IPQs from LWE all require 4 rounds.

We solve both of the above issues by constructing a new type of 3-message protocol with quantum advantage under LWE, which we refer to as a *quantum disclosure of secrets* (QDS). A QDS is an interactive protocol between a classical *sender* who has some message m and a (potentially quantum) *receiver*. No efficient classical receiver can distinguish between any two possible sender messages m_0, m_1 at the end of the protocol, while a quantum receiver can fully recover m. We construct a 3-message QDS under LWE and we give an overview of this construction further below.[8] For now, let us assume we have such a 3-message QDS, whose execution consists of three messages s_1, r_1, s_2, where s_i denotes sender messages and r_i the receiver message. We use it to get various counterexamples to post-quantum security of one-time primitives under LWE. For simplicity, we just discuss one-time signatures and one-time symmetric-key encryption (with public parameters), but the other counterexamples are all similar:

- One-time Signatures: Take any secure one-time signature scheme and augment it by running a QDS on the side, where the sender's message is set to be the signing key of the original scheme. Append the first message s_1 of the QDS to the verification key and st to the signing key. To sign some message, sign it under the original signature scheme, but also interpret the message as the receiver's message r_1 in the QDS protocol and run the QDS on it to produce the response s_2 (using st), and append s_2 to the signature. The verification algorithm ignores the appended components.

  A classical attacker cannot break one-time security since it does not learn anything about the signing key from the QDS when making one signing query. However, a quantum attacker can break security by recovering the original signing key from the QDS using one signing query, and then can forge the signature of an arbitrary new message.

- One-time Symmetric-Key Encryption (with public parameters): Take any secure one-time encryption (e.g., one-time pad) and augment it with a QDS, where the sender's message is set to be the secret key of the original encryption scheme. Set the public parameters to consist of the first round QDS message s_1 and append st to the secret key. To encrypt a message, use the original one-time encryption scheme, but also interpret the message as the receiver's message r_1 in the QDS protocol and run the QDS on it to produce the response s_2 (using st), and append s_2 to the ciphertext.

  To argue (computational) classical security, we rely on the fact that, for a classical receiver in the QDS, not only is the sender's message hidden but entire sender response s_2 sent in the third round looks pseudorandom. On the other hand, a quantum adversary can recover the key of the original encryption scheme and decrypt.

We note that the 3-message QDS scheme that we construct is *not* resettably secure: if a classical receiver can rewind the sender with many different values of r_1 and get the corresponding values s_2 then it can learn

[7] It is easy to make an IPQ publicly verifiable simply by adding an additional round where the verifier publicly declares whether it accepted or rejected, but this would require 5 rounds and we need 4.

[8] A 3-message QDS also implies a 4-message publicly verifiable IPQ. This is shown implicitly by our one-time signature counterexample below, but can be done more directly as follows. Use a QDS to send a random message x and append a one-way function $f(x)$ to the 3rd round; then accept in the 4th round if the prover replies a valid preimage x' for $f(x)$.

the sender's message. This is the reason that our results above are incomparable to the previous ones and only achieve one-time classical security. If we were able to construct a resettably secure QDS, we would get the best of both worlds and construct schemes that are fullly secure in the standard sense against classical adversaries, but not even one-time secure against quantum adversaries.

Quantum Disclosure of Secrets from LWE. We now give an overview of our construction of 3-message QDS from LWE. Our main idea is to start with a special 4-message IPQ from LWE that has a *unique final answer*: given (v_1, p_1, v_2) and st, the verifier can efficiently compute a unique prover answer p_2 that would cause it to accept. We can convert such a 4-message IPQ into a 3-message QDS. We keep the first two messages of the IPQ and QDS the same with $s_1 = v_1, r_1 = p_1$. Then, in the beginning of the third round, we have the sender choose a random v_2 as the IPQ verifier would, compute the unique correct p_2^* that would make the IPQ verifier accept, take a Goldreich-Levin hardcore bit $GL(p_2^*)$ and use it to one-time pad the sender-message m by setting $s_2 = (v_2, GL(p_2^*) \oplus m)$.[9] By relying on Goldreich-Levin decoding, we can translate any classical attack on the 3-message QDS into a classical attack on the original 4-message IPQ. On the other hand, we can use a quantum attack on the 4-message IPQ to easily recover the message m in the 3-message QDS by computing the correct p_2 from v_2 and then using the hardcore bit of p_2 to un-blind the message.

Therefore, to construct a 3-message QDS, we need to construct a 4-message IPQ with a unique final answer. Unfortunately, the IPQ schemes of [BCM+18] do not have this property (either directly or with any simple modification). On the other hand, the work of [KLVY22] gives a general template for constructing 4-message IPQ schemes. We review this template and show that there is a careful instantiation of it that does have a unique final answer.

The template of [KLVY22] construct a (4-message) IPQ from any 2-prover non-local game. A 2-prover non-local game consists of 2 provers who cannot communicate and are given two questions (q_1, q_2 respectively) sampled from some joint distribution. Their goal is to reply with answers a_1, a_2 respectively, and they win if some relation $R(q_1, q_2, a_1, a_2)$ holds. Such a game has quantum advantage if quantum provers who share entangled quantum state at the beginning of the game can have a noticeably larger winning probability than classical provers who only share classical shared randomness. For example, the CHSH game [CHSH69] sets q_1, q_2, a_1, a_2 to be bits, samples (q_1, q_2) uniformly and independently, and defines $R(q_1, q_2, a_1, a_2)$ to hold if $a_1 \oplus a_2 = q_1 \wedge q_2$. Classical provers can only win with probability .75, but quantum provers can win with probability $\cos^2(\pi/8) > .85$.

The work of [KLVY22] compiles any such game into a 4-message IPQ with a single prover by using quantum fully homomorphic encryption. The verifier sends $v_1 = \mathsf{Enc}(q_1)$ the prover responds with $p_1 = \mathsf{Enc}(a_1)$, the verifier sends q_2 and the prover responds with a_2: the verifier accepts if $R(q_1, q_2, a_1, a_2)$ holds. The good news is that, if we instantiate this template with the CHSH game, then there is a unique final answer $a_2 = (q_1 \wedge q_2) \oplus a_1$. However, the resulting IPQ only has a noticable gap between the success of a classical prover and a quantum one (.75 vs .85), but we want an IPQ where the classical prover only has a negligible success probability while the quantum one can win with all but negligible probability. We can achieve this by using parallel repetition of many copies of the CHSH game and accepting if the prover wins in $> .8$ fraction of them. But now there is no longer a unique final answer that wins the IPQ, since the prover can win any .8 fraction of the games to get the verifier to accept (and even a quantum prover won't be able to win significantly more that .85 fraction)! Instead, we start with a different non-local game, which is a variant of the *magic square game* [Ara02, CHTW04].[10] In this game, there is a unique final answer a_2 determined by q_1, q_2, a_1, and there is a pair of entangled quantum provers that can win with probability 1, while classical provers only win with probability at most 17/18. By taking a sufficiently large parallel repetition and accepting if *all* copies accept, we can drive down the winning probability of classical provers to

[9] This allows us to encrypt a single bit, but we can repeat this in parallel to encrypt a multi-bit message one bit at a time. Security follows via a simple hybrid argument.

[10] We think of a 3×3 square of bits. The challenge q_1 corresponds to a random row or column (6 possibilities) and q_2 corresponds to a random location inside that row/column. The provers are supposed to answer with a_1 being the 3 bits in the given row/column specified by q_1 and a_2 being the bit in the position specified by q_2. They win if the answers are consistent and if the bits of a_1 have parity 0 when q_1 is a row or parity 1 when q_1 is a column.

negligible, while allowing quantum provers to win with probability 1 and preserving a unique final answer a_2 determined by q_1, q_2, a_1. Therefore, if we apply the [KLVY22] framework with the parallel-repeated variant of Magic Square as above, we get a 4-message IPQ with a unique final answer as desired.[11]

3 Open Problems

We mention several fascinating open problems left by our work.

- Can we construct a CPA-secure public-key encryption scheme which is classically secure under LWE but post-quantum insecure? The CPA security game for public-key encryption consists of 3 rounds, so it may seem like we should be able to embed a QDS scheme inside it. But the 3rd round of the CPA security game must be publicly computable from the first 2 rounds, while our QDS requires secret state to compute the 3rd round.
- Can we construct a 3-message stateless/resettable QDS under LWE? This would allow us to construct cryptosystems that are classically secure in the standard sense under LWE, but fail to be even one-time post-quantum secure.
- Can we construct IPQs and classically secure / quantum-insecure cryptosystems under other plausibly post-quantum assumptions beyond LWE? Ideally we would even be able to do so under generic assumptions, such as one-way functions.
- Can we construct 3-message (resettably secure) IPQs from LWE? This would allow us to get rid of the public parameters in our symmetric-key examples.
- Inspired by [BBK22], can we construct one-way functions under post-quantum assumptions (e.g., LWE), where the one-way function is classically secure, but post-quantum insecure given quantum auxiliary input? As noted in [BBK22], this may be possible even if classical security is proven via a black-box reduction.
- Can we construct one-way functions under a post-quantum assumptions (e.g., LWE), where the one-way function is classically secure but post-quantum insecure, even without quantum auxiliary input? Since the security game of one-way function is non-interactive, there is no possibility of rewinding distinguishing between classical and quantum adversaries. Therefore, the classical security of such one-way functions could not be proven via a black-box reduction. Could we perhaps have such an example nevertheless by using a non-black-box reduction?

4 Preliminaries

We use QPT to denote quantum polynomial time and PPT to denote classical probabilistic polynomial time. We say that a function $f(n)$ is *negligible* if for all constants $c > 0$, $f(n) < n^{-c}$ for all but finitely many n.

Lemma 4.1 (Goldreich-Levin Decoder)**.** *There exists a polynomial-time oracle algorithm* $\mathsf{Decode}^{\mathcal{O}}(1^B, 1^n)$ *satisfying the following property: Let* Pred *be any algorithm such that*

$$\Pr\left[\mathsf{Pred}(r) = \langle x, r\rangle\right] = \frac{1}{2} + \varepsilon$$

[11] Unfortunately, if we use this 2-prover non-local game, then the resulting 4-message IPQ cannot be made resettably sound. This is because the challenge q_2 gives information about q_1. By rewinding the verifier and seeing many values of q_2, a classical adversary can learn q_1 and win the game. (Even if the 4-message IPQ was resettably sound, it wouldn't guarantee that the 3-message QDS would be, because it reveals various GL bits in the 3rd round.) In contrast, in the original instantiation of the [KLVY22] framework with the CHSH game and threshold parallel repetition, the resulting 4-message IPQ does not have unique final answers, but can be given resettable security using a PRF to generate q_2, because q_2 is random and independent of q_1.

for some fixed string $x \in \{0,1\}^n$*. Then, for any* $B \geq 1/\varepsilon$*,*

$$\Pr\left[\mathsf{Decode}^{\mathsf{Pred}(\cdot)}(1^B, 1^n) = x\right] \geq \mathrm{poly}(\varepsilon, 1/n)$$

for some fixed polynomial poly*. In particular, if* $\varepsilon = \varepsilon(n) \geq 1/n^c$ *for some constant* c *and infinitely many* n*, by setting* $B = n^c$ *we obtain that* $\Pr\left[\mathsf{Decode}^{\mathsf{Pred}(\cdot)}(1^B, 1^n) = x\right]$ *is non-negligible in* n*.*

4.1 Two Magic Square Games

In this section, we recall two variants of the "magic square game" [Ara02, CHTW04], which are both two-prover non-local games [CHTW04]. These are games played by two provers P_1, P_2 and a verifier V, in which:

- V samples a pair of (classical) questions (q_1, q_2).
- V sends q_1 to P_1 and q_2 to P_2.
- P_1 and P_2 send (classical) answers a_1, a_2 to V respecively.
- V computes a decision predicate $V(q_1, q_2, a_1, a_2)$ denoting whether the provers win or lose.

