Secure Quantum Extraction Protocols

Prabhanjan Ananth
UCSB [*]

Rolando L. La Placa
MIT [†]

Abstract

Knowledge extraction, typically studied in the classical setting, is at the heart of several cryptographic protocols. The prospect of quantum computers forces us to revisit the concept of knowledge extraction in the quantum setting.

We introduce the notion of secure quantum extraction protocols. A secure quantum extraction protocol for an NP relation $\mathcal{R}$ is a classical interactive protocol between a sender and a receiver, where the sender gets the instance $\mathbf{z}$, witness $\mathbf{w}$ while the receiver only gets the instance $\mathbf{z}$. For any efficient quantum adversarial sender (who follows the protocol but can choose its own randomness), there exists a quantum extractor that can extract a witness $\mathbf{w}'$ such that $(\mathbf{z}, \mathbf{w}') \in \mathcal{R}$ while a malicious receiver should not be able to output any valid witness. We study and construct two types of secure quantum extraction protocols.

- Quantum extraction protocols secure against quantum malicious receivers based on quantum fully homomorphic encryption satisfying some mild properties and quantum hardness of learning with errors. In this construction, we introduce a non black box technique in the quantum setting. All previous extraction techniques in the quantum setting were solely based on quantum rewinding.
- Quantum extraction protocols secure against classical malicious receivers based on quantum hardness of learning with errors. Moreover, our construction has the property that a malicious receiver cannot later, long after the protocol has been executed, use a quantum computer to extract a valid witness from the transcript of the protocol.

As an application, based on the quantum hardness of learning with errors, we present a construction of constant round quantum zero-knowledge argument systems for NP that guarantee security even against quantum malicious verifiers; however, our soundness only holds against classical probabilistic polynomial time adversaries. Prior to our work, such protocols were known based, additionally, on the assumptions of decisional Diffie-Hellman (or other cryptographic assumptions that do not hold against polynomial time quantum algorithms).

[*] prabhanjan@cs.ucsb.edu
[†] rlaplaca@mit.edu

1 Introduction

Knowledge extraction is a useful concept employed in many constructions of classical zero-knowledge and secure two-party and multi-party computation protocols. The seminal work of Feige, Lapidot and Shamir [FLS99] shows how to leverage knowledge extraction to construct zero-knowledge protocols. Traditional simulation-based notions necessarily require the simulator to be able to extract the inputs of the adversaries to argue the security of secure computation protocols.

Typically, knowledge extraction is formalized by defining a knowledge extractor that given access to the adversarial machine, outputs a trapdoor that is related to the input of the adversary. The prototypical extraction technique employed in several cryptographic protocols is rewinding. In the rewinding technique, the extractor, with oracle access to the adversary, can rewind the adversary to a previous state to obtain more than one protocol transcript which in turn gives the ability to the extract from the adversary. While rewinding has proven to be quite powerful, it has several limitations [GK96]. Over the years, researchers realized the importance of circumventing the disadvantages of rewinding and in the process, proposed new extraction techniques; a couple of them include: (i) non black-box techniques [Bar01a]: the extractor has access to the description of the adversary and, (ii) super-polynomial extraction [Pas03]: the extractor is allowed to run in super-polynomial time.

Extracting from Quantum Adversaries. The prospect of quantum computers introduces new challenges in the design of zero-knowledge and secure computation protocols. As a starting step towards designing these protocols, we need to address the challenge of extracting from quantum adversaries. So far, the only technique used to extract from quantum adversaries is quantum rewinding [Wat09a], which has already been studied by a few works [Wat09a, JKMR06a, Unr12, ARU14, Unr16] in the context of quantum zero-knowledge protocols.

Rewinding a quantum adversary, unlike its classical counterpart, turns out to be tricky in part due to the no-cloning theorem. Informally speaking, the main issue with quantum rewinding is that if the adversary uses a quantum state to generate messages in a protocol, it could potentially perform operations that would destroy its state, thus it wouldn't be able to rewind and answer other queries. To do this, it would need to copy the state before generating the messages. As a result, the existing quantum rewinding techniques tend to be "oblivious" [Unr12], to rewind the adversary back to an earlier point, the extract should necessarily forget all the information it has gained from that point onwards. As a result of these subtle issues, the analysis of quantum rewinding turns out to be quite involved making it difficult to use in security proofs. Moreover, existing quantum rewinding techniques [Wat09a, Unr12] poses a bottleneck towards achieving a constant round extraction technique; we will touch upon this later.

In order to advance the progress of constructing quantum-secure (post-quantum) cryptographic protocols, it is necessary that we look beyond quantum rewinding and explore new quantum extraction techniques.

1.1 Results

We introduce and study new techniques that enable us to extract from quantum adversaries.

Our Notion: Secure Quantum Extraction Protocols. We formalize this by first introducing the notion of secure quantum extraction protocols. This is a classical interactive protocol between a sender and a receiver and is associated with a NP relation. The sender has an NP instance and a witness while the receiver only gets the NP instance. In terms of properties, we require the following to hold:

- **Extractability**: An extractor, implemented as a quantum polynomial time algorithm, can extract a valid witness from an adversarial sender. We model the adversarial sender as a quantum polynomial time algorithm that follows the protocol but is allowed to choose its randomness; in the classical setting, this is termed as semi-malicious and we call this semi-malicious quantum adversaries. We also require the additional guarantee that the adversarial sender cannot distinguish whether its interacting with the honest receiver or an extractor.

- **Zero-Knowledge**: A malicious receiver should not be able to extract a valid witness after interacting with the sender. The malicious receiver can either be a classical probabilistic polynomial time algorithm or a quantum polynomial time algorithm. Correspondingly, there are two notions of quantum extraction protocols we study: quantum extraction protocols secure against quantum adversarial receiver (qQEXT) and quantum extraction protocols secure against classical adversarial receiver (cQEXT).

There are two reasons why we only study extraction against semi-malicious adversaries, instead of malicious adversaries (who can arbitrarily deviate from the protocol): first, the constructions tend to be simpler and second, in the classical setting, there are works in the classical setting that show how to leverage extraction from semi-malicious adversaries to achieve zero-knowledge protocols [BCPR16a, BKP19] or secure two-party computation protocols [AJ17].

Quantum extraction protocols are interesting even if we only had classical adversaries, as they present a new method for proving zero-knowledge. The zero-knowledge simulator requires more computational power than the adversaries. Allowing quantum simulators in the classical setting [KK19] is another way to achieve this asymmetry between the power of the simulator and the adversary besides the few mentioned before (rewinding, superpolynomial, or non-black box). Furthermore, quantum simulators capture the notion of knowledge that could be learnt if a malicious verifier had access to a quantum computer.

Quantum-Lasting Security. A potential concern regarding the security of cQEXT protocols is that the classical malicious receiver participating in the cQEXT protocol could later, long after the protocol has been executed, could use a quantum computer to learn the witness of the sender from the transcript of the protocol and its own private state. For instance, the transcript could contain an ElGamal encryption of the witness of the sender; while a malicious classical receiver cannot break it, after the protocol is completed, it could later use a quantum computer to learn the witness. This is especially interesting in the event (full-fledged) quantum computers might become more easily accessible in the future. First introduced by Unruh [Unr13], we study the concept of quantum-lasting security; any quantum polynomial time (QPT) adversary given the transcript and the private state of the malicious receiver, should not be able to learn the witness of the sender.

Constructions. We propose constructions of qQEXT and cQEXT protocols.

We show how to construct a constant round quantum extraction protocol secure against quantum adversaries.

Theorem 1 (Informal)**.** *Assuming quantum hardness of learning with errors and a quantum fully homomorphic encryption scheme (for arbitrary poly-time computations), satisfying, (1) perfect correctness for classical messages and, (2) ciphertexts of poly-sized classical messages have a poly-sized classical description, there exists a constant round quantum extraction protocol secure against quantum poly-time adversaries.*

We first clarify what we mean by perfect correctness. For every valid public key, every valid fresh ciphertext of a classical message can always be decrypted correctly. Moreover, we require that for every valid fresh ciphertext, of a classical message, the evaluated ciphertext can be decrypted correctly with probability negligibly close to 1. We note that the works of [Mah18a, Bra18a] give candidates for quantum fully homomorphic encryption schemes satisfying both the above properties.

En route to proving the above theorem, we introduce a new non black extraction technique in the quantum setting. Non black box extraction overcomes the disadvantage quantum rewinding poses in achieving constant round extraction; the quantum rewinding employed by [Wat09a] requires polynomially many rounds (due to sequential repetition) or constant rounds with non-negligible gap between extraction and verification error [Unr12].

The novelty of our approach involves identifying the appropriate classical non black box extraction technique and then porting it to the quantum setting; in particular, we rely upon the work of [BKP19] who introduced a new non black box technique in the context of designing classical protocols. For instance, it is unclear how to utilize the well known non black box technique of Barak [Bar01b]; at a high level, the idea of Barak [Bar01b] is to commit to the code of the verifier and then prove using a succinct argument system that either the instance is in the language or it has the code of the verifier. In our setting, the verifier is a quantum circuit which means that we would require succinct arguments for quantum computations which we currently don't know how to achieve.

We also present a construction of quantum extraction protocols secure against classical adversaries (cQEXT). This result is incomparable to the above result; on one hand, it is a weaker setting but on the other hand, the security of this construction can solely be based on the hardness of learning with errors while the above result requires stronger assumptions.

Theorem 2 (Informal)**.** *Assuming quantum hardness of learning with errors, there exists a constant round quantum extraction protocol secure against classical PPT adversaries and satisfying quantum-lasting security.*

Our main idea is to turn the "test of quantumness" protocol introduced in [BCM+18] into a quantum extraction protocol; en route, we use cryptographic tools to achieve this. In fact, our techniques are general enough that it might be useful to turn any protocol that can verify a quantum computer versus a classical computer into a quantum extraction protocol secure against classical adversaries. In more detail, we use a cryptographic protocol that releases the witness to the receiver if and only if the receiver has passed the test of quantumness.

Application: Constant Round QZK for NP with Classical Soundness. As an application, we show how to construct constant quantum zero-knowledge argument systems secure against quantum verifiers based on quantum hardness of learning with errors; however, the soundness is still against classical PPT adversaries. Previously, no such result was known[1].

Theorem 3 (Constant Round Quantum ZK with Classical Soundness; Informal)**.** *Assuming quantum hardness of learning with errors, there exists a constant round black box quantum zero-knowledge system with negligible soundness against classical PPT algorithms.*

1.2 Technical Overview

Quantum extraction with security against classical receivers: Overview. As mentioned earlier, our main idea is to turn the "test of quantumness" from [BCM+18] into an extraction protocol. Our starting point is a noisy trapdoor claw-free function (NTCF) family [Mah18a, Mah18b, BCM+18], parameterized by key space $\mathcal{K}$, input domain $\mathcal{X}$ and output domain $\mathcal{Y}$. Using a key $\mathbf{k} \in \mathcal{K}$, NTCFs allows for computing the distributions, denoted by $f_{\mathbf{k},0}(x) \in \mathcal{Y}$ and $f_{\mathbf{k},1}(x) \in \mathcal{Y}$ [2], where $x \in \mathcal{X}$. Using a trapdoor td associated with a key $\mathbf{k}$, any y in the support of $f_{\mathbf{k},b}(x)$, can be efficiently inverted to obtain x. Moreover, there are "claw" pairs (x_0, x_1) such that $f_{\mathbf{k},0}(x_0)$ outputs the same distribution as $f_{\mathbf{k},1}(x_1)$. Roughly speaking, the security property states that it is computationally hard even for a quantum computer to simultaneously produce $y \in \mathcal{Y}$, values (b, x_b) and (d, u) such that $f_{\mathbf{k},b}(x_b) = y$ and $\langle d, J(x_0) \oplus J(x_1) \rangle = u$, where $J(\cdot)$ is an efficienctly computable injective function mapping $\mathcal{X}$ into bit strings. What makes this primitive interesting is its quantum capability that we will discuss when we recall below the test of [BCM+18].

Using NTCFs, [BCM+18] devised the following test: the classical verifier, who wants to test whether the server its interacting with is quantum or classical, first generates a key $\mathbf{k}$ along with a trapdoor td associated with a noisy trapdoor claw-free function (NTCF) family. It sends $\mathbf{k}$ to the server who responds back with $y \in \mathcal{Y}$. Now, the classical verifier can ask the server to show either a pre-image x_b along with bit b such that $f_{\mathbf{k},b}(x_b) = y$ OR a vector d along with bit u such that $\langle d, J(x_0) \oplus J(x_1) \rangle = u$. Intuitively, since the server does not know, at the point when it sends y, whether it will be queried for (b, x_b) or (d, u), by the security of NTCFs, it can only answer one of the queries. While the quantum capability of NTCFs allows for a quantum server to maintain a superposition of a claw at the time it sent y and depending on the query made by the verifier it can then perform the appropriate quantum operations to answer the verifier; thus it will always pass the test.

A natural attempt to achieve extraction is the following: the sender takes the role of the client and the receiver takes the role of the server and if the test passes, the sender sends the witness to the receiver. This solution guarantees zero-knowledge against classical receivers. To extract, we exploit the quantum capability of NTCFs to pass the test and thus obtain the witness. Unfortunately, a semi-malicious sender can distinguish whether its interacting with a honest classical reciever or an extractor. There are two issues to solve here: (i) first, the sender should not know if the receiver

[1] It is conceivable that existing constant round zero-knowledge protocols can be instantiated with quantum-**in**secure assumptions (such as DDH) and additionally quantum hardness of learning with errors to achieve the same result. However, our result solely relies upon the quantum hardness of learning with errors.

[2] The efficient implementation of f only approximately computes f and we denote this by f'. We ignore this detail for now.

passed the test or not (the classical receiver will never pass the test) and, (ii) second, the sender should not know whether the receiver obtained the witness.

A first attempt to tackle both of these issues is to use a secure two party computation protocol where only the receiver receives the output; we later instantiate this with secure function evaluation [GHV10, BCPR16a] that guarantees security against quantum adversaries. The extraction protocol proceeds as follows: the sender sends key $\mathbf{k}$ and the receiver responds back with $y \in \mathcal{Y}$. The sender then sends the challenge bit w. Instead of the receiver answering with either (b, x_b) or (d, u); it inputs this to the two party secure computation protocol. The sender, on the other hand, inputs the witness w along with trapdoor td associated with $\mathbf{k}$. The functionality associated with the protocol checks if (b, x_b) is valid if $w = 0$ or if (d, u) if $w = 1$. If so, it outputs the witness. This ensures that the sender does not know whether the test passed or not and at the same time, the extractor receives the witness since it does pass the test. In fact, we can formalize this argument and show that indeed, extractability is satisfied. To argue zero-knowledge, we need to rely upon the security property of NTCFs which suggest that we need to ability to extract a valid (b, x_b) and a valid (d, u); since the receiver inputs these values only into the two-party secure computation protocol, it would seem that we need the ability to extract from this protocol. However, the instantiation of the two-party secure computation protocol we use does not guarantee efficient extraction. Thus, we circumvent this problem by adding an quantum-secure extractable commitment before the secure two-party computation protocol begins. The receiver instead of feeding (b, x_b) or (d, u) directly into the two-party protocol, it commits using an extractable commitment scheme [PW09]; an extractable commitment allows for a rewinding extractor to extract from the committer. This enables the zero-knowledge simulator to be able to extract both (b, x_b) and (d, u) using an extractable commitment scheme if indeed the classical receiver passes the test and we use this to contradict the security of NTCFs. Otherwise, if the test did not pass, we can invoke the security of the two-party protocol to argue that the receiver never receives the witness. There are a couple of issues that we defer to the technical sections: firstly, a classical adversary can pass the test of [BCM⁺18] with probability $1/2$ but we require that the classical adversary passes it with only negligible probability and secondly, we would need an extratable commitment scheme that is quantum secure; we instantiate the construction of extractable commitment scheme of [PW09] with perfectly binding commitments that satisfies computational hiding property against quantum adversaries.

QZK with classical provers. We go from cQEXT to quantum zero-knowledge argument with classical verifier by the FLS trick [FLS99]. The argument system has 2 phases: (1) prover and verifier engage in a cQEXT protocol, with the prover taking the role of the receiver and the verifier that of the sender, (2) prover and verfier engage in a witness indistinguishable (WI) proof system protocol. In the WI phase, instead of showing that $\mathbf{z} \in \mathcal{L}$, the prover will show a statement of the form $(\mathbf{z} \in \mathcal{L}) \bigvee (\mathsf{td}' = \mathsf{td})$, where td is a trapdoor that can be extracted from the verifier in the cQEXT protocol. BY security of cQEXT, the adversarial classical prover cannot find the trapdoor td, so it will not be able to prove anything in the WI round unless it has a witness to the statement $\mathbf{z} \in \mathcal{L}$. However, a quantum simulator can first extract td from the cQEXT protocol, and then use the trapdoor as the witness in the WI rounds.

To implement this idea, we need to deal with a possibly malicious sender that engaged in the cQEXT protocol, and not just a semi-malicious one. To do this, we again use extractable commitments after the cQEXT protocol is completed, in order for the prover to commit to the witness that it will use in the WI. After this commitment is sent, the verifier is required to reveal all the random coins it used in the cQEXT, so that the prover can check that the verifier behaved honestly in the cQEXT. If this is indeed the case, then the prover proceeds with the WI stage, othwerwise it aborts. The malicious verifier could try aborting in the cQEXT protocol, but our construction satisfies a stronger extractability property: a semi-malicious sender cannot detect if it is interacting with an extractor or a verifier even if it is allowed to abort. If the verifier aborts in the cQEXT, then the simulator would abort the same way that the honest prover would. On the other hand, if the verifier completes the protocol, it will have to reveal the random coins of the cQEXT protocol. At this point, both the simulator and the prover can abort if the verifier does not provide valid random coins.

Quantum extraction with security against quantum receivers: Overview. To construct a quantum extraction protocol secure against quantum receivers (qQEXT), we consider a fully homomorphic encryption qFHE scheme[3] having that allows for public homomorphic evaluation of quantum circuits. For the current discussion, we assume that qFHE is 2-circular **in**secure (we show later how to replace this with lockable obfuscation): that is, given $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{SK}_2)$ (i.e., encryption of SK_2 under PK_1), $\mathsf{qFHE.Enc}(\mathsf{PK}_2, \mathsf{SK}_1)$, where $(\mathsf{PK}_1, \mathsf{SK}_1)$ and $(\mathsf{PK}_2, \mathsf{SK}_2)$ are independently generated public key-secret key pairs, we can recover SK_1 and SK_2. Using this, we attempt to construct qQEXT protocols as follows:

- The sender, on input instance $\mathbf{z}$ and witness $\mathbf{w}$, sends three ciphertexts: $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{td})$, $\mathsf{CT}_2 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathbf{w})$ and $\mathsf{CT}_3 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_2, \mathsf{SK}_1)$.

- The receiver sends td'.

- If $\mathsf{td}' = \mathsf{td}$ then the sender outputs SK_2.

A quantum extractor with non black box access to the private (quantum) state of S, after receiving $\mathsf{CT}_1, \mathsf{CT}_2$ and CT_3, encrypts the private (quantum) state and homomorphically evaluates S on CT_1 and the result is $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{SK}_2)$. Now, that the extractor has both $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{SK}_2)$ and $\mathsf{qFHE.Enc}(\mathsf{PK}_2, \mathsf{SK}_1)$, it can recover $\mathsf{SK}_1, \mathsf{SK}_2$ using the 2-circular **in**security of qFHE. Now it can decrypt CT_1 to recover the witness $\mathbf{w}$! This approach sidesteps the quantum rewinding issues mentioned earlier, as we never require the sender to rewind to a previous state in order to recover the witness.

The ability that the extractor recovers the witness is not alone sufficient; we need to argue that the transcript generated by the extractor is indistinguishable from the one generated during the interaction of the sender and the honest receiver. However, the transcript generated by the extractor is encrypted under the public key PK_1 of qFHE. Luckily, the extractor does recover SK_1 and thus, can decrypt the encrypted final state of the sender to obtain the transcript of the protocol[4]. This still does not guarantee the indistinguishability of transcripts generated by the extractor and the the transcripts generated by the honest receiver. This is because, the transcript output by the extracor

[3] Recall that a classical FHE scheme [G$^+$09, BV14] allows for publicly evaluating an encryption of a message x using a circuit C to obtain an encryption of $C(x)$.

[4] We assume that without loss of generality, the final state of the sender contains the transcript of the protocol.

contains SK_2 as the third message but not the one generated by the honest receiver. In particular, we need a mechanism where the sender does not know whether the receiver received SK_2 or not. We already encountered a similar issue when we were devising cQEXT protocols; we solved the earlier issue using secure two-party computation (instantiated using secure function evaluation) and we use the same solution here as well. Instead of the receiver sending td' directly to S, it instead feeds this into the two-party protocol which checks if $\mathsf{td}' = \mathsf{td}$ and if so releases SK_2 to the receiver. That is, we have the following modified template.

- The sender, on input instance $\mathbf{z}$ and witness $\mathbf{w}$, sends three ciphertexts: $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{td})$, $\mathsf{CT}_2 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathbf{w})$ and $\mathsf{CT}_3 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_2, \mathsf{SK}_1)$.

- The sender and the receiver executes a secure two-party computation protocol, where the receiver feeds td' and the sender feeds in $(\mathsf{td}, \mathbf{w})$. After the protocol finishes, the receiver recovers $\mathbf{w}$ if $\mathsf{td}' = \mathsf{td}$.

We can argue that this still allows for extraction – the extractor now computes the two-party protocol homomorphically – moreover, the secure computation protocol guarantees indistinguishability of the transcripts.

Arguing zero-knowledge involves more challenges: the adversarial receiver could "maul" the ciphertext it receives in the first round into the messages of the secure two-party computation protocol. To prevent this, we require the receiver to commit to td' in parallel with the secure computation protocol and furthermore, the randomness used in this commitment is committed even before it sees the first message. Using this, we demonstrate a quantum simulator that demonstrates the quantum zero-knowledge property of the protocol.