We consider two computational models for P_1 and P_2: local prover strategies are strategies in which a_1 and a_2 are, respectively, functions of q_1 and q_2 alone (possibly also using some pre-shared classical randomness). $\mathsf{entangled}$ prover strategies are strategies in which P_1, P_2 may share an entangled state $|\psi\rangle_{\mathcal{AB}}$ (where P_1 has register $\mathcal{A}$ and P_2 has register $\mathcal{B}$) and can compute their answers by applying some (POVM) measurements to their respective registers. The "entangled value" and "local value" of a game $\mathcal{G}$ is defined to be the maximum (or supremum) value of all entangled and local strategies, respectively.

Definition 4.2 (Independent Question Magic Square Game)**.** *The independent question magic square game is a two-prover nonlocal game in which* q_1, q_2 *are samplied i.i.d. from* $\{1,2,3\}$*, answers* a_1, a_2 *are in the set* $\{0,1\}^3$*, and the verifier accepts if the following three conditions hold:*

- $(a_1)_{q_2} = (a_2)_{q_1}$ *("row* q_1*" is consistent with "column* q_2*"),*
- $(a_1)_1 \oplus (a_1)_2 \oplus (a_1)_3 = 0$ *(rows have even parity), and*
- $(a_2)_1 \oplus (a_2)_2 \oplus (a_2)_3 = 1$ *(columns have odd parity).*

Theorem 4.3 ([CHTW04])**.** *The independent question magic square game has entangled value* 1 *and local value* 8/9*.*

Definition 4.4 (Unique Answer Magic Square Game)**.** *The unique answer magic square game is a two-prover nonlocal game in which* $(q_1, q_2) \in (\{0,1\} \times \{1,2,3\}) \times \{1,2,3\}^2$ *are sampled uniformly from the subset* $\{(b, u, v_0, v_1) \in \{0,1\} \times \{1,2,3\}^3 : v_b = u\}$*. An answer* a_1 *is required to be in the set* $\{0,1\}^3$ *while an answer* a_2 *is required to be in the set* $\{0,1\}$*. Finally, the verifier accepts if the following two conditions hold:*

- $(a_1)_1 \oplus (a_1)_2 \oplus (a_1)_3 = b$ *for* $q_1 = (b, u)$*.*
- $(a_1)_{v_{1-b}} = a_2$*, for* $q_1 = (b, u)$ *and* $q_2 = (v_0, v_1)$*.*

The relationship between the two magic square game variants is that the "unique answer" P_1 is being asked to behave as *either* P_1 or P_2 from the "independent question" variant, while the "unique answer P_2" is asked to provide the single-bit intersection of the "independent question" answers a_1, a_2. In particular, given (q_1, a_1, q_2), there is an unique answer a_2 that makes the verifier accept.

Theorem 4.5 ([CHTW04])**.** *The unique answer magic square game has entangled value* 1 *and local value* 17/18*.*

Finally, we note that by [Raz95], if either of the above two games is repeated t times in parallel, the local value of the repeated game is at most $2^{-\Omega(t)}$.

4.2 The [KLVY22] Compiler

We briefly recall the cryptographic compiler of [KLVY22] that converts 2-prover non-local games into *interactive* single-prover games:

Definition 4.6 (KLVY Compiler)**.** *For any 2-prover non-local game $\mathcal{G}$ and any encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, we can define the following game between a* single *prover and verifier:*

- *The verifier samples a key k for the encryption scheme, along with question pair (q_1, q_2) for $\mathcal{G}$. The verifier sends $\mathsf{Enc}(k, q_1)$ to the prover.*
- *The prover responds with some ciphertext $\hat{a}_1$.*
- *The verifier sends q_2 to the prover.*
- *The prover responds with some answer a_2.*
- *The verifier computes $a_1 = \mathsf{Dec}(k, \hat{a}_1)$ and checks the $\mathcal{G}$ predicate $V(q_1, q_2, a_1, a_2)$.*

Theorem 4.7 ([KLVY22])**.** *Under the LWE assumption, there exists an encryption scheme such that:*

- *If $\mathcal{G}$ has entangled value $1 - \mathsf{negl}(\lambda)$, then the compiled game has quantum value $1 - \mathsf{negl}(\lambda)$.*
- *If (1) $\mathcal{G}$ is obtained by λ-wise parallel repetition of a constant-size game with local value < 1, and (2) LWE is classically hard, then the compiled game has negligible classical value.*

4.3 Interactive Proofs of Quantumness

For concreteness and simplicity of notation, we will focus throughout this work on interactive proofs of quantumness with 4 messages in total. Note that this corresponds to the best round complexity known for interactive proofs of quantumness in the plain model.

Definition 4.8. *An* interactive proof of quantumness *is an interactive protocol Π between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, with the following properties:*

- *Quantum completeness: there exists a efficient quantum prover $\mathcal{P}$ such that:*

$$\Pr\left[(\mathcal{P}, \mathcal{V})(1^\lambda) = 1\right] \geq 1 - \mathrm{negl}(\lambda).$$

- *Classical soundness: for any efficient classical prover $\mathcal{P}^*$:*

$$\Pr\left[(\mathcal{P}^*, \mathcal{V})(1^\lambda) = 1\right] \leq \mathrm{negl}(\lambda).$$

Let v_1, v_2 (resp. p_1, p_2) denote the messages sent by the verifier (resp. the prover) during the execution of an interactive proof of quantumness Π.

An interactive proof of quantumness can furthermore satisfy the following optional properties:

1. Public-coin second verifier message: the second verifier message v_2 consists of uniformly and independently sampled random coins.
2. (Classically) Pseudorandom verifier messages: for any efficient classical prover $\mathcal{P}^*$, the messages (v_1, v_2), output by the verifier in a protocol execution with $\mathcal{P}^*$, are computationally indistinguishable from uniformly random strings, even if $\mathcal{P}^*$ learns the outcome of the execution.[12]

[12] Allowing $\mathcal{P}^*$ to learn the outcome of the protocol execution is without loss of generality by negligible classical soundness: all executions of the protocol with $\mathcal{P}^*$ will be rejected with overwhelming probability.

3. Unique final answer: given any partial transcript $\tau = (v_1, p_1, v_2)$ and any verifier state st, there exists an efficient algorithm $\mathsf{UniqueAnswer}(v_1, p_1, v_2, \mathsf{st}) \to p_2^* \in \{0,1\}^\ell$ which outputs the unique final prover message that can make the verifier accept (namely, output 1) if such a final prover message exists.

We will make use of constructions of two different interactive proofs of quantumness in this paper:

Lemma 4.9. *Under the LWE assumption, there exists a 4-message interactive proof of quantumness satisfying properties 1 (public-coin second verifier messages) and 2 (classically pseudorandom verifier messages) (Definition 4.8).*

Proof. For simplicity,[13] we appeal to Theorem 4.7 ([KLVY22], Theorem 3.7) using a λ-wise parallel repetition of the independent question magic square game Definition 4.2. By inspecting the [KLVY22] protocol (Definition 4.6), we note that if the queries in the non-local game are independent and uniformly random, then the message v_2 in this protocol is public coin. Finally, what remains is to argue that the first verifier message v_1 is pseudorandom: in the [KLVY22] scheme, the message v_1 is an encryption of a query q_1, so as long as the encryption scheme has pseudorandom ciphertexts (as it does holds for the schemes used in Theorem 4.7 [Mah18, Bra18]), v_1 is pseudorandom. □

We will also use a proof of quantumness with unique answers (while still requiring completeness $1-\mathrm{negl}(\lambda)$ and negligible soundness). While we are not aware of any explicit constructions satisfying this property in the literature, we observe that instantiating [KLVY22] with an appropriate non-local game gives such a proof of quantumness.

Lemma 4.10. *Under the LWE assumption, there exists a 4-message interactive proof of quantumness satisfying properties 2 (classically pseudorandom verifier messages) and 3 (unique final answers) (Definition 4.8).*

Proof. We again appeal to Theorem 4.7. In this case, we note that the resulting 4-message interactive protocol has a unique final answer provided that the non-local game has a unique answer a_2 conditioned on partial transcript (q_1, a_1, q_2). Thus, we instantiate the non-local game with the unique answer magic square game (Definition 4.4). The queries q_1, q_2 are *not* independent in this game, but the resulting 4-message protocol has pseudorandom verifier messages because the marginal distribution on v_2 is uniformly random, and the verifier's first message v_1 is pseudorandom even conditioned on v_2 (and the prover's messsages). □

5 Deterministic Oracles with Quantum Advantage

5.1 Quantum Advantage for Unbounded-Classical Query Algorithms

We introduce quantum advantage functions, which are by default stateless and deterministic functions that demonstrate a quantum advantage given only classical query access. In its stronger form, such a function acts as a pseudorandom function against classical adversaries.

Definition 5.1 (Quantum Advantage Functions)**.** *A* quantum advantage function family *is a pair of efficient algorithms* $(\mathsf{Setup}, F_{\mathsf{sk}})$ *with the following syntax:*

- $\mathsf{Setup}(1^\lambda)$*: sample some public parameters* pp*, a secret key* sk *and outputs* $(\mathsf{pp}, \mathsf{sk})$*. Without loss of generality, we will consider throughout the paper that* sk *includes the public parameters* pp*.*
- $F_{\mathsf{sk}}(\cdot)$*: on input a message* x*, either output a message* y*, or a special "accept" symbol denoted* accept*, or a special "reject" symbol denoted* reject*. We require by default that* F_{sk} *is stateless and deterministic.*

[13] An alternative approach is to use [BCM+18]-style proofs of quantumness. However, we want a 4-message proof of quantumness with negligible soundness error (ideally based solely on the classical hardness of LWE), and there does not seem to be an explicit construction of such a variant in the literature. However, the recent work [LLQ22] suggests an approach along these lines [LLQ]: first, consider a "repeated" variant of the [BCM+18] protocol that still has one-bit challenges (which will e.g. have soundness error 1/2 rather than 3/4), and then argue that parallel repetition amplifies the soundness of this protocol. Finally, another viable approach is to apply a "random-terminating parallel repetition" [Hai09] to the original [BCM+18] protocol. We stick with [KLVY22] for the construction that follows most directly from the literature.

We additionally require the following properties:

1. *(k-Quantum easiness) There exists a QPT oracle algorithm $\mathcal{A}^{F(\cdot)}(\mathsf{pp})$ such that:*

$$\Pr\left[\mathcal{A}^{F_{\mathsf{sk}}(\cdot)}(\mathsf{pp}) = x^* \ \wedge \ F_{\mathsf{sk}}(x^*) = \mathsf{accept}\right] = 1 - \mathrm{negl}(\lambda),$$

where $\mathcal{A}^{F_{\mathsf{sk}}(\cdot)}(\mathsf{pp})$ makes k classical *oracle queries in total to $F_{\mathsf{sk}}(\cdot)$ before outputting x^*,and where the probability is over $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. We simply say that $(\mathsf{Setup}, F_{\mathsf{sk}})$ satisfies* quantum easiness *if it satisfies 1-quantum easiness.*

2. *(Classical hardness) For all PPT oracle algorithms $\mathcal{A}^{\mathcal{O}(\cdot)}(\mathsf{pp})$:*

$$\Pr\left[\mathcal{A}^{F_{\mathsf{sk}}(\cdot)}(\mathsf{pp}) = x^* \ \wedge \ F_{\mathsf{sk}}(x^*) = \mathsf{accept}\right] = \mathrm{negl}(\lambda).$$

over $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$.

We optionally require the following stronger notion of classical hardness:

3 *((Classical) Pseudorandomness of outputs and public parameters) For all PPT distinguishers $\mathcal{A}$:*

$$\left|\Pr\left[\mathcal{A}^{F_{\mathsf{sk}}(\cdot)}(\mathsf{pp}) = 1\right] - \Pr\left[\mathcal{A}^{R}(\widetilde{\mathsf{pp}}) = 1\right]\right| \leq \mathrm{negl}(\lambda).$$

over $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$, and where R is a uniformly random function, and $\widetilde{\mathsf{pp}}$ is uniformly random.

Theorem 5.2. *Let Π be a 4-message interactive proof of quantumness satisfying the properties specified in Lemma 4.9: (Item 1) the second verifier message is public-coin and (Item 2) verifier messages are pseudorandom (Definition 4.8). Then additionally assuming one-way functions, there exists a quantum advantage function with pseudorandom outputs satisfying 2-quantum easiness (Definition 5.1).*

Combined with Lemma 4.9, we obtain the following:

Corollary 5.3. *Assuming the (classical) hardness of LWE, there exists a quantum advantage function with pseudorandom outputs satisfying 2-quantum easiness (Definition 5.1).*

Construction. Let Π be a 4-message interactive proof of quantumness. Let $(\mathsf{PRF.KeyGen}, \mathsf{PRF})$ be a one-wise independent PRF (Definition A.3).

We define our quantum advantage function $(\mathsf{Setup}, F_{\mathsf{sk}})$ as follows:

- $\mathsf{Setup}(1^\lambda)$: Sample $K \leftarrow \mathsf{PRF.KeyGen}(1^\lambda)$. Compute a first verifier message v_1 for Π, using some fresh randomness ρ. Set $\mathsf{pp} = v_1$, $\mathsf{sk} = (\mathsf{pp}, K, \rho)$, and output $(\mathsf{pp}, \mathsf{sk})$.
- F_{sk} : on input x, we consider two distinguished cases:[14]
  - If x is of the form p_1: Compute the public-coin verifier message $v_2 = \mathsf{PRF}_K(p_1)$, which we interpret as a second verifier message with partial transcript (v_1, p_1) (where $v_1 = \mathsf{pp}$). Output $y = v_2$.
  - If x is of the form (p_1, p_2): Compute $v_2 = \mathsf{PRF}_K(p_1)$. If the verifier for Π accepts the transcript (v_1, p_1, v_2, p_2) with secret state ρ, output accept, otherwise output reject.
  - Otherwise output reject.

Lemma 5.4 (Quantum easiness)**.** *Suppose Π satisfies quantum completeness (Definition 4.8), and $(\mathsf{PRF.KeyGen}, \mathsf{PRF})$ is one-wise independent (Definition A.3). Then $(\mathsf{Setup}, F_{\mathsf{sk}})$ satisfies quantum easiness.*

[14] Technically, to have F_{sk} be defined over a fixed input domain, we actually distinguish the cases $x = (0\|p_1\|*)$ and $x = (1\|p_1, p_2)$ where $*$ denotes a 0 padding of appropriate length, and where F_{sk} outputs reject on inputs not of this form. We keep the notation of the construction above for clarity of exposition.

Proof. Let $\mathcal{P}$ denote the efficient quantum prover for Π such that

$$\Pr\left[(\mathcal{P},\mathcal{V})(1^\lambda)=1\right]\geq 1-\mathrm{negl}(\lambda).$$

Define the following QPT algorithm $\mathcal{A}(\mathsf{pp})$:

- On input pp, parse $\mathsf{pp}=v_1$ as a first verifier message in Π, and compute a first prover message p_1 according to $\mathcal{P}$. Query F_{sk} on input p_1, and receive v_2.
- Given (v_1,p_1,v_2), compute the second prover message p_2 according to $\mathcal{P}$. Output $x^*=(p_1,p_2)$.

By construction, (v_1,p_1,v_2,p_2) denotes a transcript generated by $\mathcal{P},\mathcal{V}$, where $\mathcal{V}$ uses randomness ρ and $\rho_2=\mathsf{PRF}_K(p_1)$ to generate its messages v_1 and v_2 respectively. Since PRF is one-wise independent, $\mathcal{A}$ perfectly simulates the view of $\mathcal{P}$ in an interaction with $\mathcal{V}$. Thus $F_{\mathsf{sk}}(x^*)$ outputs accept with probability $1-\mathrm{negl}(\lambda)$. □

Lemma 5.5 (Classical hardness)**.** *Suppose Π is sound against classical provers and has public-coin intermediate verifier messages (Definition 4.8, Property 1) and that $(\mathsf{PRF.KeyGen},\mathsf{PRF})$ is a (classically secure) PRF. Then $(\mathsf{Setup},F_{\mathsf{sk}})$ satisfies classical hardness.*

Proof. Let $\mathcal{A}(\mathsf{pp})$ denote a PPT adversary with oracle access to F_{sk}. Without loss of generality, we assume that $\mathcal{A}$ queries its output x^* to F_{sk} before halting, and that $\mathcal{A}$ outputs the first x^* it queries such that $F_{\mathsf{sk}}(x^*)=\mathsf{accept}$, if such a query exists. Let Q denote the number of oracle queries $\mathcal{A}$ makes. We define a sequence of hybrid experiments, where we change the input-output behaviour of F_{sk}, as follows:

- **Hybrid 0**: This is the classical hardness experiment (Definition 5.1, Property 2) where $\mathcal{A}$ has oracle access to $\mathcal{O}^0_{\mathsf{sk}}:=F_{\mathsf{sk}}$, where $(\mathsf{pp},\mathsf{sk})\leftarrow\mathsf{Setup}(1^\lambda)$. We say that the adversary *wins* the experiment if he outputs x^* such that $\mathcal{O}^0_{\mathsf{sk}}(x^*)=\mathsf{accept}$.
- **Hybrid 1**: We change how the oracle queries are handled, and define $\mathcal{O}^1_{\mathsf{sk}}$ as follows. The (now stateful) oracle computes v_2 using a lazily-sampled random function R instead of a PRF. Specifically, on queries of the form $x=p_1$ if $R(x)$ is not yet defined, sample v_2 uniformly and set $R(x)=v_2$, then output v_2.
- **Hybrid 2**: We do not change the behavior of the oracle ($\mathcal{O}^2_{\mathsf{sk}}=\mathcal{O}^1_{\mathsf{sk}}$), but we change the win condition of the experiment. We now guess two uniformly random indices $j_1,j_2\leftarrow[Q]$, where Q denotes the number of oracle queries made by $\mathcal{A}$. We now say that $\mathcal{A}$ wins if and only if the following conditions hold:
  (1) the j_2th oracle query from $\mathcal{A}$, on input x_{j_2}, is of the form $x_{j_2}=(p_1^*,p_2^*)$,
  (2) $\mathcal{O}^2_{\mathsf{sk}}(x_{j_2})=\mathsf{accept}$, and, for all prior oracle queries x, $\mathcal{O}^2_{\mathsf{sk}}(x)\neq\mathsf{accept}$,
  (3) the j_1th oracle query from $\mathcal{A}$, on input x_{j_1} has p_1^* as a prefix (i.e. either $x_{j_1}=p_1^*$ or $x_{j_1}=(p_1^*,p_2)$ for some p_2), and, for all prior oracle queries x, the prefix of x with appropriate length is not equal to p_1^*.
- **Hybrid 3**: We change how oracle queries are handled and define $\mathcal{O}^3_{\mathsf{sk}}$ as follows. On any query $j\neq j_2$ of the form $x_j=(p_1,p_2)$, $\mathcal{O}^3_{\mathsf{sk}}$ rejects.

Claim 5.6. *Suppose $(\mathsf{PRF.KeyGen},\mathsf{PRF}_K)$ is a (classically) secure PRF. Then for all PPT algorithms $\mathcal{A}(\mathsf{pp})$:*

$$\left|\Pr\left[\mathcal{A}(\mathsf{pp})\text{ wins in hybrid 1 }\right]-\Pr\left[\mathcal{A}(\mathsf{pp})\text{ wins in hybrid 0 }\right]\right|\leq\mathrm{negl}(\lambda).$$

Proof. This follows directly by reduction to PRF security. The reduction generates the other parameters ρ and $\mathsf{pp}=v_1$ itself and uses its oracle to respond to queries of the form p_1. It then outputs say 1 whenever $\mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{pp})=x^*\wedge\mathcal{O}_{\mathsf{sk}}(x^*)=\mathsf{accept}$, which it can test efficiently given ρ. □

Claim 5.7. *For all PPT algorithms* $\mathcal{A}(\mathsf{pp})$*:*

$$\Pr\left[\mathcal{A}(\mathsf{pp}) \text{ wins in hybrid } 2\right] = \frac{1}{Q^2}\Pr\left[\mathcal{A}(\mathsf{pp}) \text{ wins in hybrid } 1\right].$$

Proof. For any execution of the experiment in hybrid 1 such that $\mathcal{A}(\mathsf{pp})$ outputs $x^* = (p_1^*, p_2^*)$ such that $\mathcal{O}_{\mathsf{sk}}^2(x^*) = \mathsf{accept}$, recall that we assume without loss of generality that $\mathcal{A}$ queries the oracle on x^*, and that x^* corresponds to the first query from $\mathcal{A}$ such that $\mathcal{O}_{\mathsf{sk}}^2(x^*) = \mathsf{accept}$. Let j_2^* denote the index corresponding to the first query $\mathcal{A}$ makes on x^*, and let $j_1^* \leq j_2^*$ denote the index corresponding to the first query $\mathcal{A}$ makes that has p_1^* as a prefix (that is, either $x_{j_1} = p_1^*$ or $x_{j_1} = (p_1^*, p_2)$ for some p_2). Over the sole randomness of $j_1, j_2 \leftarrow [Q]$, the probability of $\mathcal{A}$ winning in such an execution in hybrid 2 is therefore the probability that $(j_1, j_2) = (j_1^*, j_2^*)$, which is $1/Q^2$, and the conclusion follows. □

Claim 5.8. *For all PPT algorithms* $\mathcal{A}(\mathsf{pp})$*:*

$$\Pr\left[\mathcal{A}(\mathsf{pp}) \text{ wins in hybrid } 3\right] = \Pr\left[\mathcal{A}(\mathsf{pp}) \text{ wins in hybrid } 2\right].$$

Proof. By definition of the winning condition in hybrid 2, $\mathcal{A}$ loses in the experiment if its jth oracle query on input x_j satisfies $j < j_2$ and $\mathcal{O}_{\mathsf{sk}}(x_j) = \mathsf{accept}$. Furthermore, the queries made by $\mathcal{A}$ after querying its first accepting input x^*, if such an x^* exists, don't affect its output (as we assume $\mathcal{A}$ would then output x^*). Therefore the outputs of the oracles in hybrid 2 and hybrid 3 differ only in executions where $\mathcal{A}$ loses in both hybrid experiments, or for queries that do not affect the output of $\mathcal{A}$. □

Claim 5.9. *Assume* Π *is sound against classical provers, and has public-coin second verifier messages (Definition 4.8, Property 1). Then for all PPT algorithms* $\mathcal{A}(\mathsf{pp})$*:*

$$\Pr\left[\mathcal{A}(\mathsf{pp}) \text{ wins in hybrid } 3\right] \leq \mathrm{negl}(\lambda).$$

Proof. Let $\mathcal{A}$ be a PPT algorithm such that $\mathcal{A}$ wins with probability ε in hybrid 3. We build a prover $\mathcal{P}^*$ that breaks classical soundness of Π with probability ε as follows:

1. Let R be an initially empty lazily-sampled random function. Upon receiving the first verifier message v_1, set $\mathsf{pp} = v_1$.
2. Make two guesses $j_1, j_2 \leftarrow [Q]$. If $j_1 > j_2$, abort. Run $\mathcal{A}(\mathsf{pp})$.
3. Upon receiving the jth query x_j from $\mathcal{A}$ where $j \in [Q]$, let p_1 denote the prefix of x_j, and compute $R(p_1)$ as follows:
   - If $j = j_1$, set $p_1^* = p_1$ and send p_1^* as the first prover message in Π. Upon receiving the verifier's response v_2^*, set $R(p_1^*) = v_2^*$.
   - Otherwise, if $R(p_1)$ is undefined, sample v_2 uniformly and set $R(p_1) = v_2$.

   To answer the query, if x_j is of the form p_1, respond to the query from $\mathcal{A}$ with $R(p_1)$. Otherwise parse $x_j = (p_1, p_2)$ and respond to the query according to the following cases:
   - If $j = j_2$, check that $p_1 = p_1^*$, set $p_2^* = p_2$ and send p_2^* as the second prover message in Π. (If query x_{j_2} is not of this form $\mathcal{A}$ loses in hybrid 4 and our reduction fails.)
   - Otherwise, respond to the query from $\mathcal{A}$ with reject.

First, our reduction perfectly simulates the view of $\mathcal{A}(\mathsf{pp})$ in hybrid 3, given that second verifier messages in Π are uniformly distributed. Furthermore, if $\mathcal{A}$ wins in hybrid 3, then by definition the prefix p_1^* of the input query x_{j_1} defines v_2^*, and thus $x_{j_2} = (p_1^*, p_2^*)$, which gives an accepting transcript $(v_1, p_1^*, v_2^*, p_2^*)$ for Π between $\mathcal{P}^*$ and $\mathcal{V}$. □

Overall these claims show that the probability of $\mathcal{A}$ winning in hybrid 0 is negligible, and finishes the proof of Lemma 5.5. □

Last, we show that we can obtain pseudorandomness of F_{sk} with a simple modification.