While the above protocol is a candidate for quantum extraction protocol secure against quantum receivers; it is still unsatisfactory since we assume a quantum FHE scheme satisfying 2-circular **in**security.We show how to replace 2-circular insecure QFHE with *any* QFHE scheme (satisfying some mild properties already satisfied by existing candidates) and lockable obfuscation for classical circuits. A lockable obfuscation scheme is an obfuscation scheme for a specific class of functionalities called compute-and-compare functionalities; a compute-and-compare functionality is parameterized by C, α (lock), β such that on input x, it outputs β if $C(x) = \alpha$. As long as α is sampled uniformly at random and independently of C, lockable obfuscation completely hides the circuit C, α and β. The idea to replace 2-circular insecure QFHE with lockable obfuscation is as follows: obfuscate the circuit, with secret key SK_2, ciphertext $\mathsf{qFHE.Enc}(\mathsf{SK}_2, r)$ hardwired, that takes as input $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{SK}_2)$, decrypts it to obtain SK_2', then decrypts $\mathsf{qFHE.Enc}(\mathsf{SK}_2, r)$ to obtain r' and outputs SK_1 if $r' = r$. If the adversary does not obtain $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{SK}_2)$ then we can first invoke the security of lockable obfuscation to remove SK_1 from the obfuscated circuit and then it can replace $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathbf{w})$ with $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \perp)$. The idea of using fully homomorphic encryption along with lockable obfuscation to achieve non black box extraction was introduced, in the classical setting, by [BKP19].

Unlike our cQEXT construction, the non black box technique used for qQEXT does not directly give us a constant round quantum zero-knowledge protocol for NP. This is because an adversarial verifier that aborts can distinguish between the extractor or the honest prover (receiver in qQEXT). The main issue is that the extractor runs the verifier homomorphically, so it cannot detect if the verifier aborted at any point in the protocol without decrypting. But if the verifier aborted, the extractor wouldn't be able to decrypt in the first place – it could rewind but then this would destroy the initial quantum auxiliary state.

1.3 Related Work

Quantum Rewinding. Watrous [Wat09b] introduced quantum analogue of the rewinding technique. Later, Unruh [Unr12] introduced yet another notion of quantum rewinding with the purpose of constructing quantum zero-knowledge proofs of knowledge. Unruh's rewinding does have extractability, but it requires that the underlying protocol satisfy *strict soundness*. Furthermore, the probability that the extractor succeeds is not negligibly close to 1. The work of [ARU14] shows that relative to an oracle, many classical zero-knowledge protocols are quantum insecure, and that the strict soundness condition from [Unr12] is necessary in order for a sigma protocol to be a quantum proofs of knowledge.

Quantum and Classical Zero-Knowledge. Zero-knowledge against quantum adversaries was first studied by Watrous [Wat09b]. He showed how the GMW protocol [GMW86] for graph 3-colorability is still zero-knowledge against quantum verifiers. Other work [HKSZ08, CCKV08, JKMR06b, Kob08, Mat06, Unr12] has extended the study of classical protocols that are quantum zero-knowledge, and more recently, Broadbent et al. [BJSW16] extended the notion of zero-knowledge to QMA languages. By using ideas from [Mah18b] to classically verify quantum computation, the protocol in [BJSW16] was adapted to obtained classical argument systems for quantum computation in [VZ19]. All known protocols, with non-negligible soundness error, take non-constant rounds.

On the other hand, zero knowledge proof and argument systems have been extensively studied in classical cryptography. In particular, a series of recent works [BCPR16b, BBK+16, BKP18, BKP19] resolved the round complexity of zero knowledge argument systems.

2 Preliminaries

We denote the security parameter by λ. We denote (classical) computational indistiguishability of two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ by $\mathcal{D}_0 \approx_{c,\varepsilon} \mathcal{D}_1$. In the case when ε is negligible, we drop ε from this notation.

Languages and Relations. A language $\mathcal{L}$ is a subset of $\{0,1\}^*$. A relation $\mathcal{R}$ is a subset of $\{0,1\}^* \times \{0,1\}^*$. We use the following notation:

- Suppose $\mathcal{R}$ is a relation. We define $\mathcal{R}$ to be *efficiently decidable* if there exists an algorithm A and fixed polynomial p such that $(x,w) \in \mathcal{R}$ if and only if $A(x,w) = 1$ and the running time of A is upper bounded by $p(|x|,|w|)$.

- Suppose $\mathcal{R}$ is an efficiently decidable relation. We say that $\mathcal{R}$ is a NP relation if $\mathcal{L}(\mathcal{R})$ is a NP language, where $\mathcal{L}(\mathcal{R})$ is defined as follows: $x \in \mathcal{L}(R)$ if and only if there exists w such that $(x,w) \in \mathcal{R}$ and $|w| \leq p(|x|)$ for some fixed polynomial p.

2.1 Learning with Errors

In this work, we are interested in the decisional learning with errors (LWE) problem. This problem, parameterized by n, m, q, χ, where $n, m, q \in \mathbb{N}$, and for a distribution χ supported over $\mathbb{Z}$ is to distinguish between the distributions $(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times 1}$, $\mathbf{e} \overset{\$}{\leftarrow} \chi^{m\times 1}$ and $\mathbf{u} \leftarrow \mathbb{Z}_q^{m\times 1}$. Typical setting of m is $n\log(q)$, but we also consider $m = \mathrm{poly}(n\log(q))$.

We base the security of our constructions on the quantum hardness of learning with errors problem.

2.2 Notation and General Definitions

For completeness, we present some of the basic quantum definitions, for more details see [NC02].

Quantum states and channels. Let $\mathcal{H}$ be any finite Hilbert space, and let $L(\mathcal{H}) := \{\mathcal{E} : \mathcal{H} \to \mathcal{H}\}$ be the set of all linear operators from $\mathcal{H}$ to itself (or endomorphism). Quantum states over $\mathcal{H}$ are the positive semidefinite operators in $L(\mathcal{H})$ that have unit trace. Quantum channels or quantum operations acting on quantum states over $\mathcal{H}$ are completely positive trace preserving (CPTP) linear maps from $L(\mathcal{H})$ to $L(\mathcal{H}')$ where $\mathcal{H}'$ is any other finite dimensional Hilbert space.

A state over $\mathcal{H} = \mathbb{C}^2$ is called a qubit. For any $n \in \mathbb{N}$, we refer to the quantum states over $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ as n-qubit quantum states. To perform a standard basis measurement on a qubit means projecting the qubit into $\{|0\rangle, |1\rangle\}$. A quantum register is a collection of qubits. A classical register is a quantum register that is only able to store qubits in the computational basis.

A unitary quantum circuit is a sequence of unitary operations (unitary gates) acting on a fixed number of qubits. Measurements in the standard basis can be performed at the end of the unitary circuit. A (general) quantum circuit is a unitary quantum circuit with 2 additional operations: (1) a gate that adds an ancilla qubit to the system, and (2) a gate that discards (trace-out) a qubit from the system. A quantum polynomial-time algorithm (QPT) is a uniform collection of quantum circuits $\{C_n\}_{n\in\mathbb{N}}$.

Quantum Computational Indistinguishability. When we talk about quantum distinguishers, we need the following definitions, which we take from [Wat09b].

Definition 4 (Indistinguishable collections of states)**.** *Let I be an infinite subset $I \subset \{0,1\}^*$, let $p : \mathbb{N} \to \mathbb{N}$ be a polynomially bounded function, and let ρ_x and σ_x be $p(|x|)$-qubit states. We say that $\{\rho_x\}_{x\in I}$ and $\{\sigma_x\}_{x\in I}$ are **quantum computationally indistinguishable collections of quantum states** if for every QPT $\mathcal{E}$ that outputs a single bit, any polynomially bounded $q : \mathbb{N} \to \mathbb{N}$, and any auxiliary $q(|x|)$-qubits state ν, and for all $x \in I$, we have that*

$$\left|\Pr\left[\mathcal{E}(\rho_x \otimes \nu) = 1\right] - \Pr\left[\mathcal{E}(\sigma_x \otimes \nu) = 1\right]\right| \leq \epsilon(|x|)$$

for some negligible function $\epsilon : \mathbb{N} \to [0,1]$. We use the following notation

$$\rho_x \approx_{Q,\epsilon} \sigma_x$$

and we ignore the ϵ when it is understood that it is a negligible function.

Definition 5 (Indistinguishability of channels). *Let I be an infinite subset $I \subset \{0,1\}^*$, let $p, q : \mathbb{N} \longrightarrow \mathbb{N}$ be polynomially bounded functions, and let $\mathcal{D}_x, \mathcal{F}_x$ be quantum channels mapping $p(|x|)$-qubit states to $q(|x|)$-qubit states. We say that $\{\mathcal{D}_x\}_{x\in I}$ and $\{\mathcal{F}_x\}_{x\in I}$ are **quantum computationally indistinguishable collection of channels** if for every QPT $\mathcal{E}$ that outputs a single bit, any polynomially bounded $t : \mathbb{N} \longrightarrow \mathbb{N}$, any $p(|x|) + t(|x|)$-qubit quantum state ρ, and for all $x \in I$, we have that*

$$\left|\Pr\left[\mathcal{E}\left((\mathcal{D}_x \otimes \mathsf{Id})(\rho)\right) = 1\right] - \Pr\left[\mathcal{E}\left((\mathcal{F}_x \otimes \mathsf{Id})(\rho)\right) = 1\right]\right| \leq \epsilon(|x|)$$

for some negligible function $\epsilon : \mathbb{N} \longrightarrow [0,1]$. We will use the following notation

$$\mathcal{D}_x(\cdot) \approx_{Q,\epsilon} \mathcal{F}_x(\cdot)$$

and we ignore the ϵ when it is understood that it is a negligible function.

Interactive Models. We model an interactive protocol between a prover, Prover, and a verifier, Verifier, as follows. There are 2 registers $\mathrm{R}_{\mathsf{Prover}}$ and $\mathrm{R}_{\mathsf{Verifier}}$ corresponding to the prover's and the verifier's private registers, as well as a message register, R_{M}, which is used by both Prover and Verifier to send messages. In other words, both prover and verifier have access to the message register. We denote the size of a register R by $|\mathsf{R}|$ – this is the number of bits or qubits that the register can store. We will have 2 different notions of interactive computation. Our honest parties will perform classical protocols, but the adversaries will be allowed to perform quantum protocols with classical messages.

1. **Classical protocol:** An interactive protocol is classical if $\mathrm{R}_{\mathsf{Prover}}$, $\mathrm{R}_{\mathsf{Verifier}}$, and R_{M} are classical, and Prover and Verifier can only perform classical computation.

2. **Quantum protocol with classical messages:** An interactive protocol is quantum with classical messages if either one of $\mathrm{R}_{\mathsf{Prover}}$ or $\mathrm{R}_{\mathsf{Verifier}}$ is a quantum register, and R_{M} is classical. Prover and Verifier can perform quantum computations if their respective private register is quantum, but they can only send classical messages.

When a protocol has classical messages, we can assume that the adversarial party will also send classical messages. This is without loss of generality, because the honest party can enforce this condition by always measuring the message register in the computational basis before proceeding with its computations.

Non Black-Box Access. Let S be a QPT party (e.g. either prover or verifier in the above descriptions) involved in specific quantum protocol. In particular, S can be seen as a collection of QPTs, $S = (S_1, ..., S_\ell)$, where ℓ is the number of rounds of the protocol, and S_i is the quantum operation that S performs on the ith round of the protocol.

We say that a QPT Q has *non black-box access* to S, if Q has access to an efficient classical description for the operations that S performs in each round, $(S_1, ..., S_\ell)$, as well as access to the initial auxiliary inputs of S.