Lemma 5.10 (Pseudorandomness)**.** *Under the same hypotheses as Lemmas 5.4 and 5.5 there exists a quantum advantage function $\widetilde{F}_{\mathsf{sk}}$ satisfying pseudorandomness.*

Proof. Let $(\mathsf{Setup}, F_{\mathsf{sk}})$ denote the previous construction. We define $\widetilde{F}_{\mathsf{sk}}$ as follows: on input x, compute $F_{\mathsf{sk}}(x)$. If $F_{\mathsf{sk}}(x) = \mathsf{reject}$, output $\mathsf{PRF}_K(x)$; otherwise output $F_{\mathsf{sk}}(x)$. Pseudorandomness of non-special outputs of F_{sk} (that is, accept or reject) follows by the public-coin property of second verifier messages of Π (Definition 4.8, Property 1). Furthermore, it is classically hard to find inputs x such that $F_{\mathsf{sk}}(x) = \mathsf{accept}$ by classical hardness of F_{sk}, and inputs x such that $F_{\mathsf{sk}}(x) = \mathsf{reject}$ are mapped by $\widetilde{F}_{\mathsf{sk}}$ to pseudorandom outputs by PRF security. The proofs of quantum easiness and classical hardness for $\widetilde{F}_{\mathsf{sk}}$ follow almost identically to the ones for F_{sk}. □

Remark 5.11 (Generalizing to constant-round proofs of quantumness). Our definitions, construction and proofs can readily be extended to work starting with any constant-round interactive proof of quantumness, assuming all intermediate verifier messages are public-coin (that is, not counting the first verifier message if the verifier produces the first message of the protocol). Starting with a $2k$-message protocol, this gives a quantum advantage function with $(k-1)$-quantum easiness (and where classical hardness and pseudorandomness hold as in Definition 5.1).

Removing public parameters. We observe that any quantum advantage function with public parameters induces one without public parameters. Let $(\overline{\mathsf{Setup}}, \overline{F}_{\overline{\mathsf{sk}}})$ be a quantum advantage function. Consider the following algorithms $(\mathsf{Setup}, F_{\mathsf{sk}})$:

- $\mathsf{Setup}(1^\lambda)$: run $(\overline{\mathsf{pp}}, \overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{Setup}}(1^\lambda)$ and output $\mathsf{sk} = (\overline{\mathsf{pp}}, \overline{\mathsf{sk}})$.
- F_{sk}: on input x, if $x = \mathsf{init}$ where init is a special input symbol, output $\overline{\mathsf{pp}}$. Otherwise output $\overline{F}_{\overline{\mathsf{sk}}}(x)$.[15]

Claim 5.12. *Assume that $(\overline{\mathsf{Setup}}, \overline{F}_{\overline{\mathsf{sk}}})$ is a quantum advantage function. Then $(\mathsf{Setup}, F_{\mathsf{sk}})$ satisfies 2-quantum easiness, and classical hardness (Definition 5.1). Furthermore, assuming that $(\overline{\mathsf{Setup}}, \overline{F}_{\overline{\mathsf{sk}}})$ has pseudorandom outputs and public parameters (Definition 5.1), then $(\mathsf{Setup}, F_{\mathsf{sk}})$ also has pseudorandom outputs (against classical distinguishers).*

Corollary 5.13. *Assuming the (classical) hardness of LWE, there exists a quantum advantage function without public parameters, that satisfies 2-quantum easiness, and have pseudorandom outputs (against classical distinguishers).*

Randomized Quantum Advantage Functions. It will also be useful to us in some cases to consider *randomized* quantum advantage functions, for which we can consider the following stronger notion of pseudorandomness:

3' (Strong pseudorandomness of outputs and public parameters) For all PPT distinguishers $\mathcal{A}$:

$$\left|\Pr\left[\mathcal{A}^{F_{\mathsf{sk}}(\cdot)}(\mathsf{pp}) = 1\right] - \Pr\left[\mathcal{A}^{U}(\widetilde{\mathsf{pp}}) = 1\right]\right| \leq \mathrm{negl}(\lambda).$$

over $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$, and where U is defined as sampling and outputting *fresh* independent randomness at every call, and where $\widetilde{\mathsf{pp}}$ is uniformly random.

[15] Technically, we pad the shorter of $\overline{\mathsf{pp}}$ and $\overline{F}_{\overline{\mathsf{sk}}}(x)$ to obtain outputs with fixed length. We define the padding as an independent PRF of the input to conserve pseudorandomness of outputs.

We observe that our previous construction of (deterministic) quantum advantage function can be extended to satisfy the stronger property above, by modifying the construction as follows (the construction of Setup is unchanged):

- F_{sk} : on input x, sample a fresh uniformly random string $r \leftarrow \{0,1\}^\lambda$. We consider two distinguished cases:
  - If x is of the form p_1:
    - Compute the public-coin verifier message $v_2 = \mathsf{PRF}_K(p_1 \| r)$, which we interpret as a second verifier message with partial transcript (v_1, p_1) (where $v_1 = \mathsf{pp}$). and output $y = (v_2, r)$.
  - If x is of the form $(p_1, \overline{r}, p_2)$ for some $\overline{r} \in \{0,1\}^\lambda$:
    - Compute $v_2 = \mathsf{PRF}_K(p_1 \| \overline{r})$. If the verifier for Π accepts the transcript (v_1, p_1, v_2, p_2) with secret state ρ, output accept, otherwise output $\mathsf{PRF}_K(x \| r)$.
  - Otherwise output $\mathsf{PRF}_K(x \| r)$.

Quantum completeness follows by quantum completeness for Π and one-wise independence of PRF. To see why pseudorandomness holds, consider, for any 4-message interactive proof of quantumness Π, the following modified protocol $\widetilde{\Pi}$. Prover messages in $\widetilde{\Pi}$ have the form $\tilde{p}_i = (p_i, r_i)$, where p_i corresponds to original prover messages in Π. The new verifier algorithm simply ignores the r_i part of $\tilde{p}_i$, and computes its messages as the verifier in Π (with prover messages p_i). It is immediate to see that if Π has public-coin intermediate verifier messages and (classically) pseudorandom verifier messages (Definition 4.8, Properties 1,2), then $\widetilde{\Pi}$ also satisfies these properties. By our previous proof, the public parameters pp along with outputs of F_{sk}, are classically indistinguishable from a uniform $\widetilde{\mathsf{pp}}$ along with outputs of a random function R, which, on input x, outputs $(R(x\|r), r)$ for a freshly sampled $r \leftarrow \{0,1\}^\lambda$. This in turn is statistically indistinguishable from truly uniform outputs and public parameters, and strong pseudorandomness follows.

5.2 Quantum Disclosure of Secrets

Definition 5.14 (Quantum Disclosure of Secrets)**.** *Let Π_{QDS} denote an interactive protocol between a sender and receiver. The sender $\mathcal{S}$ has as input a message m, while the receiver $\mathcal{R}$ has no input.*

We say that Π_{QDS} is a quantum disclosure of secrets *if there is the following quantum-classical gap:*

1. *(Quantum correctness) There is an efficient quantum receiver $\mathcal{R}^*$ such that, if $\mathcal{R}^*$ interacts with the honest sender $\mathcal{S}$, $\mathcal{R}^*$ outputs the sender's message m with probability $1 - \mathrm{negl}(\lambda)$.*

2. *(Classical privacy) For any efficient classical receiver $\mathcal{R}$, if $\mathcal{R}$ interacts with the honest sender S, for any pair of messages m_0, m_1, the view of $\mathcal{R}$ when interacting with $\mathcal{S}(m_0)$ is computationally indistinguishable from the view of $\mathcal{R}$ when interacting with $\mathcal{S}(m_1)$.*

Theorem 5.15. *Let Π be a 4-message interactive proof of quantumness with unique final answer (Definition 4.8, Property 3). Then there exists a 3-message quantum disclosure of secrets protocol. Furthermore, if Π has pseudorandom verifier messages (Definition 4.8, Property 2), then the sender messages in Π_{QDS} are jointly classically indistinguishable from uniformly random.*

Combined with Lemma 4.10, we obtain the following:

Corollary 5.16. *Assuming the classical hardness of LWE, there exists a 3-message quantum disclosure of secrets protocol, such that sender messages are jointly classically indistinguishable from uniformly random.*

Construction. We focus on one-bit messages. Extending it to arbitrary length messages is then done by executing independent copies of the protocol in parallel for each bit of the message; security follows by a hybrid argument.

Let Π be a 4-round interactive proof of quantumness with unique final answer (Lemma 4.10). We define our 3-message quantum disclosure of secrets protocol Π_{QDS} as follows:

- The sender $\mathcal{S}$ generates a first verifier message v_1 for the interactive proof of quantumness and internal state st. The sender sends a first message $s_1 = v_1$ to the receiver.
- The receiver $\mathcal{R}$ responds with a prover message $r_1 = p_1$ for the interactive proof of quantumness.
- The sender $\mathcal{S}$ computes a third message v_2 for the interactive proof of quantumness as well as $p_2^* = \mathsf{UniqueAnswer}(v_1, p_1, v_2, \mathsf{st})$. The sender sends its second message $s_2 = (v_2, r, y = \langle r, p_2^* \rangle \oplus m)$ for uniformly random $r \leftarrow \{0,1\}^\ell$ where $\ell = |p_2^*|$.

Lemma 5.17 (Quantum correctness)**.** *Suppose Π is a 4-message interactive proof of quantumness with unique final answer (Definition 4.8). Then Π_{QDS} satisfies quantum correctness.*

Proof. By completeness of the interactive proof of quantumness and its unique final answer property, if the receiver $\mathcal{R}^*$ emulates the quantum prover in the interactive proof of quantumness, $\mathcal{R}^*$ can compute p_2^* with probability $1 - \mathrm{negl}(\lambda)$, and therefore recover m from s_2. □

Lemma 5.18 (Classical privacy)**.** *Suppose Π is a 4-round interactive proof of quantumness with unique final answer (Definition 4.8). Then Π_{QDS} satisfies classical privacy.*

Proof. We begin by defining a hybrid experiment:

- **Hybrid 1:** We modify the behavior of the sender as follows. In computing the second sender message s_2, sample y uniformly at random and send $s_2 = (v_2, r, y)$.

It suffices to show that the view of the receiver in hybrid 1 is indistinguishable from its view in an interaction with $S(m)$ for any $m \in \{0,1\}$. Suppose there is a message $m \in \{0,1\}$ and an algorithm $\mathcal{D}$ along with a (classical) receiver $\mathcal{R}$ that distinguishes an interaction with $S(m)$ from the one in hybrid 1 with non-negligible probability $\varepsilon(\lambda)$, namely:

$$\left| \Pr_{\tau,r}\left[\mathcal{D}(\tau, r, \langle r, p_2^* \rangle \oplus m) = 1\right] - \Pr_{\tau,r,y}\left[\mathcal{D}(\tau, r, y) = 1\right] \right| \geq \varepsilon(\lambda),$$

where the probability is over the internal randomness of $\mathcal{D}$, $(\tau, r, \langle r, p_2^* \rangle \oplus m)$ where $\tau = (v_1, p_1, v_2)$ is distributed according to an interaction between $S(m)$ and $\mathcal{R}$, and $y \leftarrow \{0,1\}$ is uniformly random.