Interaction Channel. For a particular protocol $(\mathsf{Prover}, \mathsf{Verifier})$, the interaction between Prover and $\mathsf{Verifier}$ on input $\mathbf{z}$ induces a quantum channel $\mathcal{E}_{\mathbf{z}}$ acting on their private input states, ρ_{Prover} and $\sigma_{\mathsf{Verifier}}$. We denote the view of $\mathsf{Verifier}$ when interacting with Prover by

$$\mathsf{View}_{\mathsf{Verifier}}\left(\left\langle \mathsf{Prover}\left(\mathbf{z}, \rho_{\mathsf{Prover}}\right), \mathsf{Verifier}\left(\mathbf{z}, \sigma_{\mathsf{Verifier}}\right)\right\rangle\right),$$

and this view is defined as the verifiers output. Specifically,

$$\mathsf{View}_{\mathsf{Verifier}}\left(\left\langle \mathsf{Prover}\left(\mathbf{z}, \rho_{\mathsf{Prover}}\right), \mathsf{Verifier}\left(\mathbf{z}, \sigma_{\mathsf{Verifier}}\right)\right\rangle\right) := \mathsf{Tr}_{\mathsf{R}_{\mathsf{Prover}}}\left[\mathcal{E}_{\mathbf{z}}\left(\rho_{\mathsf{Prover}} \otimes \sigma_{\mathsf{Verifier}}\right)\right].$$

From the verifier's point of view, the interaction induces the channel $\mathcal{E}_{\mathbf{z},V}(\sigma) = \mathcal{E}_{\mathbf{z}}(\sigma \otimes \rho_{\mathsf{Prover}})$ on its private input state.

2.3 Perfectly Binding Commitments

A commitment scheme consists a classical PPT algorithm[5] Comm that takes as input security parameter 1^{λ}, input message x and outputs the commitment $\mathbf{c}$. There are two properties that need to be satisfied by a commitment scheme: binding and hiding. In this work, we are interested in commitment schemes that are perfectly binding and computationally hiding; we define both these notions below. We adapt the definition of computational hiding to the quantum setting.

Definition 6 (Perfect Binding). *A commitment scheme* Comm *is said to be perfectly binding if for every security parameter* $\lambda \in \mathbb{N}$, *there does not exist two messages* x, x' *with* $x \neq x'$ *and randomness* r, r' *such that* $\mathsf{Comm}(1^{\lambda}, x; r) = \mathsf{Comm}(1^{\lambda}, x'; r')$.

Definition 7 (Quantum-Computational Hiding). *A commitment scheme* Comm *is said to be computationally hiding if for sufficiently large security parameter* $\lambda \in \mathbb{N}$, *for any two messages* x, x', *the following holds:*

$$\left\{\mathsf{Comm}\left(1^{\lambda}, x\right)\right\} \approx_Q \left\{\mathsf{Comm}\left(1^{\lambda}, x'\right)\right\}$$

Instantiation. A construction of perfectly binding non-interactive commitments was presented in the works of [GHKW17, LS19] assuming the hardness of learning with errors. Thus, we have the following:

Lemma 8 ([GHKW17, LS19]). *Assuming the quantum hardness of learning with errors, there exists a construction of perfectly binding quantum-computational hiding non-interactive commitment schemes.*

2.4 Noisy Trapdoor Claw-Free Functions

Noisy trapdoor claw-free functions is a useful tool in quantum cryptography. Most notably, they are a key ingredient in the construction of certifiable randomness protocols [BCM+18], classical client quantum homomorphic encryption [Mah18a], and classifal verification of quantum computation [Mah18b]. We present the formal definition directly from [BCM+18].

[5] Typically, commitment schemes are also associated with a opening algorithm; we don't use the opening algorithm in our work.

Definition 9 (Noisy Trapdoor Claw-Free Functions)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets, let $D_{\mathcal{Y}}$ be the set of distributions over $\mathcal{Y}$, and let $\mathcal{K}$ be a finite set of keys. A collection of functions $\{f_{\mathbf{k},b} : \mathcal{X} \rightarrow D_{\mathcal{Y}}\}_{\mathbf{k}\in\mathcal{K},b\in\{0,1\}}$ is noisy trapdoor claw-free if*

- ***(Key-Trapdoor Generation):*** *There is a PPT* $\mathsf{Gen}(1^\lambda)$ *to generate a key and a corresponding trapdoor,* $\mathbf{k}, \mathsf{td}_{\mathbf{k}} \leftarrow \mathsf{Gen}(1^\lambda)$.
- *For all* $\mathbf{k} \in \mathcal{K}$
  - ***(Trapdoor):*** *For all $b \in \{0,1\}$, and any distinct $x, x' \in \mathcal{X}$, we have that* $\mathsf{Supp}(f_{\mathbf{k},b}(x)) \cap \mathsf{Supp}(f_{\mathbf{k},b}(x')) = \emptyset$*. There is also an efficient deterministic algorithm* Inv*, that for any* $y \in \mathsf{Supp}(f_{\mathbf{k},b}(x))$*, outputs* $x \leftarrow \mathsf{Inv}(\mathsf{td}_{\mathbf{k}}, b, y)$.
  - ***(Injective Pair):*** *There exists a perfect matching $\mathcal{R}_{\mathbf{k}} \subseteq \mathcal{X} \times \mathcal{X}$ such that $f_{\mathbf{k},0}(x_0) = f_{\mathbf{k},1}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_{\mathbf{k}}$*
- ***(Efficient Range Superposition):*** *For all $\mathbf{k} \in \mathcal{K}$ and $b \in \{0,1\}$, there exists functions $f'_{\mathbf{k},b} : \mathcal{X} \rightarrow D_{\mathcal{Y}}$ such that the following holds.*
  - *For all $(x_0, x_1) \in \mathcal{R}_{\mathbf{k}}$, and all $y \in$* $\mathsf{Supp}(f'_{\mathbf{k},b}(x_b))$*, the inversion algorithm still works, i.e.* $x_b \leftarrow \mathsf{Inv}(\mathsf{td}_{\mathbf{k}}, b, y)$ *and* $x_{b\oplus 1} \leftarrow \mathsf{Inv}(\mathsf{td}_{\mathbf{k}}, b \oplus 1, y)$.
  - *There is an efficient deterministic checking algorithm* $\mathsf{Chk} : \mathcal{K} \times \{0,1\} \times \mathcal{X} \times \mathcal{Y} \rightarrow \{0,1\}$ *such that* $\mathsf{Chk}(\mathbf{k}, b, x, y) = 1$ *iff* $y \in \mathsf{Supp}(f'_{\mathbf{k},b}(x))$
  - *For every $\mathbf{k} \in \mathcal{K}$ and $b \in \{0,1\}$,*

    $$\mathop{\mathbb{E}}_{x\leftarrow\mathcal{X}}\left(H^2\left(f_{\mathbf{k},b}(x), f'_{\mathbf{k},b}(x)\right)\right) \leq \mu(\lambda)$$

    for some negligible function μ, and where H^2 is the Hellinger distance.
  - *For any $\mathbf{k} \in \mathcal{K}$ and $b \in \{0,1\}$, there exists an efficient way to prepare the superposition*

    $$\frac{1}{\sqrt{|\mathcal{X}|}}\sum_{x\in\mathcal{X},y\in\mathcal{Y}}\sqrt{f'_{\mathbf{k},b}(x)(y)}|x\rangle|y\rangle$$
- ***(Adaptive Hardcore Bit)****: for all keys $\mathbf{k} \in \mathcal{K}$, for some polynomially bounded $w : \mathbb{N} \rightarrow \mathbb{N}$, the following holds.*
  - *For all $b \in \{0,1\}$ and for all $x \in \mathcal{X}$ there exists a set $G_{\mathbf{k},b,x} \subseteq \{0,1\}^{w(\lambda)}$, s.t. $\Pr_{d\leftarrow\{0,1\}^{w(\lambda)}}\left[d \notin G_{\mathbf{k},b,x}\right] \leq$ negl(λ). Furthermore, membership in $G_{\mathbf{k},b,x}$ can be checked given $t_{\mathbf{k}}, \mathbf{k}, b$ and x.*
  - *There is an efficiently computable injection $J : \mathcal{X} \rightarrow \{0,1\}^{w(\lambda)}$, that can be inverted efficiently in its range, and for which the following holds. Let*

    $$H_{\mathbf{k}} := \left\{(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1)))\,|b \in \{0,1\}, (x_0, x_1) \in \mathcal{R}_{\mathbf{k}}, d \in G_{\mathbf{k},0,x_0} \cap G_{\mathbf{k},1,x_1}\right\}$$

    $$\overline{H_{\mathbf{k}}} := \{(b, x_b, d, c)\,|(b, x, d, c \oplus 1) \in H_{\mathbf{k}}\}$$

For any QPT $\mathcal{A}$ there is a negligible function μ s.t.

$$\left|\Pr_{\mathbf{k},\mathsf{td}_{\mathbf{k}}}[\mathcal{A}(\mathbf{k}) \in H_{\mathbf{k}}] - \Pr_{\mathbf{k},\mathsf{td}_{\mathbf{k}}}\left[\mathcal{A}(\mathbf{k}) \in \overline{H_{\mathbf{k}}}\right]\right| \leq \mu(\lambda)$$

Instantiation. The work of [BCM+18] presented a construction of noisy trapdoor claw-free functions from learning with errors.

2.5 Quantum Fully Homomorphic Encryption

Quantum Homomorphic Encryption schemes have the same syntax as traditional classical homomorphic encryption schemes, but are extended to support quantum operations and to allow plaintexts and ciphertexts to be quantum states. We take our definition directly from [BJ15].

Definition 10. *A quantum fully homomorphic encryption scheme is a tuple of QPT* $\mathsf{qFHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *satisfying*

- $\mathsf{qFHE.Gen}(1^\lambda)$*: outputs a a public and a secret key,* $(\mathsf{PK}, \mathsf{SK})$*, as well as a quantum state* ρ_{evk}*), which can serve as an evaluation key.*
- $\mathsf{qFHE.Enc}(\mathsf{PK}, \cdot) : L(\mathcal{M}) \to L(C)$*: takes as input a qubit* ρ *and outputs a ciphertext* σ
- $\mathsf{qFHE.Dec}(\mathsf{SK}, \cdot) : L(C) \to L(\mathcal{M})$*: takes a quantum ciphertext* σ *in correct, and outputs a qubit* ρ *in the message space* $L(\mathcal{M})$*.*
- $\mathsf{qFHE.Eval}(\mathcal{E}, \cdot) : L(\mathcal{R}_{evk} \otimes C^{\otimes n}) \to L(C^{\otimes m})$*: takes as input a quantum circuit* $\mathcal{E} : L(C^{\otimes n}) \to L(C^{\otimes m})$*, and a ciphertext in* $L(C^{\otimes n})$ *and outputs a ciphertext in* $L(C^{\otimes m})$*, possibly consuming the evaluation key* ρ_{evk} *in the proccess.*

Semantic security and compactness are defined analogously to the classical setting, and we defer to [BJ15] for a definition. We require an qFHE scheme to satisfy the following properties.

(Perfect) Correctness of classical messages. We require the following properties to hold: for every quantum circuit $\mathcal{E}$ acting on ℓ qubits, message x, every $r_1, r_2 \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

- $\Pr[x \leftarrow \mathsf{qFHE.Dec}(\mathsf{SK}, \mathsf{qFHE.Enc}(\mathsf{PK}, x)) : (\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{qFHE.Gen}(1^\lambda)] = 1$
- $\Pr[\mathsf{qFHE.Dec}(\mathsf{SK}, \mathsf{qFHE.Eval}(\mathsf{PK}, \mathsf{CT}))] \geq 1 - \mathsf{negl}(\lambda)$, for some negligible function negl, where: (1) $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{qFHE.Setup}(1^\lambda; r_1)$ and, (2) $\sigma \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}, x; r_2)$. The probability is defined over the randomness of the evaluation procedure.

Instantiation. The works of [Mah18a, Bra18b] give lattice-based candidates for quantum fully homomorphic encryption schemes; we currently do not know how to base this on learning with errors alone[6]. There are two desirable propertiess required from the quantum FHE schemes and the works of [Mah18a, Bra18b] satisfy both of them. We formalize them in the lemma below.

[6] Brakerski [Bra18b] remarks that the security of their candidate can be based on a circular security assumption that is also used to argue the security of existing constructions of unbounded depth multi-key FHE [CM15, MW16, PS16, BP16].

Lemma 11 ([Mah18a, Bra18b])**.** *There is a quantum fully homomorphic encryption scheme that satisfies: (1) perfect correctness of classical messages and, (2) ciphertexts of classical poly-sized messages have a poly-sized classical description.*

2.6 Quantum-Secure Function Evaluation

As a building block in our construction, we consider a secure function evaluation protocol [GHV10] for classical functionalities. A secure function evaluation protocol is a two message two party secure computation protocol; we designate the parties as sender and receiver (who receives the output of the protocol). Unlike prior works, we require the secure function evaluation protocol to be secure against polynomial time quantum adversaries.

Security. We require malicious (indistinguishability) security against a quantum adversary R and semantic security against a quantum adversary S. We define both of them below.

First, we define an indistinguishability security notion against malicious R. To do that, we employ an extraction mechanism to extract R's input x_1^*. We then argue that R should not be able to distinguish whether S uses x_2^0 or x_2^1 in the protocol as long as $f(x_1^*, x_2^0) = f(x_1^*, x_2^1)$. We don't place any requirements on the computational complexity of the extraction mechanism.

Definition 12 (Indistinguishability Security: Malicious Quantum R)**.** *Consider a secure function evaluation protocol for a functionality f between a sender S and a receiver R. We say that the secure evaluation protocol satisfies* **indistinguishability security against malicious** R^* *if for every adversarial QPT R^*, there is an extractor Ext (not necessarily efficient) such the following holds. Consider the following experiment:*

$\underline{\mathsf{Expt}(1^\lambda, b)}$*:*

- R^* *outputs the first message* msg_1.
- *Extractor* Ext *on input* msg_1 *outputs* x_1^*.
- *Let x_2^0, x_2^1 be two inputs such that $f(x_1^*, x_2^0) = f(x_1^*, x_2^1)$. Party S on input msg_1 and x_2^b, outputs the second message* msg_2.
- R^* *upon receiving the second message outputs a bit* $\mathfrak{out}$.
- *Output* $\mathfrak{out}$.

We require that,

$$\left|\Pr[1 \leftarrow \mathsf{Expt}(1^\lambda, 0)] - \Pr[1 \leftarrow \mathsf{Expt}(1^\lambda, 1)]\right| \leq \mathsf{negl}(\lambda),$$

for some negligible function negl.

We now define semantic security against S. We insist that S should not be able to distinguish which input S used to compute its messages. Note that S does not get to see the output recovered by the receiver.

Definition 13 (Semantic Security against Quantum S^*)**.** *Consider a secure function evaluation protocol for a functionality f between a sender S and a receiver R where R gets the output. We say that the secure function evaluation protocol satisfies* **semantic security against** S^* *if for every adversarial QPT S^*, the following holds: Consider two strings x_1^0 and x_2^1. Denote by $\mathcal{D}_b$ the distribution of the first message (sent to S^*) generated using x_1^b as R's input. The distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ are computationally indistinguishable.*

Instantiation. A secure function evaluation protocol can be built from garbled circuits and oblivious transfer that satisfies indistinguishability security against malicious receivers. Garbled circuits can be based on the hardness of learning with errors by suitably instantiating the symmetric encryption in the construction of Yao's garbled circuits [Yao86] with one based on the hardness of learning with errors [Reg09]. Oblivious transfer with indistinguishability security against malicious receivers based on learning with errors was presented in a recent work of Brakerski et al. [BD18]. Thus, we have the following lemma.

Lemma 14 ([Yao86, Reg09, BD18])**.** *Assuming the quantum hardness of learning with errors, there exists a quantum-secure function evaluation protocol for polynomial time classical functionalities.*

2.7 Lockable Obfuscation

We first recall the definition of circuit obfuscation schemes [BGI$^+$01]. A circuit obfuscation scheme associated with the class of circuits $\mathcal{C}$ consists of the classical PPT algorithms $(\mathsf{Obf}, \mathsf{ObfEval})$ defined below:

- **Obfuscation,** $\mathsf{Obf}(1^\lambda, C)$: it takes as input the security parameter λ, circuit C and produces an obfuscated circuit $\widetilde{\mathbf{C}}$.
- **Evaluation,** $\mathsf{ObfEval}(\widetilde{\mathbf{C}}, x)$: it takes as input the obfuscated circuit $\widetilde{\mathbf{C}}$, input x and outputs y.

Perfect Correctness. A program obfuscation scheme $(\mathsf{Obf}, \mathsf{ObfEval})$ is said to be correct if for every circuit $C \in \mathcal{C}$ with $C : \{0,1\}^{\ell_{in}} \to \{0,1\}^{\ell_{out}}$, for every input $x \in \{0,1\}^{\ell_{in}}$, we have $\widetilde{\mathbf{C}}(x) = C(x)$.

We are interested in program obfuscation schemes that are (i) defined for a special class of circuits called compute-and-compare circuits and, (ii) satisfy distributional virtual black box security notion [BGI$^+$01]. Such obfuscation schemes were first introduced by [WZ17, GKW17] and are called lockable obfuscation schemes. We recall their definition, adapted to quantum security, below.

Definition 15 (Quantum-Secure Lockable Obfuscation)**.** *An obfuscation scheme* $(\mathsf{Obf}, \mathsf{ObfEval})$ *for a class of circuits $\mathcal{C}$ is said to be a* ***quantum-secure lockable obfuscation scheme*** *if the following properties are satisfied:*

- *It satisfies the above mentioned correctness property.*
- **Compute-and-compare circuits***: Each circuit $\mathbf{C}$ in $\mathcal{C}$ is parameterized by strings $\alpha \in \{0,1\}^{\mathrm{poly}(\lambda)}$, $\beta \in \{0,1\}^{\mathrm{poly}(\lambda)}$ and a poly-sized circuit C such that on every input x, $\mathbf{C}(x)$ outputs β if and only if $C(x) = \alpha$.*

- **Security**: *For every polynomial-sized circuit C, string $\beta \in \{0,1\}^{\mathrm{poly}(\lambda)}$,for every QPT adversary $\mathcal{A}$ there exists a QPT simulator* Sim *such that the following holds: sample $\alpha \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$,*

  $$\left\{\mathsf{Obf}\left(1^{\lambda}, \mathbf{C}\right)\right\} \approx_{Q,\varepsilon} \left\{\mathsf{Sim}\left(1^{\lambda}, 1^{|C|}\right)\right\},$$

  where $\mathbf{C}$ *is a circuit parameterized by C, α, β with $\varepsilon \leq \frac{1}{2^{|\alpha|}}$.*

Instantiation. The works of [WZ17, GKW17, GKVW19] construct a lockable obfuscation scheme based on polynomial-security of learning with errors (see Section 2.1). Since learning with errors is conjectured to be hard against QPT algorithms, the obfuscation schemes of [WZ17, GKW17, GKVW19] are also secure against QPT algorithms. Thus, we have the following theorem.

Theorem 16 ([GKW17, WZ17, GKVW19])**.** *Assuming quantum hardness of learning with errors, there exists a quantum-secure lockable obfuscation scheme.*

3 Secure Quantum Extraction Protocols

We define the notion of quantum extraction protocols below. An extraction protocol, associated with an NP relation, is a *classical* interactive protocol between a sender and a receiver. The sender has an NP instance and a witness; the receiver only has the NP instance. Roughly speaking, we require the property that there is a QPT extractor that can extract the witness from a semi-malicious sender (i.e., follows the protocol but is allowed to choose its own randomness) even if the sender is a QPT algorithm. In addition, we require the following property (zero-knowledge): the interaction of any malicious receiver with the sender should be simulatable without the knowledge of the witness. The malicious receiver can either be classical or quantum and thus, we have two notions of quantum extraction protocols corresponding to both of these cases.

In terms of properties required, this notion closely resembles the concept of zero-knowledge argument of knowledge (ZKAoK) systems. There are two important differences:

- Firstly, we do not require any completeness requirement from the underlying interactive protocol.
- In ZKAoK systems, the prover can behave maliciously (i.e., deviates from the protocol) and the argument of knowledge property states that the probability with which the extractor can extract is negligibly close to the probability with which the prover can convince the verifier. In our definition, there is no guarantee of extraction if the sender behaves maliciously.

Definition 17 (Quantum extraction protocols secure against quantum adversaries)**.** *A **quantum extraction protocol secure against quantum adversaries**, denoted by* qQEXT *is a classical protocol between two parties, sender* S *and a receiver* R *and is associated with an NP relation $\mathcal{R}$. The input to both the parties is an instance $\mathbf{z} \in \mathcal{L}(\mathcal{R})$. In addition, the sender also gets as input the witness $\mathbf{w}$ such that $(\mathbf{z}, \mathbf{w}) \in \mathcal{R}$. At the end of the protocol, the receiver gets the output $\mathbf{w}'$. The following properties are satisfied by* qQEXT*:*

- **Quantum Zero-Knowledge**: *Let $p : \mathbb{N} \to \mathbb{N}$ be any polynomially bounded function. For every $(\mathbf{z}, \mathbf{w}) \in \mathcal{R}$, for any QPT algorithm R^* with private quantum register of size $|\mathrm{R}_{\mathsf{R}^*}| = p(\lambda)$, for any large enough security parameter $\lambda \in \mathbb{N}$, there exists a QPT simulator Sim such that*

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) \approx_Q \mathsf{Sim}(1^\lambda, \mathsf{R}^*, \mathbf{z}, \cdot).$$

- **Semi-Malicious Extractability**: *Let $p : \mathbb{N} \to \mathbb{N}$ be any polynomially bounded function. For any large enough security parameter $\lambda \in \mathbb{N}$, for every $(\mathbf{z}, \mathbf{w}) \in \mathcal{L}(\mathcal{R})$, for every QPT S^* with private quantum register of size $|\mathrm{R}_{\mathsf{S}^*}| = p(\lambda)$, there exists a QPT extractor $\mathsf{Ext} = (\mathsf{Ext}_1, \mathsf{Ext}_2)$ (possibly using the code of S^* in a non black box manner), the following holds:*
  - $\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{R}(1^\lambda, \mathbf{z})\rangle\right) \approx_Q \mathsf{Ext}_1\left(1^\lambda, \mathsf{S}^*, \mathbf{z}, \cdot\right)$
  - *If S^* is semi-malicious*[7] *then the probability that Ext_2 outputs $\mathbf{w}'$ such that $(\mathbf{z}, \mathbf{w}') \in \mathcal{R}$ is negligibly close to 1.*

Definition 18 (Quantum extraction protocols secure against classical adversaries)**.** *A **quantum extraction protocol secure against classical adversaries** cQEXT is defined the same way as in Definition 17 except that instead of quantum zero-knowledge, cQEXT satisfies classical zero-knowledge defined below:*

- **Classical Zero-Knowledge**: *Let $p : \mathbb{N} \to \mathbb{N}$ be any polynomially bounded function. For any large enough security parameter $\lambda \in \mathbb{N}$, for every $(\mathbf{z}, \mathbf{w}) \in \mathcal{R}$, for any classical PPT algorithm R^* with auxiliary information $\mathsf{aux} \in \{0,1\}^{\mathrm{poly}(\lambda)}$, there exists a classical PPT simulator Sim such that*

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \mathsf{aux})\rangle\right) \approx_c \mathsf{Sim}(1^\lambda, \mathsf{R}^*, \mathbf{z}, \mathsf{aux}).$$

Quantum-Lasting Security. A desirable property of cQEXT protocols is that a classical malicious receiver, long after the protocol has been executed cannot use a quantum computer to learn the witness of the sender from the transcript of the protocol and its own private state. We call this property *quantum-lasting security*; first introduced by Unruh [Unr13]. We formally define quantum-lasting security below.