Such a distinguisher immediately gives an algorithm $\mathsf{Pred}_m(\tau, r)$ that, with m hard-coded and on input $\tau = (v_1, p_1, v_2)$ and uniform r satisfies

$$\Pr_{\tau,r}[\mathsf{Pred}_m(\tau, r) = \langle r, p_2^* \rangle] \geq \frac{1}{2} + \frac{\varepsilon}{2},$$

where the probability is over the internal randomness of Pred, $r \leftarrow \{0,1\}^\ell$ (where $\ell = |p_2^*|$) and τ which is distributed according to an interaction between $S(m)$ and $\mathcal{R}$. Define the set GOOD consisting of τ such that,

$$\Pr_{r}[\mathsf{Pred}_m(\tau, r) = \langle r, p_2^* \rangle] \geq \frac{1}{2} + \frac{\varepsilon}{4},$$

where the probability is only over the internal randomness of Pred and $r \leftarrow \{0,1\}^\ell$. By a standard averaging argument $\Pr_\tau[\tau \in \mathsf{GOOD}] \geq \varepsilon/4$. Since $\varepsilon/4$ is non-negligible, let $c > 0$ be a constant such that $\varepsilon/4 > 1/\lambda^c$ for infinitely many λ. For a *fixed* $\tau \in \mathsf{GOOD}$, applying Lemma 4.1 to $\mathsf{Pred}_m(\tau, \cdot)$ yields a polynomial-time oracle algorithm $\mathsf{Decode}^{\mathcal{O}}$ satisfying

$$\Pr[\mathsf{Decode}^{\mathsf{Pred}_m(\tau,\cdot)}(1^{\lambda^c}, 1^\lambda) = p_2^*] \geq \mathrm{poly}(\varepsilon/4, 1/\lambda),$$

which is non-negligible in λ. Finally we define a classical cheating prover $\mathcal{P}^*$ for Π as follows:

1. Upon receiving the first verifier message v_1, run an interaction with $\mathcal{R}$ sending $s_1 = v_1$ as the first sender message.

2. When $\mathcal{R}$ sends a receiver message r_1, send $p_1 = r_1$ to the verifier.

3. Upon receiving the second verifier message v_2, let $\tau = (v_1, p_1, v_2)$, construct $\mathsf{Pred}_m(\tau, \cdot)$ and compute $p_2 \leftarrow \mathsf{Decode}^{\mathsf{Pred}_m(\tau,\cdot)}(1^{\lambda^c}, 1^\lambda)$. Send p_2 to the verifier as the second prover message.

Conditioned on $\tau \in \mathsf{GOOD}$ and the decoder succeeding, by construction of the sender in Π_{QDS}, $p_2 = \mathsf{UniqueAnswer}(v_1, p_1, v_2, \mathsf{st}) = p_2^*$ where st is the verifier's internal state. By the above arguments, the events that $\tau \in \mathsf{GOOD}$ and the decoder succeeds simultaneously occur with non-negligible probability. □

Lemma 5.19 (Pseudorandomness of verifier messages)**.** *Suppose that* Π *has pseudorandom verifier messages (Definition 4.8, Property 2). Then the sender messages in* Π_{QDS} *are jointly classically indistinguishable from uniformly random.*

Proof. The proof of Lemma 5.18 shows that the second sender message is computationally indistinguishable from (v_2, z, y) where $z \leftarrow \{0,1\}^\ell$ and $y \leftarrow \{0,1\}$. Pseudorandomness of sender messages then follows by pseudorandomness of verifier messages in Π. □

Quantum Disclosure of Secrets Function. Let Π_{QDS} be a quantum disclosure of secrets. We define, for all messages m, the following *quantum disclosure of secrets function* $(\mathsf{Setup}, F_{\mathsf{sk},m})$:

- $\mathsf{Setup}(1^\lambda)$:[16] Sample the first sender message s_1 in Π_{QDS}, along with an internal state st and some (potentially correlated) randomness for the second sender message ρ_2, and output $(\mathsf{pp} = s_1, \mathsf{sk} = (s_1, \mathsf{st}, \rho_2))$.
- $F_{\mathsf{sk},m}$: On input x, parse x as a receiver message r_1 in Π_{QDS}, and compute a second sender message s_2 given $(s_1, r_1, \mathsf{st}, m)$ using randomness ρ_2.

We note that $F_{\mathsf{sk},m}$ is stateless and deterministic. The properties of Π_{QDS} translate directly to properties of $(\mathsf{Setup}, F_{\mathsf{sk},m})$:

- Quantum easiness: there exists a QPT algorithm $\mathcal{A}$ such that
$$\Pr\left[\mathcal{A}^{F_{\mathsf{sk},m}}(\mathsf{pp}) = m\right] = 1 - \mathrm{negl}(\lambda),$$
where $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$, and where $\mathcal{A}$ makes one classical query to $F_{\mathsf{sk},m}$;
- Weak pseudorandomness: for all PPT algorithms $\mathcal{A}$ *that make at most one oracle query*:
$$\left|\Pr\left[\mathcal{A}^{F_{\mathsf{sk},m}(\cdot)}(\mathsf{pp}) = 1\right] - \Pr\left[\mathcal{A}^{R}(\widetilde{\mathsf{pp}}) = 1\right]\right| \leq \mathrm{negl}(\lambda),$$
where $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$, R denotes a random function and $\widetilde{\mathsf{pp}}$ is uniformly sampled.

Removing Public Parameters from the QDS Function. We observe that any QDS function with public parameters induces a QDS function without public parameters as follows. Let $(\overline{\mathsf{Setup}}, \overline{F}_{\overline{\mathsf{sk}},m})$ be a QDS function, and $\mathcal{H}$ be a family of pairwise independent hash functions with uniformly random description.[17] Consider the following algorithms $(\mathsf{Setup}, F_{\mathsf{sk},m})$:

[16] In general, the first sender message in the QDS s_1 depends on the message m, and so in general Setup would take m as input. For simplicity of notation, we note that our construction of QDS above is delayed-input, in the sense that s_1 is computed independently of m, which allows Setup to be independent of m. Our counterexamples in Section 6 would work even if the QDS was not delayed input.

[17] Uniform description follows by considering for instance random affine functions over the field $\{0,1\}^n$ where n denotes the input size, so that hash functions have descriptions $h = (a, b) \leftarrow \{0,1\}^n \times \{0,1\}^n$.

- $\mathsf{Setup}(1^\lambda)$: Sample $(\overline{\mathsf{pp}}, \overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{Setup}}(1^\lambda)$, and sample a pairwise independent hash function $h \leftarrow \mathcal{H}$. Output $\mathsf{sk} = (\overline{\mathsf{pp}}, \overline{\mathsf{sk}}, h)$.
- $F_{\mathsf{sk},m}$: on input x, if $x = \mathsf{init}$ where init is a special input symbol, output $y = (h, \overline{\mathsf{pp}})$. Otherwise output $y = \overline{F}_{\overline{\mathsf{sk}},m}(x) \oplus h(x)$.

The resulting QDS function $(\mathsf{Setup}, F_{\mathsf{sk},m})$ has the following properties:

- 2-Quantum easiness: there exists a QPT algorithm $\mathcal{A}$ that outputs m using two classical queries to $F_{\mathsf{sk},m}$. This follows by calling $F_{\mathsf{sk},m}$ on input init, receving $(\overline{\mathsf{pp}}, h)$, and then calling the quantum easiness algorithm for $(\overline{\mathsf{Setup}}, \overline{F}_{\overline{\mathsf{sk}},m})$ to (1) obtain an input query x, and (2) recover m from the output from $(\overline{\mathsf{Setup}}, \overline{F}_{\overline{\mathsf{sk}},m})$ (which can be recovered by computing $h(x)$ given h and unmasking the output of $F_{\mathsf{sk},m}$).
- 2-Query weak pseudorandomness: for any PPT algorithm $\mathcal{A}$ making at most 2 oracle queries, $F_{\mathsf{sk},m}$ is computationally indistinguishable from a random function. This follows by considering the following cases. If none of the two queries are made on input $x = \mathsf{init}$, pseudorandomness follows by pairwise independence of h. Otherwise at most one query is made on an input $x \neq \mathsf{init}$, and weak pseudorandomness follows by 1-query weak pseudorandomness of $(\overline{\mathsf{Setup}}, \overline{F}_{\overline{\mathsf{sk}},m})$.

6 Counterexamples for Post-Quantum Security

In this section we use our functions from Section 5 to give examples of classically secure primitives that are quantum insecure.

6.1 Counterexamples for Standard Cryptographic Primitives

We first focus on cryptographic primitives with usual security notions. We refer to Appendix A for formal definitions of the cryptographic primitives we consider. Note that that the precise formulations of the security experiments do influence the exact query complexity in the theorem below.

Theorem 6.1. *Assuming the existence of a quantum advantage function with pseudorandom outputs (Definition 5.1), there exists:*

- *A signature scheme that is secure against classical adversaries, but insecure against quantum adversaries making two classical queries to the signing oracle.*
- *Additionally assuming the existence of CCA-1 (resp. CCA-2)-secure public-key encryption, there exists a CCA-1 (resp. CCA-2)-secure public-key encryption scheme that is secure against classical adversaries, but insecure against quantum adversaries making two classical queries to the decryption oracle before making its challenge query.*[18]
- *A PRF with public parameters that is secure against classical adversaries, but insecure against quantum adversaries making two classical queries to the PRF.*
- *A CPA-secure symmetric-key encryption scheme with public parameters that is secure against classical adversaries, but insecure against quantum adversaries making one query to the encryption oracle before making its challenge query (see Definition A.5).*
- *A MAC with public parameters that is secure against classical adversaries, but insecure against quantum adversaries making one query to the authentication oracle.*

[18] In other words, the quantum attack is a CCA-1 attack.

Furthermore there exists a PRF, MAC and CPA-secure symmetric encryption scheme each without public parameters and with the same classical security, but insecurity against quantum adversaries making one additional query to the respective oracles than listed above.

Combined with Corollary 5.3, such constructions exist assuming the (classical) hardness of LWE.

Counterexample for Signatures. Let $(\mathsf{Setup}, F_{\mathsf{sk}})$ be a quantum advantage function (Definition 5.1). Let $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$ be a (classically) secure signature scheme. We define the following signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$:

- $\mathsf{KeyGen}(1^\lambda)$: Sample $(\overline{\mathsf{Sig}}.\overline{\mathsf{vk}}, \overline{\mathsf{Sig}}.\overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Output $(\mathsf{Sig.vk} = (\overline{\mathsf{Sig}}.\overline{\mathsf{vk}}, \mathsf{pp}), \mathsf{Sig.sk} = (\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}, \mathsf{sk}))$.
- $\mathsf{Sign}(\mathsf{Sig.sk}, m)$: Compute $\overline{\sigma} \leftarrow \overline{\mathsf{Sign}}(\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}, m)$ and $y = F_{\mathsf{sk}}(m)$. If $y = \mathsf{accept}$, output $\sigma = (\overline{\sigma}, \overline{\mathsf{Sig}}.\overline{\mathsf{sk}})$. Otherwise, output $\sigma = (\overline{\sigma}, y)$.
- $\mathsf{Verify}(\overline{\mathsf{Sig}}.\mathsf{vk}, m, \sigma)$: Output $\overline{\mathsf{Verify}}(\overline{\mathsf{Sig}}.\overline{\mathsf{vk}}, m, \overline{\sigma})$.

Correctness of $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ follows directly from correctness of $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$.

Claim 6.2. *Suppose that* $(\mathsf{Setup}, F_{\mathsf{sk}})$ *satisfies quantum easiness (Definition 5.1), and that* $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$ *is correct. Then there exists a QPT adversary* $\mathcal{F}$ *that breaks unforgeability of* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *using two (classical) signing queries.*

Proof. Let $\mathcal{A}$ be the QPT algorithm associated to the quantum easiness of $(\mathsf{Setup}, F_{\mathsf{sk}})$ (Definition 5.1). Define $\mathcal{F}$ as follows. Run $\mathcal{A}$ to obtain $x_1 \leftarrow \mathcal{A}(\mathsf{pp})$, and send a signing query with message x_1. Upon receiving $\sigma_1 = (\overline{\sigma}_1, y_1)$, continue the execution of $\mathcal{A}$, setting the oracle response as y_1, and submit x_2 as the second query. $\mathcal{F}$ receives as response σ_2 which it parses as $\sigma_2 = (\overline{\sigma}_2, y_2)$. It picks an arbitrary $m \neq q_1, q_2$ and outputs as its forgery $\sigma^* = \overline{\mathsf{Sign}}(y_2, m)$.