Definition 19 (Quantum-Lasting Security)**.** *A cQEXT protocol is said to be* **quantum-lasting secure** *if the following holds: for any large enough security parameter $\lambda \in \mathbb{N}$, for any semi-malicious classical PPT R^*, for any QPT adversary $\mathcal{A}^*$, for any auxiliary information $\mathsf{aux} \in \{0,1\}^{\mathrm{poly}(\lambda)}$, for any auxiliary state of polynomially many qubits, ρ, there exist a QPT simulator Sim^* such that:*

$$\mathcal{A}^*\left(\mathsf{View}_{\mathsf{R}^*}\left\langle \mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \mathsf{aux})\right\rangle, \rho\right) \approx_Q \mathsf{Sim}^*(1^\lambda, \mathbf{z}, \mathsf{aux}, \rho)$$

4 QEXT Secure Against Classical Adversaries

In this section, we show how to construct quantum extraction protocols secure against classical adversaries. While such protocols can be constructed from assumptions such as decisional Diffie-

[7] A QPT algorithm is said to be semi-malicious in the quantum extraction protocol if it follows the protocol but is allowed to choose its own randomness.

Hellman (DDH), factoring etc., i.e., cryptographic assumptions that are broken by polynomial-time quantum attacks[8] and additionally post-quantum secure cryptographic assumptions such as learning with errors (needed to argue quantum-lasting security), we show that rather surprisingly such protocols can be based solely on learning with errors.

Tools.

- Quantum-secure computationally-hiding and perfectly-binding non-interactive commitments, Comm (see Section 2.3).
- Noisy trapdoor claw-free functions $\{f_{\mathbf{k},b} : \mathcal{X} \to D_{\mathcal{Y}}\}_{\mathbf{k}\in\mathcal{K},b\in\{0,1\}}$ (see Section 2.4).
- Quantum-secure secure function evaluation protocol $\mathsf{SFE} = (\mathsf{SFE.S}, \mathsf{SFE.R})$ (see Section 2.6).

Construction. We present the construction of the quantum extraction protocol (S, R) in Figure 1 for an NP language $\mathcal{L}$. Our extraction mechanism of committing to the shares of the values and later releasing only one of the shares is borrowed from [PW09].

Lemma 20. *Assuming the quantum security of* Comm, SFE *and NTCFs, the protocol* (S, R) *is a quantum extraction protocol secure against classical adversaries for NP.*

Proof.

Classical Zero-Knowledge. Let R^* be a classical PPT algorithm. We first describe a classical simulator Sim such that R^* cannot distinguish whether its interacting with S or with Sim.

Description of Sim**.**

- Until the SFE round, it behaves as the honest sender would,
  - For every $i \in [k]$, it computes $(\mathbf{k}_i, \mathsf{td}_i) \leftarrow \mathsf{Gen}(1^\lambda; r_i)$. Send $\left(\{\mathbf{k}_i\}_{i\in[k]}\right)$.
  - It receives $\{y_i\}_{i\in[k]}$ from R^*.
  - It sends bits $(v_1, \ldots, v_k)$, where $v_i \xleftarrow{\$} \{0,1\}$ for $i \in [k]$.
  - It receives $\left(\left\{\mathbf{c}_{i,0}^{(j)}, \mathbf{c}_{i,1}^{(j)}\right\}_{i,j\in[k]}\right)$ from R^*.
  - For every $i, j \in [k]$, it sends random bits $w_i^{(j)} \in \{0,1\}$.
  - It receives $\left(\left\{(sh_{i,w_i}^{(j)})', (\mathbf{d}_{i,w_i}^{(j)})'\right\}_{i,j\in[k]}\right)$ from R^*.
- It then executes SFE with R^*, associated with the two-party functionality $\mathbf{F}$ defined in Figure 1; the input of Sim in SFE is $\perp$.

[8] An example of such a protocol is the sender encrypts the witness using an encryption scheme, based on DDH, factoring etc.

F

Input of sender: $\left(\left\{\mathbf{c}_{i,0}^{(j)}, \mathbf{c}_{i,1}^{(j)}, (sh_{i,w_i}^{(j)})', (\mathbf{d}_{i,w_i}^{(j)})', \mathsf{td}_i, \mathbf{k}_i, y_i, v_i, w_i^{(J)}\right\}_{i,j\in[k]}, \mathbf{w}\right)$
Input of receiver: $\left(\left\{sh_{i,\overline{w_i}}^{(j)}, \mathbf{d}_{i,\overline{w_i}}^{(j)}\right\}_{i,j\in[k]}\right)$

- If for any $i, j \in [k]$, $\mathbf{c}_{i,w_i}^{(j)} \neq \mathsf{Comm}\left(1^\lambda, (sh_{i,w_i}^{(j)})'; (\mathbf{d}_{i,w_i}^{(j)})'\right)$ or $\mathbf{c}_{i,\overline{w_i}}^{(j)} \neq \mathsf{Comm}\left(1^\lambda, sh_{i,\overline{w_i}}^{(j)}; \mathbf{d}_{i,\overline{w_i}}^{(j)}\right)$, output $\perp$.
- For every $i \in [k]$, let $(x_{i,0}, x_{i,1}) \leftarrow \mathsf{Inv}(\mathbf{k}_i, \mathsf{td}_i, y_i)$.
  - Output $\perp$ if the following does not hold: for every $j, j' \in [k]$, we have $(sh_{i,w_i}^{(j)})' \oplus sh_{i,w_i}^{(j)} = (sh_{i,w_i}^{(j')})' \oplus sh_{i,w_i}^{(j')}$.
  - If $v_i = 0$: let $(b_i, J(x'_{i,b_i})) = (sh_{i,w_i}^{(j)})' \oplus sh_{i,\overline{w_i}}^{(j)}$, where $J(\cdot)$ is the injection in the definition of NTCF. If $x'_{i,b_i} \neq x_{i,b_i}$, output $\perp$.
  - If $v_i = 1$: let $(u_i, d_i) = (sh_{i,w_i}^{(j)})' \oplus sh_{i,\overline{w_i}}^{(j)}$. If $\langle d_i, J(x_{i,0}) \oplus J(x_{i,1})\rangle \neq u_i$, or if $d_i \notin G_{\mathbf{k}_i,0,x_{i,0}} \cap G_{\mathbf{k}_i,1,x_{i,1}}$ output $\perp$.
- Otherwise, output $\mathbf{w}$.

Figure 1: Description of the function **F** associated with the SFE.

We prove the following by a sequence of hybrids. For some arbitrary auxiliary information $\mathsf{aux} \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \mathsf{aux})\rangle\right) \approx_c \mathsf{Sim}(1^\lambda, \mathsf{R}^*, \mathbf{z}, \mathsf{aux}),$$

$\underline{\mathsf{Hyb}_1}$: The output of this hybrid is $\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \mathsf{aux})\rangle\right)$.

$\underline{\mathsf{Hyb}_2}$: Consider the following sender, $\mathsf{Hyb}_2.\mathsf{S}$, that behaves as follows:

1. R^*: Sends $\{y_i\}_{i\in[k]}$.
2. $\mathsf{Hyb}_2.\mathsf{S}$: Sends $(v_1, \ldots, v_k)$ uniformly at random. If R^* aborts in this step, $\mathsf{Hyb}_2.\mathsf{S}$ aborts.
3. R^*: Sends $\left\{\left(\mathbf{c}_{i,0}^{(j)}, \mathbf{c}_{i,1}^{(j)}\right)\right\}_{i,j\in[k]}$. If R^* aborts in this step, $\mathsf{Hyb}_2.\mathsf{S}$ aborts.
4. $\mathsf{Hyb}_2.\mathsf{S}$: Sends $w_i^{(j)} \in \{0,1\}$ uniformly at random for all $i, j \in [k]$.

Input of sender: $(\mathbf{z}, \mathbf{w})$.
Input of receiver: $\mathbf{z}$

- S: Compute $\forall i \in [k], (\mathbf{k}_i, \mathsf{td}_i) \leftarrow \mathsf{Gen}(1^\lambda; r_i)$, where $k = \lambda$. Send $\left(\{\mathbf{k}_i\}_{i \in [k]}\right)$.
- R: For every $i \in [k]$, choose a random bit $b_i \in \{0,1\}$ and sample a random $y_i \leftarrow f'_{\mathbf{k}_i, b_i}(x_{i,b_i})$, where $x_{i,b_i} \xleftarrow{\$} \mathcal{X}$. Send $\{y_i\}_{i \in [k]}$. (Recall that $f'_{\mathbf{k},b}(x)$ is a distribution over $\mathcal{Y}$.)
- S: Send bits $(v_1, \ldots, v_k)$, where $v_i \xleftarrow{\$} \{0,1\}$ for $i \in [k]$.
- R: For every $i, j \in [k]$, compute $\mathbf{c}_{i,0}^{(j)} \leftarrow \mathsf{Comm}(1^\lambda, sh_{i,0}^{(j)}; \mathbf{d}_{i,0}^{(j)})$ and $\mathbf{c}_{i,1}^{(j)} \leftarrow \mathsf{Comm}(1^\lambda, sh_{i,1}^{(j)}; \mathbf{d}_{i,1}^{(j)})$, where $sh_{i,0}^{(j)}, sh_{i,1}^{(j)} \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$ for $i, j \in [k]$. Send $\left(\left\{\mathbf{c}_{i,0}^{(j)}, \mathbf{c}_{i,1}^{(j)}\right\}_{i,j \in [k]}\right)$.
- S: For every $i, j \in [k]$, send random bits $w_i^{(j)} \in \{0,1\}$.
- R: Send $\left(\left\{(sh_{i,w_i}^{(j)})', (\mathbf{d}_{i,w_i}^{(j)})'\right\}_{i,j \in [k]}\right)$.
- S and R run SFE, associated with the two-party functionality $\mathbf{F}$ defined in Figure 1; S takes the role of $\mathsf{SFE.S}$ and R takes the role of $\mathsf{SFE.R}$. The input to $\mathsf{SFE.S}$ is $\left(\left\{\mathbf{c}_{i,0}^{(j)}, \mathbf{c}_{i,1}^{(j)}, (sh_{i,w_i}^{(j)})', (\mathbf{d}_{i,w_i}^{(j)})', \mathsf{td}_i, \mathbf{k}_i, y_i, v_i, w_i^{(j)}\right\}_{i,j \in [k]}, \mathbf{w}\right)$ and the input to $\mathsf{SFE.R}$ is $\left(\left\{sh_{i,\overline{w_i}}^{(j)}, \mathbf{d}_{i,\overline{w_i}}^{(j)}\right\}_{i,j \in [k]}\right)$.