By quantum easiness of $(\mathsf{Setup}, F_{\mathsf{sk}})$, we have with overwhelming probability $y_2 = \overline{\mathsf{Sig}}.\overline{\mathsf{sk}}$. Thus $\mathcal{F}$ produces a valid forgery with overwhelming probability by correctness of $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$. □

Claim 6.3. *Suppose* $(\mathsf{Setup}, F_{\mathsf{sk}})$ *satisfies classical hardness (Definition 5.1), and that* $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$ *is unforgeable (against classical adversaries). Then* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is unforgeable against classical adversaries.*

Proof. We define the following hybrid experiment:

- **Hybrid 1:** We modify the behavior of the signing oracle. Compute $\overline{\sigma} \leftarrow \overline{\mathsf{Sign}}(\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}, m)$ and $y = F_{\mathsf{sk}}(m)$ as normal. If $y = \mathsf{accept}$, abort. Otherwise, output $\sigma = (\overline{\sigma}, y)$.

For any PPT adversary $\mathcal{F}$, the probability of $\mathcal{F}$ making a signing query with some input m that makes the signing oracle abort in hybrid 1 is negligible by classical hardness of $(\mathsf{Setup}, F_{\mathsf{sk}})$ (Theorem 5.2). Therefore the output of the unforgeability experiment for $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is indistinguishable from its output in hybrid 1.

Now unforgeability in hybrid 1 follows directly from (classical) unforgeability of $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$, where the reduction samples $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and computes $y = F_{\mathsf{sk}}(m)$ on its own upon receiving a signing query with message m. □

The counterexamples for CCA-secure encryption, PRFs, symmetric-key encryption and MACS, along with the claimed classical security and quantum insecurity, follow in an almost identical manner. We present the constructions below.

Counterexample for CCA-Secure Public-Key Encryption. Let $(\mathsf{Setup}, F_{\mathsf{sk}})$ be a quantum advantage function (Definition 5.1). Let $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ be a (classically) secure CCA-1-secure (resp. CCA-2-secure) encryption scheme. We define the following encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$:

- $\mathsf{KeyGen}(1^\lambda)$: Sample $(\overline{\mathsf{Enc}}.\overline{\mathsf{pk}}, \overline{\mathsf{Enc}}.\overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Output $(\mathsf{Enc.pk} = (\overline{\mathsf{pk}}, \mathsf{pp}), \mathsf{Enc.sk} = (\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \mathsf{sk}))$.
- $\mathsf{Enc}(\mathsf{Enc.pk}, m)$: Compute $\overline{\mathsf{ct}} = \overline{\mathsf{Enc}}(\overline{\mathsf{Enc}}.\overline{\mathsf{pk}}, m)$, and output $(0\|\overline{\mathsf{ct}})$.
- $\mathsf{Dec}(\mathsf{Enc.sk}, \mathsf{ct})$: Parse $\mathsf{ct} = (b\|\overline{\mathsf{ct}})$ where $b \in \{0,1\}$. If $b = 0$, output $m = \overline{\mathsf{Dec}}(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \overline{\mathsf{ct}})$. If $b = 1$ and $F_{\mathsf{sk}}(\overline{\mathsf{ct}}) = \mathsf{accept}$, output $m = \mathsf{Enc}.\overline{\mathsf{sk}}$. Otherwise output $m = F_{\mathsf{sk}}(\overline{\mathsf{ct}})$.

Quantum insecurity follows from being able to find an input x such that $F_{\mathsf{sk}}(x) = \mathsf{accept}$ and therefore obtain $\mathsf{Enc}.\overline{\mathsf{sk}}$, making only two queries to the decryption oracle. This gives a quantum attack on the CCA-1 security of $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Classical CCA-1 security (resp. CCA-2 security) follows from the CCA-1 security (resp. CCA-2 security) of $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ along with the classical hardness of $(\mathsf{Setup}, F_{\mathsf{sk}})$, using an identical argument as above.

Counterexample for Pseudorandom Functions. Let $(\mathsf{Setup}, F_{\mathsf{sk}})$ be a quantum advantage function with pseudorandom outputs (Definition 5.1). The pseudorandomness property of $(\mathsf{Setup}, F_{\mathsf{sk}})$ immediately implies that $(\mathsf{KeyGen} = \mathsf{Setup}, \mathsf{PRF}_{\mathsf{sk}} = F_{\mathsf{sk}})$ is a classically secure PRF with public parameters.[19] Additionally, by quantum easiness of $(\mathsf{Setup}, F_{\mathsf{sk}})$, a QPT adversary can find, in one query, an input x such that $F_{\mathsf{sk}}(x) = \mathsf{accept}$ and, thus making a second query on that input x allows the adversary to distinguish.

Counterexample for Symmetric-Key Encryption. Let $(\mathsf{Setup}, F_{\mathsf{sk}})$ be a *randomized* quantum advantage function with strong pseudorandom outputs and public parameters (see Section 5.1) . Let $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ be a (classically) secure symmetric-key encryption scheme. We define the following encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with public parameters (see Footnote 19):

- $\mathsf{KeyGen}(1^\lambda)$: Sample $(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Output $(\mathsf{Enc.pp} = \mathsf{pp}, \mathsf{Enc.sk} = (\mathsf{pp}, \overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \mathsf{sk}))$.
- $\mathsf{Enc}(\mathsf{Enc.sk}, m)$: Parse $m = (b, m')$ where $b \in \{0,1\}$. Compute $y \leftarrow F_{\mathsf{sk}}(m')$ and $\overline{\mathsf{ct}} \leftarrow \overline{\mathsf{Enc}}(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, m)$. If $y = \mathsf{accept}$, output $\mathsf{ct} = (\overline{\mathsf{ct}}, \mathsf{Enc.sk})$, and otherwise output $\mathsf{ct} = (\overline{\mathsf{ct}}, y)$.
- $\mathsf{Dec}(\mathsf{Enc.sk}, \mathsf{ct})$: Parse $\mathsf{ct} = (\overline{\mathsf{ct}}, y)$ and output $m = \overline{\mathsf{Dec}}(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \mathsf{ct})$.

Quantum insecurity follows from being able to makes two adaptive classical queries x_1, x_2 to F_{sk} such that $F_{\mathsf{sk}}(x_2) = \mathsf{accept}$. The quantum adversary queries the encryption oracle on message x_1, obtains a response, and makes a challenge query on input $m_0 = (0, x_2)$ and $m_1 = (1, x_2)$ where x_2 denotes the second query to F_{sk}, and obtains $(\overline{\mathsf{Enc}}(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, m_b), \mathsf{Enc.sk})$, from which b can be recovered. Classical security follows from strong pseudorandomness of $(\mathsf{Setup}, F_{\mathsf{sk}})$, which allows the reduction to switch y to fresh uniform values (independent of m'), along with the security of $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$.

Counterexample for MACs. Let $(\mathsf{Setup}, F_{\mathsf{sk}})$ be a quantum advantage function (Definition 5.1). Let $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{MAC}}, \overline{\mathsf{Verify}})$ be a (classically) secure MAC. We define the following MAC $(\mathsf{KeyGen}, \mathsf{MAC}, \mathsf{Verify})$ with public parameters:

- $\mathsf{KeyGen}(1^\lambda)$: Sample $\overline{\mathsf{MAC}}.\overline{\mathsf{sk}} \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Output $(\mathsf{MAC.pp} = \mathsf{pp}, \mathsf{MAC.sk} = (\mathsf{pp}, \overline{\mathsf{MAC}}.\overline{\mathsf{sk}}, \mathsf{sk}))$.
- $\mathsf{MAC}(\mathsf{MAC.sk}, m)$: Compute $\overline{\sigma} \leftarrow \overline{\mathsf{MAC}}(\overline{\mathsf{MAC}}.\overline{\mathsf{sk}}, m)$ and $y = F_{\mathsf{sk}}(m)$. Output $\sigma = (\overline{\sigma}, y)$.

[19] We show how to remove the use of public parameters, at the cost of weakening quantum attack to use one more classical query later in the section.

- $\mathsf{Verify}(\mathsf{MAC.sk}, m, \sigma)$: If $F_{\mathsf{sk}}(m) = \mathsf{accept}$, output accept. Otherwise output $\overline{\mathsf{Verify}}(\overline{\mathsf{MAC}}.\overline{\mathsf{sk}}, m, \overline{\sigma})$.

Quantum insecurity follows from being able to find (after making a *single* classical query) an input x such that $F_{\mathsf{sk}}(x) = \mathsf{accept}$. Therefore x together with any arbitrary tag σ constitutes a forgery for $(\mathsf{KeyGen}, \mathsf{MAC}, \mathsf{Verify})$. Classical security of $(\mathsf{KeyGen}, \mathsf{MAC}, \mathsf{Verify})$ follows from security of $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{MAC}}, \overline{\mathsf{Verify}})$ and classical hardness of $(\mathsf{Setup}, F_{\mathsf{sk}})$, using an identical argument to above.

Removing Public Parameters in Secret-Key Primitives. Using a (deterministic) quantum advantage function without public parameters (Claim 5.12 and Corollary 5.13), we obtain a PRF (respectively, a MAC) without public parameters, that is quantum insecure using three classical PRF queries (resp. two MAC queries).

To remove public parameters from the secret-key encryption counterexample, we simply modify the scheme to append the public parameters pp of the randomized quantum advantage function to all ciphertexts (and new ciphertexts therefore have the form $(\mathsf{pp}, \overline{\mathsf{ct}}, y)$, where either $y = F_{\mathsf{sk}}(m')$ for some m' or $y = \mathsf{Enc.sk}$). The new scheme is still quantumly broken using 2 (classical) queries, where the additional query (on a dummy input) is used to obtain pp. Classical security is maintained given that classical security for the original counterexample held given pp.

6.2 Counterexamples for One-time Primitives

We now study one-time counterparts of the primitives considered in the previous section. Using the results from Section 5.2 we obtain constructions of "one-time" analogs of counterexamples in Section 6.1, that are only secure against classical attackers that are allowed to make only a limited number of queries to their respective oracles. However they are broken by quantum attackers that make one fewer query than their counterparts for the constructions from the previous section. We refer again to Appendix A for more formal definitions (again, note that the precise formulations of the security experiments do influence the exact query complexity in the theorem below)

Theorem 6.4. *Assuming the existence of a quantum disclosure of secrets function (see Section 5.2), there exists:*

- *A one-time signature scheme that is secure against classical adversaries making one query to the signing oracle, but insecure against quantum adversaries making one classical query.*
- *Additionally assuming the existence of single-decryption CCA-1 (resp. CCA-2)-secure public-key encryption, there exists a single-decryption CCA-1 (resp. CCA-2)-secure public-key encryption scheme that is secure against classical adversaries making one query to the decryption oracle, but insecure against quantum adversaries making one classical query.*
- *A one-query PRF with public parameters that is secure against classical adversaries making one query to the PRF, but insecure against quantum adversaries making one classical query. Furthermore, there exists a PRF (without public parameters) that is secure against classical adversaries making two queries to the PRF but insecure against quantum adversaries making two classical queries.*
- *A one-time symmetric-key encryption scheme with public parameters that is secure against classical adversaries (making one challenge query and no encryption queries), but insecure against quantum adversaries. Furthermore, there exists a symmetric-key encryption scheme (without public parameters) that is secure against classical adversaries making one encryption query and one challenge query but insecure against quantum adversaries making one classical encryption query and one challenge query.*

Combined with Corollary 5.16, such constructions exist assuming the (classical) hardness of LWE.

Counterexample for One-Time Signatures. Let $(\mathsf{Setup}, F_{\mathsf{sk},\cdot})$ be a quantum disclosure of secrets function (see Section 5.2). Let $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$ be a (classically) secure one-time signature scheme (Definition A.1). We define the following one-time signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$:

- $\mathsf{KeyGen}(1^\lambda)$: Sample $(\overline{\mathsf{Sig}}.\overline{\mathsf{vk}}, \overline{\mathsf{Sig}}.\overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Output $(\mathsf{Sig.vk} = (\overline{\mathsf{Sig}}.\overline{\mathsf{vk}}, \mathsf{pp}), \mathsf{Sig.sk} = (\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}, \mathsf{sk}))$.
- $\mathsf{Sign}(\mathsf{Sig.sk}, m)$: Compute $\overline{\sigma} \leftarrow \overline{\mathsf{Sign}}(\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}, m)$ and compute the quantum disclosure of secrets function with message $\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}$: $y = F_{\mathsf{sk},\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}}(m)$. Output $\sigma = (\overline{\sigma}, y)$.
- $\mathsf{Verify}(\mathsf{Sig.vk}, m, \sigma)$: Parse $\sigma = (\overline{\sigma}, y)$. Output $\overline{\mathsf{Verify}}(\overline{\mathsf{Sig}}.\overline{\mathsf{vk}}, m, \overline{\sigma})$

Claim 6.5. *Assume $(\mathsf{Setup}, F_{\mathsf{sk},\cdot})$ satisfies quantum easiness (see Section 5.2), and $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$ is correct. Then there exists a QPT adversary $\mathcal{F}$ that breaks unforgeability of $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ using one (classical) signing query.*

Proof. By the quantum easiness property of $(\mathsf{Setup}, F_{\mathsf{sk},\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}})$, $\mathcal{F}$ can recover $\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}$ with overwhelming probability by making only one (classical) query to the signing oracle. Then $\mathcal{F}$ can produce a forgery by running $\overline{\mathsf{Sign}}(\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}, m)$ for an arbitrary message m (different from the one used in the query). □

Claim 6.6. *Assume $(\mathsf{Setup}, F_{\mathsf{sk},\cdot})$ satisfies weak pseudorandomness (see Section 5.2), and $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$ is one-time unforgeable. Then $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is one-time unforgeable against classical adversaries (Definition A.1).*

Proof. We define the following hybrid experiment:

- **Hybrid 1**: We modify the behavior of the signing oracle. Instead of computing $y = F_{\mathsf{sk},\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}}(m)$, sample y uniformly at random.

Given that forgers in the one-time experiment are only allowed to make a single signing query, the output of the experiment defined by hybrid 1 is indistinguishable from that of the one-time unforgeability experiment for $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$, by weak pseudorandomness of $(\mathsf{Setup}, F_{\mathsf{sk},\overline{\mathsf{Sig}}.\overline{\mathsf{sk}}})$. (One-time) unforgeability in hybrid 1 follows directly from (one-time) unforgeability of $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Sign}}, \overline{\mathsf{Verify}})$. □

The counterexamples for single-decryption CCA-secure public-key encryption, one-query PRFs and one-time secure symmetric-key encryption are constructed in a nearly identical manner to the corresponding ones from Section 6.1, with similar modifications as in the above construction for one-time signatures.

Counterexample for Single-Decryption CCA-secure Public-Key Encryption. Let $(\mathsf{Setup}, F_{\mathsf{sk},\cdot})$ be a quantum disclosure of secrets function (see Section 5.2). Let $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ be a one-decryption (classically) secure CCA-1-secure (resp. CCA-2-secure) encryption scheme (Definition A.2). We define the following encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$:

- $\mathsf{KeyGen}(1^\lambda)$: Sample $(\overline{\mathsf{Enc}}.\overline{\mathsf{pk}}, \overline{\mathsf{Enc}}.\overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Output $(\mathsf{Enc.pk} = (\overline{\mathsf{pk}}, \mathsf{pp}), \mathsf{Enc.sk} = (\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \mathsf{sk}))$.
- $\mathsf{Enc}(\mathsf{Enc.sk}, m)$: Compute $\overline{\mathsf{ct}} = \overline{\mathsf{Enc}}(\overline{\mathsf{Enc}}.\overline{\mathsf{pk}}, m)$, and output $(0 \| \overline{\mathsf{ct}})$.
- $\mathsf{Dec}(\mathsf{Enc.sk}, \mathsf{ct})$: Parse $\mathsf{ct} = (b \| \overline{\mathsf{ct}})$ where $b \in \{0, 1\}$. If $b = 0$, output $m = \overline{\mathsf{Dec}}(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \overline{\mathsf{ct}})$. If $b = 1$, compute the quantum disclosure of secrets function with message $\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}$ and input $\overline{\mathsf{ct}}$: $y = F_{\mathsf{sk},\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}}(\overline{\mathsf{ct}})$, and output y.

Counterexample for One-Query PRFs. Let $(\mathsf{Setup}, F_{\mathsf{sk},.})$ be a quantum disclosure of secrets function (see Section 5.2). The weak pseudorandomness property of $(\mathsf{Setup}, F_{\mathsf{sk},.})$ immediately implies that, $(\mathsf{KeyGen} = \mathsf{Setup}, \mathsf{PRF}_{\mathsf{sk}} = F_{\mathsf{sk},\mathsf{sk}})$ is a classically secure one-query PRF with public parameters.[20] Additionally, by quantum easiness of $(\mathsf{Setup}, F_{\mathsf{sk},\mathsf{sk}})$, a QPT adversary can recover sk, and thus distinguish, by making only one query.

Counterexample for One-Time Symmetric-Key Encryption. Let $(\mathsf{Setup}, F_{\mathsf{sk},.})$ be a quantum disclosure of secrets function (see Section 5.2) . Let $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ be a one-time (classically) secure symmetric-key encryption scheme (Definition A.5). We define the following encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with public parameters (see Footnote 20):

- $\mathsf{KeyGen}(1^\lambda)$: Sample $(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Output $(\mathsf{Enc.pp} = \mathsf{pp}, \mathsf{Enc.sk} = (\mathsf{pp}, \overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \mathsf{sk}))$.
- $\mathsf{Enc}(\mathsf{Enc.sk}, m)$: Parse $m = (b, m')$ where $b \in \{0,1\}$. Compute the (randomized) quantum disclosure of secrets function with message $\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}$: $y = F_{\mathsf{sk},\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}}(m')$ and $\overline{\mathsf{ct}} = \overline{\mathsf{Enc}}(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, m)$. Output $\mathsf{ct} = (\overline{\mathsf{ct}}, y)$.
- $\mathsf{Dec}(\mathsf{Enc.sk}, \mathsf{ct})$: Parse $\mathsf{ct} = (\overline{\mathsf{ct}}, y)$ and output $m = \overline{\mathsf{Dec}}(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \mathsf{ct})$.

The proofs of classical security and quantum insecurity all follow almost identically to the above using the one-query properties of the quantum disclosure of secrets function $(\mathsf{Setup}, F_{\mathsf{sk},m})$.[21]

Removing Public Parameters in Secret-Key Primitives. Using the deterministic QDS function without public parameters $(\mathsf{Setup}, F_{\mathsf{sk},m})$ from Section 5.2, we obtain a two-time PRF without public parameters (namely, that is secure against classical adversaries making at most two PRF queries), but that is insecure against quantum attackers making two classical queries. The proofs follow in an almost identical manner, using the two-query properties of $(\mathsf{Setup}, F_{\mathsf{sk},m})$ instead.

Removing public parameters from the symmetric-key encryption counterexample is slightly trickier, because our counterexample with public parameters was deterministic, and therefore security given two QDS queries (one for the encryption query in the experiment, one for the challenge query) is compromised. As a result, we modify the construction as follows.

- $\mathsf{KeyGen}(1^\lambda)$: Sample $(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}) \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Sample $r^* \leftarrow \{0,1\}^\lambda$. Output $\mathsf{Enc.sk} = (\mathsf{pp}, \overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, \mathsf{sk}, r^*)$.
- $\mathsf{Enc}(\mathsf{Enc.sk}, m)$: Parse $m = (b, r, m')$ where $b \in \{0,1\}$ and $r \in \{0,1\}^\lambda$. Compute $\overline{\mathsf{ct}} = \overline{\mathsf{Enc}}(\overline{\mathsf{Enc}}.\overline{\mathsf{sk}}, m)$. If $r \neq r^*$, sample z uniformly at random and output $(\overline{\mathsf{ct}}, r^*, z)$. Otherwise, compute the quantum disclosure of secrets function with message $\mathsf{Enc.sk}$: $y = F_{\mathsf{sk},\mathsf{Enc.sk}}(m')$. Output $\mathsf{ct} = (\overline{\mathsf{ct}}, r^*, y)$.

Note first that appending r^* to all ciphertexts does not hurt security. Then, classical security for this new scheme given one encryption query and one challenge query now follows by *one-query* security of the QDS function, over the randomness of r^*: with overwhelming probability the encryption query will satisfy $r \neq r^*$, and therefore the resulting ciphertext is independent from F_{sk}. Meanwhile, there exists a QPT attack that breaks the scheme by using one encryption query (to obtain r^*) and one challenge query to recover the secret key from the QDS function.

References

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[20] We also show below how to remove the use of public parameters, at the cost of weakening the quantum attack to use one more classical query later in the section.

[21] Note that in the case of symmetric-key encryption, our counterexample has deterministic encryption, which doesn't contradict one-time security.

[Ara02] PK Aravind. The magic squares and bell's theorem. Technical report, 2002.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th FOCS*, pages 474–483. IEEE Computer Society Press, October 2014.

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

[BBK22] Nir Bitansky, Zvika Brakerski, and Yael Tauman Kalai. Constructive post-quantum reductions. *Cryptology ePrint Archive*, 2022.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.

[BIK+22] Saikrishna Badrinarayanan, Yuval Ishai, Dakshita Khurana, Amit Sahai, and Daniel Wichs. Refuting the dream xor lemma via ideal obfuscation and resettable mpc. *ITC*, 2022. https://eprint.iacr.org/2022/681.

[BIN97] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *38th FOCS*, pages 374–383. IEEE Computer Society Press, October 1997.

[BR94] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 232–249. Springer, Heidelberg, August 1994.

[Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 67–95. Springer, Heidelberg, August 2018.

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 269–279. ACM Press, June 2020.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

[BZ13a] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, Heidelberg, May 2013.

[BZ13b] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013.

[CCY21] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. A black-box approach to post-quantum zero-knowledge in constant rounds. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 315–345, Virtual Event, August 2021. Springer, Heidelberg.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CHSH69] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.

[CHTW04] Richard Cleve, Peter Hoyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 236–249. IEEE, 2004.

[CMSZ21] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: breaking the quantum rewinding barrier. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 49–58. IEEE, 2021.

[DJMW12] Yevgeniy Dodis, Abhishek Jain, Tal Moran, and Daniel Wichs. Counterexamples to hardness amplification beyond negligible. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 476–493. Springer, Heidelberg, March 2012.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534. IEEE Computer Society Press, October 1999.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pages 102–115. IEEE Computer Society Press, October 2003.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

[Hai09] Iftach Haitner. A parallel repetition theorem for any interactive argument. In *50th FOCS*, pages 241–250. IEEE Computer Society Press, October 2009.

[HRW16] Dennis Hofheinz, Vanishree Rao, and Daniel Wichs. Standard security does not imply indistinguishability under selective opening. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 121–145. Springer, Heidelberg, October / November 2016.

[KLVY22] Yael Tauman Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. Cryptology ePrint Archive, Report 2022/400, 2022. https://ia.cr/2022/400.

[KMCVY21] Gregory D Kahanamoku-Meyer, Soonwon Choi, Umesh V Vazirani, and Norman Y Yao. Classically-verifiable quantum advantage from a computational bell test. *arXiv preprint arXiv:2104.00687*, 2021.

[KRW15] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 378–400. Springer, Heidelberg, March 2015.

[LLQ] Jiahui Liu, Qipeng Liu, and Luowen Qian. personal communication.

[LLQ22] Jiahui Liu, Qipeng Liu, and Luowen Qian. Beating classical impossibility of position verification. In *ITCS 2022*, 2022.

[LMS21] Alex Lombardi, Fermi Ma, and Nicholas Spooner. Post-quantum zero knowledge, revisited (or: How to do quantum rewinding undetectably). Cryptology ePrint Archive, Report 2021/1543, 2021. https://eprint.iacr.org/2021/1543.

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

[NIS] Computer Security Division. NIST. Post-quantum cryptography. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

[Raz95] Ran Raz. A parallel repetition theorem. In *27th ACM STOC*, pages 447–456. ACM Press, May / June 1995.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Rot13] Ron Rothblum. On the circular security of bit-encryption. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 579–598. Springer, Heidelberg, March 2013.

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

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, Heidelberg, April 2012.

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

[vdG97] Jeroen van de Graaf. *Towards a formal definition of security for quantum protocols*. PhD thesis, University of Montreal, 1997.

[Wat06] John Watrous. Zero-knowledge against quantum attacks. In Jon M. Kleinberg, editor, *38th ACM STOC*, pages 296–305. ACM Press, May 2006.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017.