Figure 2: Quantum Extraction Protocol (S, R)

5. R^*: Opens up the commitments queried, $\left\{\left(sh_{i,w_i}^{(j)}, \mathbf{d}_{i,w_i}^{(j)}\right)\right\}_{i,j \in [k]}$. If R^* aborts in this step, $\mathsf{Hyb}_2.\mathsf{S}$ aborts. If $\mathbf{c}_{i,w_i}^{(j)} \neq \mathsf{Comm}(1^\lambda, sh_{i,w_i}^{(j)}; \mathbf{d}_{i,w_i}^{(j)})$ for any $i, j \in [k]$, continue the execution of the protocol as in Step 11.
6. $\mathsf{Hyb}_2.\mathsf{S}$: Keep rewinding (poly($k$) times) to Step 4, until it is able to recover another commitment accepting transcript. A commitment accepting transcript is one for which all the commitments opened in Step 5 are valid, i.e. that $\mathbf{c}_{i,w_i}^{(j)} = \mathsf{Comm}(1^\lambda, sh_{i,w_i}^{(j)}; \mathbf{d}_{i,w_i}^{(j)})$. Let $\{(w_i^{(j)})'\}$ be the queries sent in the second recovered commitment accepting transcript. If for any $i \in [k]$, it is the case that for every $j \in [k]$, it holds that $(w_i^{(j)})' = w_i^{(j)}$, then abort.
7. If $\mathsf{Hyb}_2.\mathsf{S}$ did not abort in the previous step, then for every $i \in [k]$, there is $j_i \in [k]$, s.t. $(w_i^{(j_i)})' \neq w_i^{(j_i)}$. From these two transcripts, it extracts the committed value.

8. $\mathsf{Hyb}_2.\mathsf{S}$: (We call this step the NTCF condition check). From the commited values recovered, check if they satisfy the desired NTCF conditions. I.e. for every $i \in [k]$, if $v_i = 0$, check if the decommited value if a valid preimage $(b_i, J(x_{i,b_i}))$, and if $v_i = 1$ check if the decommited value is a valid correlation (u_i, d_i). If the check do not pass, continue as before. If the check pass,

   - Keep rewinding (poly(k) times) until Step 2, repeating the proccess above, including the rewinding phase for the commitment challenges. The rewinding continues until we get another transcript, for which the NTCF check passes. Let $(v'_1, \ldots, v'_k)$ be the messages sent at Step 2 in the new transcript.

9. $\mathsf{Hyb}_2.\mathsf{S}$: If $(v_1, \ldots, v_k)$ and $(v'_1, \ldots, v'_k)$ are different in less than $\omega(\log(k))$ coordinates, then abort.

10. If $\mathsf{Hyb}_2.\mathsf{S}$ has not aborted so far, let S be the set of indices at which both $(v_1, \ldots, v_k)$ and $(v'_1, \ldots, v'_k)$ differ. For $i \in S$, let (b_i, x_i) and (d_i, u_i) be the values recovered from the commitment accepting transcripts associated with bits v_i and v'_i. Denote $T = \{(b_i, x_i, d_i, u_i) : i \in S\}$. Moreover, $|T| = \omega(\log(k))$

11. Now, continue the execution of the protocol on the original thread; i.e., when the $\mathsf{Hyb}_2.\mathsf{S}$ queries $(w_1, \ldots, w_k)$ and $(v_1, \ldots, v_k)$.

The only difference between Hyb_1 and Hyb_2 is that $\mathsf{Hyb}_2.\mathsf{S}$ aborts on some transcripts; conditioned on $\mathsf{Hyb}_2.\mathsf{S}$ not aborting, the transcript produced by the receiver when interacting with S is identical to the transcript produced by $\mathsf{Hyb}_2.\mathsf{S}$. We claim that the probability that $\mathsf{Hyb}_2.\mathsf{S}$ aborts, conditioned on the event that R^* does not abort, is negligibly small.

Claim 21. $\Pr[\mathsf{Hyb}_2.\mathsf{S}\ \textit{aborts} | \mathsf{R}^*\ \textit{does not abort}] = \mathsf{negl}(k)$

Proof. To argue this, we first establish some terminology. Let p_1 be the probability with which R^* produces a commitment accepting transcript and p_2 be the probability with which R^* passes the NTCF condition check. We call the rewinding performed in Step 4 to be "inner rewinding" and the the rewinding performed in Step 8 to be "outer rewinding".

In the rest of the proof, we condition on the event that R^* does not abort. Consider the following claims.

Claim 22. *The probability that the number of outer rewinding operations performed is greater than k is negligible.*

Proof. Note that the outer rewinding is performed till the point it can recover a transcript that passes the NTCF check. Since the probability that R^* produces a transcript that passes the NTCF check is p_2, we have that the expected number of outer rewinding operations to be $(1 - p_2) + p_2 \cdot \frac{1}{p_2} \leq 2$. By Chernoff, the probability that the number of outer rewinding operations is greater than k is negligible. □

Claim 23. *The probability that the number of inner rewinding operations performed is greater than k^2 is negligible.*

Proof. Note that for every NTCF transcript, Comm is rewound many times until $\mathsf{Hyb}_2.\mathsf{S}$ can indeed recover another commitment-accepting transcript. For a given NTCF transcript, since the probability that R^* produces a commitment accepting transcript is p_1, we have that the expected number of inner rewinding operations to be $(1-p_1)+p_1 \cdot \frac{1}{p_1} \leq 2$. And thus by Chernoff, for a given NTCF transcript, the probability that the number of inner rewinding operations is greater than k is negligible. Since the number NTCF transcripts produced is at most k with probability negligibly close to 1, we have that the total number of inner rewinding operations is at most k^2 with probability neglibly close to 1. □

We now argue about the probability that $\mathsf{Hyb}_2.\mathsf{S}$ aborts on an NTCF transcript (Step 9) and the probability that it aborts on the transcript of Comm (Step 6).

Claim 24. *The probability that* $\mathsf{Hyb}_2.\mathsf{S}$ *aborts in Step 9 is negligible.*

Proof. Note that $\mathsf{Hyb}_2.\mathsf{S}$ aborts in Step 9 only if: (i) it received a valid transcript on the original thread of execution, (ii) it rewinds until the point it receives another valid NTCF transcript and, (iii) the challenge $(v'_1,\ldots,v'_k)$ on which the second transcript was accepted differs from $(v_1,\ldots,v_k)$ only in $\omega(\log(k))$ co-ordinates. Thus, the probability that it aborts is the following quantity:

$$\begin{aligned} & p_2(p_2+p_2(1-p_2)+p_2(1-p_2)^2+\cdots)\cdot \mathsf{Pr}[{}^{(v_1,\ldots,v_k) \text{ and } (v'_1,\ldots,v'_k)}_{\text{differ in less than } \omega(\log(k)) \text{ co-ordinates}}] \\ \leq\ & p_2^2\left(\frac{1}{p_2}\right)\cdot \mathsf{Pr}[{}^{(v_1,\ldots,v_k) \text{ and } (v'_1,\ldots,v'_k)}_{\text{differ in less than } \omega(\log(k)) \text{ co-ordinates}}] \\ =\ & p_2 \cdot \mathsf{negl}(k) \ \ \text{(By Chernoff Bound)} \end{aligned}$$

□

Claim 25. *The probability that* $\mathsf{Hyb}_2.\mathsf{S}$ *aborts in Step 6 is negligible.*

Proof. Since step 6 is executed for multiple NTCF transcripts, we need to argue that for any of NTCF transcripts, the probability that $\mathsf{Hyb}_2.\mathsf{S}$ aborts in Step 6 is negligible. Since we already argued in Claim 23 that the number of inner rewinding operations is poly(k), by union bound, it suffices to argue the probability that for any given NTCF transcript, the probability that $\mathsf{Hyb}_2.\mathsf{S}$ aborts in Step 6 is negligible. This is similar to the argument in Claim 24: the probability that $\mathsf{Hyb}_2.\mathsf{S}$ aborts in Step 6 is $p_1^2 \cdot \frac{1}{p_1} \cdot \mathsf{Pr}\left[\exists i \in [k], \forall j \in [k] : \left(w_i^{(j)}\right)' = \left(w_i^{(j)}\right)\right] = p_1 \cdot 2^{-k}$. □

Observe that $\mathsf{Hyb}_2.\mathsf{S}$ only aborts in Steps 6 and 9; recall that we have already conditioned on the even that R^* does not abort. Thus, we have the proof of the claim.

□

This claim shows that Hyb_1 and Hyb_2 are indistinguishable:

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{S}(1^\lambda,\mathbf{z},\mathbf{w}),\mathsf{R}^*(1^\lambda,\mathbf{z},\mathsf{aux})\rangle\right) \approx_c \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_2.\mathsf{S}(1^\lambda,\mathbf{z},\mathbf{w}),\mathsf{R}^*(1^\lambda,\mathbf{z},\mathsf{aux})\rangle\right).$$

Hyb_3: In this hybrid, $\mathsf{Hyb}_3.\mathsf{S}$ will do as $\mathsf{Hyb}_2.\mathsf{S}$ except as follows: once it gets to step 8, if the NTCF check passes, it continues as usual, but if the NTCF check does not pass, it inputs $\perp$ in the SFE.

The indistinguishability of Hyb_3 and Hyb_4 follows from the security of the SFE against malicious receivers, and we have:

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_2.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \mathsf{aux})\rangle\right) \approx_c \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_3.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \mathsf{aux})\rangle\right),$$

This is because the following holds in the event that the above check does not pass:

$$\mathbf{F}\left(\left(\left\{\mathbf{c}^{(j)}_{i,0}, \mathbf{c}^{(j)}_{i,1}, (sh^{(j)}_{i,w_i})', (\mathbf{d}^{(j)}_{i,w_i})', \mathsf{td}_i, \mathbf{k}_i, y_i, v_i, w^{(j)}_i\right\}_{i,j\in[k]}, \mathbf{w}\right), \left(\left\{sh^{(j)}_{i,\overline{w_i}}, \mathbf{d}^{(j)}_{i,\overline{w_i}}\right\}_{i,j\in[k]}\right)\right) = \mathbf{F}\left((\perp), \left(\left\{sh^{(j)}_{i,\overline{w_i}}, \mathbf{d}^{(j)}_{i,\overline{w_i}}\right\}_{i,j\in[k]}\right)\right).$$

Hyb_4: In this hybrid, $\mathsf{Hyb}_4.\mathsf{S}$ always inputs $\perp$ in the SFE.

We have the following:

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_3.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \mathsf{aux})\rangle\right) \approx_c \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_4.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \mathsf{aux})\rangle\right)$$

This is because either $\mathsf{Hyb}_3.\mathsf{S}$ inputs $\perp$ into the SFE or it can find $T = \{(b_i, x_i, u_i, d_i) : i \in S\}$ (see Hyb_2) such that both (b_i, x_i) and (u_i, d_i) pass the NTCF checks corresponding to the i^{th} instantiation. Moreover, recall that $|T| = \omega(\log(k))$. This contradicts the security of NTCFs: by the adaptive hardcore bit property of the NTCF, a PPT classical adversary can break a given instantiation with probability negligibly close to $1/2$ and thus, it can break $\omega(\log(k))$ instantiations only with negligible probability.

Hyb_5: Now the hybrid sender, $\mathsf{Hyb}_5.\mathsf{S}$ does as $\mathsf{Hyb}_4.\mathsf{S}$, but it does not rewind R^*.

The statistical distance between Hyb_4 and Hyb_5 is negligible in k; this follows from Claim 21.