[YZ21] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 568–597. Springer, Heidelberg, October 2021.

[YZ22] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. *arXiv preprint arXiv:2204.02063*, 2022.

[Zha12a] Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012.

[Zha12b] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, August 2012.

[ZYF+20] Jiang Zhang, Yu Yu, Dengguo Feng, Shuqin Fan, Zhenfeng Zhang, and Kang Yang. Interactive proofs for quantum black-box computations. *Cryptology ePrint Archive*, 2020.

A Additional Preliminaries: Cryptographic Primitives

We recall here the definitions of cryptographic primitives we use and build throughout the paper.

Definition A.1 ((One-time) Signatures)**.** *A signature scheme* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is a tuple of PPT algorithms with the following syntax:*

- $\mathsf{KeyGen}(1^\lambda)$ *outputs a verification key* vk *along with a signing key* vk*;*
- $\mathsf{Sign}(\mathsf{sk}, m)$ *outputs a signature* σ*;*
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$ *outputs* accept *or* reject*.*

We require two properties of the signature scheme:

- *Correctness: for all messages* m*;*

$$\Pr\left[\mathsf{Verify}(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = \mathsf{accept} \quad \middle| \quad (\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)\right] \geq 1 - \mathrm{negl}(\lambda).$$

- *Unforgeability: for all PPT* $\mathcal{A}$*, consider the following experiment between* $\mathcal{A}$ *and a challenger:*
  1. *The challenger generates* $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and sends* vk *to* $\mathcal{A}$*.*
  2. $\mathcal{A}$ *is allowed to repeatedly and adaptively make signing queries for messages* m *of his choice, and the challenger replies with* $\mathsf{Sign}(\mathsf{sk}, m)$*.*
  3. $\mathcal{A}$ *ouputs a forgery* (m^*, σ^*)*.*

  We say that $\mathcal{A}$ wins *the experiment if (1)* $\mathcal{A}$ *has not queried* m^* *in step 2, and (2)* $\mathsf{Verify}(\mathsf{vk}, m^*, \sigma^*) =$ accept*. We say that* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* unforgeable *if, for all PPT* $\mathcal{A}$*, its probabilty of winning the experiment is negligible.*

Let us define the query complexity *of an attack against the unforgeability experiment as the number of signing queries made in step 2.*

We say that a signature scheme is one-time secure *if, for all PPT* $\mathcal{A}$ with query complexity at most 1*, its probabilty of winning the experiment is negligible.*

Definition A.2 ((Single-decryption) CCA-secure Public-Key Encryption)**.** *A CCA-secure public-key encryption scheme* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a tuple of PPT algorithms with the following syntax:*

- $\mathsf{KeyGen}(1^\lambda)$ *outputs a public key* pk *along with a secret key* sk*;*
- $\mathsf{Enc}(\mathsf{pk}, m)$ *outputs a ciphertext* ct*;*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *outputs a message* m'*.*

We require two properties of the encryption scheme:

- *Correctness: for all messages* m*;*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m \quad \middle| \quad (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)\right] \geq 1 - \mathrm{negl}(\lambda).$$

- *CCA-2-Security: for all PPT* $\mathcal{A}$*, consider the following experiment between* $\mathcal{A}$ *and a challenger:*
  1. *The challenger generates* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and sends* pk *to* $\mathcal{A}$*.*
  2. $\mathcal{A}$ *is allowed to repeatedly and adaptively make decryption queries for ciphertexts* ct *of his choice, and the challenger replies with* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$*.*
  3. $\mathcal{A}$ *sends a* challenge query *with messages* m_0 *and* m_1 *to the challenger. The challenger picks a random bit* $b \leftarrow \{0, 1\}$ *and sends back to the adversary* $\mathsf{ct}_b^* = \mathsf{Enc}(\mathsf{pk}, m_b)$ *to* $\mathcal{A}$*.*

4. *$\mathcal{A}$ is again allowed to repeatedly and adaptively make decryption queries for ciphertexts* ct *of his choice, and the challenger replies with* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.
5. *$\mathcal{A}$ outputs a guess* $b' \in \{0,1\}$.

We say that $\mathcal{A}$ wins *the experiment if (1) $\mathcal{A}$ has not queried* ct_b^* *in step 4, and (2)* $b = b'$. *We say that* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is CCA-2-secure if, for all PPT $\mathcal{A}$, its probabilty of winning the experiment is at most* $1/2 + \mathrm{negl}(\lambda)$.

We say that $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is CCA-1-secure if, for all PPT $\mathcal{A}$* that does not make any queries in step 4, *its probabilty of winning the experiment is at most* $1/2 + \mathrm{negl}(\lambda)$.

Let us define the query complexity *of an attack against the CCA-2 experiment as the number of decryption queries made in steps 2 and 4.*

We say that an encryption scheme is a k-decryption CCA-secure scheme if for all PPT $\mathcal{A}$ with query complexity at most k, its probability of winning the experiment is at most $1/2 + \mathrm{negl}(\lambda)$.

Definition A.3 ((One-query) PRFs)**.** *A pseudorandom function (PRF)* $(\mathsf{KeyGen}, \mathsf{PRF})$ *is a tuple of PPT algorithms with the following syntax:*

- $\mathsf{KeyGen}(1^\lambda)$ *outputs a key K;*
- $\mathsf{PRF}_K(x)$ *deterministically outputs some value y.*

We require the following property:

- *Pseudorandomness: for all PPT $\mathcal{A}$, consider the following experiment between $\mathcal{A}$ and a challenger:*
  1. *The challenger generates* $K \leftarrow \mathsf{KeyGen}(1^\lambda)$, *and flips a coin* $b \leftarrow \{0,1\}$.
  2. *$\mathcal{A}$ is allowed to repeatedly and adaptively make PRF queries on inputs x of his choice, and the challenger replies with* $\mathsf{PRF}_K(x)$ *if* $b = 0$, *or $R(x)$ if* $b = 1$ *where R is a truly uniform function.*
  3. *$\mathcal{A}$ ouputs a guess b'.*

  We say that $\mathcal{A}$ wins *the experiment if* $b = b'$. *We say that* $(\mathsf{KeyGen}, \mathsf{PRF})$ *is* pseudorandom *if, for all PPT $\mathcal{A}$, its probabilty of winning the experiment is at most* $1/2 + \mathrm{negl}(\lambda)$.

Let us define the query complexity *of an attack against the pseudorandomness experiment as the number of PRF queries made in step 2.*

We say that a PRF scheme is k-query secure *if, for all PPT $\mathcal{A}$* with query complexity at most k, *its probabilty of winning the experiment is negligible.*

We will optionally require the following property:

- *One-wise independence:* PRF *maps $n(\lambda)$-bit inputs to $m(\lambda)$-bit outputs. We say* $(\mathsf{KeyGen}, \mathsf{PRF})$ *is 1-wise independent if for any input* $x \in \{0,1\}^n$ *and any* $y \in \{0,1\}^m$*:*

$$\Pr\left[\mathsf{PRF}_K(x) = y\right] = \frac{1}{2^m},$$

  where the probability is over $K \leftarrow \mathsf{KeyGen}(1^\lambda)$.

Note that a one-wise independent PRF $(\mathsf{KeyGen}, \mathsf{PRF})$ can be generically constructed from a regular PRF $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{PRF}})$ as follows:

- $\mathsf{KeyGen}(1^\lambda)$: Sample $\overline{K} \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$, $k \leftarrow \{0,1\}^m$. Output $K = (\overline{K}, k)$.
- $\mathsf{PRF}_K(x) = \overline{\mathsf{PRF}}_{\overline{K}}(x) \oplus k$.

Definition A.4 ((One-time) MACs)**.** *A MAC* $(\mathsf{KeyGen}, \mathsf{MAC}, \mathsf{Verify})$ *is a tuple of PPT algorithms with the following syntax:*

- $\mathsf{KeyGen}(1^\lambda)$ *outputs a secret key* sk*;*
- $\mathsf{MAC}(\mathsf{sk}, m)$ *outputs a tag* σ*;*
- $\mathsf{Verify}(\mathsf{sk}, m, \sigma)$ *outputs* accept *or* reject.

We require two properties of the MAC:

- *Correctness: for all messages* m*;*

$$\Pr\left[\mathsf{Verify}(\mathsf{sk}, m, \mathsf{MAC}(\mathsf{sk}, m)) = \mathsf{accept} \quad \middle| \quad \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)\right] \geq 1 - \mathrm{negl}(\lambda).$$

- *Unforgeability: for all PPT* $\mathcal{A}$*, consider the following experiment between* $\mathcal{A}$ *and a challenger:*

  1. *The challenger generates* $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and sends* vk *to* $\mathcal{A}$*.*
  2. $\mathcal{A}$ *is allowed to repeatedly and adaptively make (1) authentication queries for messages* m *of his choice, and the challenger replies with* $\mathsf{MAC}(\mathsf{sk}, m)$*, and (2) verification queries for message-tag pairs* (m, σ) *of his choice, and the challenger replies with* $\mathsf{Verify}(\mathsf{sk}, m, \sigma)$
  3. $\mathcal{A}$ *ouputs a forgery* (m^*, σ^*)*.*

  We say that $\mathcal{A}$ wins *the experiment if (1)* $\mathcal{A}$ *has not queried* m^* *in step 2 as an authentication query, and (2)* $\mathsf{Verify}(\mathsf{sk}, m^*, \sigma^*) = \mathsf{accept}$*. We say that* $(\mathsf{KeyGen}, \mathsf{MAC}, \mathsf{Verify})$ *is* unforgeable *(against adaptive authentication and verification queries) if, for all PPT* $\mathcal{A}$*, its probabilty of winning the experiment is negligible.*

Let us define the query complexity *of an attack against the unforgeability experiment as the number of* signing *queries made in step 2.*[22]

We say that a MAC is one-time secure *if, for all PPT* $\mathcal{A}$ with query complexity at most 1*, its probabilty of winning the experiment is negligible.*

Definition A.5 ((One-time) Symmetric-Key Encryption)**.** *A symmetric-key encryption scheme* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a tuple of PPT algorithms with the following syntax:*

- $\mathsf{KeyGen}(1^\lambda)$ *outputs a secret key* sk*;*
- $\mathsf{Enc}(\mathsf{sk}, m)$ *outputs a ciphertext* ct*;*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *outputs a message* m'*.*

We require two properties of the encryption scheme:

- *Correctness: for all messages* m*;*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m \quad \middle| \quad \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)\right] \geq 1 - \mathrm{negl}(\lambda).$$

- *CCA-2-Security: for all PPT* $\mathcal{A}$*, consider the following experiment between* $\mathcal{A}$ *and a challenger:*

  1. *The challenger generates* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and sends* pk *to* $\mathcal{A}$*.*
  2. $\mathcal{A}$ *is allowed to repeatedly and adaptively make encryption queries for messages* m *of his choice, and the challenger replies with* $\mathsf{Enc}(\mathsf{sk}, m)$*.*
  3. $\mathcal{A}$ *sends a* challenge query *with messages* m_0 *and* m_1 *to the challenger. The challenger picks a random bit* $b \leftarrow \{0, 1\}$ *and sends back to the adversary* $\mathsf{ct}_b^* = \mathsf{Enc}(\mathsf{pk}, m_b)$ *to* $\mathcal{A}$*.*
  4. $\mathcal{A}$ *is again allowed to repeatedly and adaptively make encryption queries for messages* m *of his choice, and the challenger replies with* $\mathsf{Enc}(\mathsf{sk}, m)$*.*

[22] We do not count verification queries in this work.

5. $\mathcal{A}$ *outputs a guess* $b' \in \{0,1\}$.

We say that $\mathcal{A}$ wins *the experiment if* $b = b'$. *We say that* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is secure if, for all PPT* $\mathcal{A}$*, its probabilty of winning the experiment is at most* $1/2 + \mathrm{negl}(\lambda)$.

Let us define the query complexity *of an attack against the security experiment as the number of encryption queries made in steps 2 and 4.*

We say that an encryption scheme is k*-time secure if for all PPT* $\mathcal{A}$ *with query complexity at most* $k-1$*, its probability of winning the experiment is at most* $1/2 + \mathrm{negl}(\lambda)$.