Extractability. Let S^* be the semi-malicious sender. We define our quantum extractor Ext as follows.

Description of Ext. The input to Ext is the instance $\mathbf{z}$.

- Run S^* to obtain $\{\mathbf{k}_i\}_{i\in[k]}$.
- For all $i \in [k]$,
  - Prepare the superpostion

$$\frac{1}{\sqrt{2|\mathcal{X}|}} \sum_{b,x\in\mathcal{X},y\in\mathcal{Y}} \sqrt{f'_{\mathbf{k}_i,b}(x)(y)}|b, x, y\rangle$$

which can be done efficiently by the required properties of NTCF.

  - Measure the y register, to obtain outcome y_i. Denote the postmeasurement quantum state by $|\Psi_i\rangle$. By NTCF,

$$|\Psi_i\rangle = \frac{|0, x_{i,0}\rangle + |1, x_{i,1}\rangle}{\sqrt{2}}$$

    where $(x_{i,0}, x_{i,1}) \leftarrow \mathsf{Inv}(\mathbf{k}_i, \mathsf{td}_i, y_i)$.
  - Compute J into a new register, $|b, x, 0\rangle \longrightarrow |b, x, J(x)\rangle$, and then uncompute the register containing x by performing J^{-1}, i.e. $|b, x, J(x)\rangle \longrightarrow |b, x \oplus J^{-1}(J(x)), J(x)\rangle$. The resulting transformation is $|b, x, 0\rangle \longrightarrow |b, 0, J(x)\rangle$.
  - Discard the second register, and keep the first register containing b and the third register with $J(x)$. At this point, the extractor has the states

$$|\Psi_i'\rangle = \frac{|0, J(x_{i,0})\rangle + |1, J(x_{i,1})\rangle}{\sqrt{2}}$$

- Send $\{y_i\}_{i\in[k]}$ to S^*, and let $\{v_i\}_{i\in[k]}$ be the message received from $\mathsf{S}*$.
- For all $i \in [k]$:
  - if $v_i = 0$, measure $|\Psi_i'\rangle$ in the standard basis, to obtain $(b_i, J(x_{i,b_i}))$.
  - if $v_i = 1$, apply the Hadamard transformation to $|\Psi_i'\rangle$, and measure in standard basis to obtain (u_i, d_i)
- For all $i, j \in [k]$, choose the shares $(sh_{i,0}^{(j)}, sh_{i,1}^{(j)})$ uniformly at random conditioned on either $(b_i, J(x_{i,b_i})) = sh_{i,0}^{(j)} \oplus sh_{i,1}^{(j)}$ or $(u_i, d_i) = sh_{i,0}^{(j)} \oplus sh_{i,1}^{(j)}$ if $v_i = 0$ or $v_i = 1$ respectively.
- Perform the rest of the protocol as the honest receiver would. Output the outcome of the SFE protocol.

Claim 26. *Assuming NTCFs, perfect correctness and security of* SFE, *the probability that* Ext *extracts from the semi-malicious sender ie negligibly close to* 1*.*

Proof. We first claim that with probability negligibly close to 1, the following is satisfied for every $v_i \in [k]$:

- If $v_i = 0$, let $(b_i, J(x_{i,b_i}))$ be the value obtained by measuring $|\Psi_i'\rangle$ in the standard basis. Then, $f'_{\mathbf{k}_i,b_i}(x_{i,b}) = y_i$,
- If $v_i = 1$, let (u_i, d_i) be the value obtained by applying the Hadamard transformation to $|\Psi_i'\rangle$, and measuring it in the standard basis. Then $\langle d_i, J(x_{i,0}) \oplus J(x_{i,1})\rangle = u_i$ and $d_i \notin G_{\mathbf{k}_i,0,x_{i,0}} \cap G_{\mathbf{k}_i,1,x_{i,1}}$.

This follows from the union bound and Lemma 5.1 of the protocol of [BCM+18]. By perfect correctness of SFE, it follows that if the extractor inputs shares $sh_{i,0}^{(j)}, sh_{i,1}^{(j)}$ that answer correctly each challenge, the output it will receive from the SFE will be the witness $\mathbf{w}$.

□

Claim 27. $\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{R}(1^\lambda, \mathbf{z})\rangle\right) \approx_Q \mathsf{Ext}_1\left(1^\lambda, \mathsf{S}^*, \mathbf{z}, \cdot\right)$

Proof. Consider the following hybrids.

$\underline{\mathsf{Hyb}_1}$: The output of this hybrid is $\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{R}(1^\lambda, \mathbf{z})\rangle\right)$.

$\underline{\mathsf{Hyb}_2}$: We define a hybrid receiver $\mathsf{Hyb}_2.\mathsf{R}$ who sets the input to SFE to be $\perp$.

The following holds from the semantic security of SFE against QPT senders:

$$\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{R}(1^\lambda, \mathbf{z})\rangle\right) \approx_Q \mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{Hyb}_2.\mathsf{R}(1^\lambda, \mathbf{z})\rangle\right)$$

$\underline{\mathsf{Hyb}_3}$: We define a hybrid receiver $\mathsf{Hyb}_3.\mathsf{R}$ that behaves as $\mathsf{Hyb}_2.\mathsf{R}$, but it samples $\{y_i\}_{i\in[k]}$ as the extractor would, by preparing the claw-free superpositions, and then measuring the y register. We claim that the distribution over y_i's is the same in Hyb_2 and Hyb_3. To see this, note that Hyb_3 samples from the distribution y_i from the distribution: $\frac{1}{2|X|}\sum_{b\in\{0,1\},x\in X} f'_{\mathbf{k}_i,b}(x)(y)$. To sample from this distribution, we can first sample $b \in \{0,1\}$, then an $x_{i,b} \in \mathcal{X}$ and then sampling y_i from the distribution $f'_{\mathbf{k}_i,b}(x_{i,b})$.

$\underline{\mathsf{Hyb}_4}$: We define a hybrid receiver $\mathsf{Hyb}_4.\mathsf{R}$ who computes $\{y_i\}_{i\in[k]}$ by performing the quantum operations that the extractor does, and then computes, for all $i \in [k]$, either $(b_i, J(x_{i,b_i}))$ or (u_i, d_i) according to whether $v_i = 0$ or $v_i = 1$ respectively. In other words, $\mathsf{Hyb}_4.\mathsf{R}$ compute correct answers to the test of quantumness, then it commits to appropriate shares,

$$sh_{i,0}^{(j)} \oplus sh_{i,1}^{(j)} = \begin{cases} (b_i, J(x_{i,b})) & \text{if } v_i = 0 \\ (u_i, d_i) & \text{if } v_i = 1 \end{cases}$$

$\mathsf{Hyb}_4.\mathsf{R}$ uses these shares for commitment $\mathbf{c}_{i,0}^{(j)} = \mathsf{Comm}(1^\lambda, sh_{i,0}^{(j)}; \mathbf{d}_{i,0}^{(j)})$ and $\mathbf{c}_{i,1}^{(j)} = \mathsf{Comm}(1^\lambda, sh_{i,1}^{(j)}; \mathbf{d}_{i,1}^{(j)})$ The rest of the steps are the same as $\mathsf{Hyb}_3.\mathsf{R}$.

The following holds from the computational hiding property of Comm by a similar argument to the one in [PW09]:

$$\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{Hyb}_3.\mathsf{R}(1^\lambda, \mathbf{z})\rangle\right) \approx_Q \mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{Hyb}_4.\mathsf{R}(1^\lambda, \mathbf{z})\rangle\right)$$

$\underline{\mathsf{Hyb}_5}$: We define a hybrid receiver $\mathsf{Hyb}_5.\mathsf{R}$ who sets the input in SFE to be $\left(\left\{sh_{i,\overline{w_i}}^{(j)}, \mathbf{d}_{i,\overline{w_i}}^{(j)}\right\}_{i\in[k]}\right)$, where $\{w_i\}_{i\in[k]}$ are the bit queried by S^* when asking the receiver to reveal commitments. Note that the output distribution of $\mathsf{Hyb}_5.\mathsf{R}$ is identical to that of the extractor Ext.

The following holds from the semantic security of SFE against quantum senders:

$$\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{Hyb}_4.\mathsf{R}(1^\lambda, \mathbf{z})\rangle\right) \approx_Q \mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{Hyb}_5.\mathsf{R}(1^\lambda, \mathbf{z})\rangle\right) \equiv \mathsf{Ext}_1\left(1^\lambda, \mathsf{S}^*, \mathbf{z}, \cdot\right)$$

□

□

Handling Aborting Adversaries. We observe that we can show our construction to satisfy a stronger extractability property: the semi-malicious sender cannot distinguish whether its interacting with the extractor or the honest receiver even if it is allowed to abort. If at any point in time, the sender aborts, so does the extractor and note that from the same arguments as above, the view of the sender when interacting with the honest sender will still be indistinguishable (against a quantum polynomial time adversary) from the view of the sender when interacting with the extractor. We formalize this in the claim below.

Claim 28. *The quantum extraction protocol (S, R) described in Figure 2 satisfies extractability even when the semi-malicious sender is allowed to abort at any point during the execution of the protocol.*

4.1 Application: Classical ZK arguments secure against quantum verifiers

In this section, we show how to construct a quantum zero-knowledge, classical prover, argument system for NP secure against quantum verifiers; that is, the protocol is classical, the malicious prover is also a classical adversary but the malicious verifier can be a polynomial time quantum algorithm. To formally define this notion, consider the following definition.

Definition 29 (Classical arguments for NP)**.** *A classical interactive protocol* $(\mathsf{Prover}, \mathsf{Verifier})$ *is a* ***classical ZK argument system*** *for an NP language $\mathcal{L}$, associated with an NP relation $\mathcal{L}(\mathcal{R})$, if the following holds:*

- **Completeness***: For any* $(\mathbf{z}, \mathbf{w}) \in \mathcal{L}(\mathcal{R})$, *we have that* $\Pr[\langle \mathsf{Prover}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{Verifier}(1^\lambda, \mathbf{z}) \rangle = 1] \geq 1 - \mathsf{negl}(\lambda)$, *for some negligible function* negl.
- **Soundness***: For any* $\mathbf{z} \notin \mathcal{L}$, *any PPT classical adversary* Prover^*, *and any polynomial-sized auxiliary information* aux, *we have that* $\Pr[\langle \mathsf{Prover}^*(1^\lambda, \mathbf{z}, \mathsf{aux}), \mathsf{Verifier}(1^\lambda, \mathbf{z}) \rangle = 1] \leq \mathsf{negl}(\lambda)$, *for some negligible function* negl.

We say that a classical argument system for NP is a QZK (quantum zero-knowledge) classical argument system for NP if in addition to the above properties, a classical interactive protocol satisfies zero-knowledge against malicious receivers.

Definition 30 (QZK classical argument system for NP)**.** *A classical interactive protocol* $(\mathsf{Prover}, \mathsf{Verifier})$ *is a* ***quantum zero-knowledge classical argument system*** *for a language $\mathcal{L}$, associated with an NP relation $\mathcal{L}(\mathcal{R})$ if both of the following hold.*

- $(\mathsf{Prover}, \mathsf{Verifier})$ *is a classical argument for $\mathcal{L}$ (Definition 29).*
- **Quantum Zero-Knowledge***: Let $p : \mathbb{N} \to \mathbb{N}$ be any polynomially bounded function. For any QPT* $\mathsf{Verifier}^*$ *that on instance* $\mathbf{z} \in \mathcal{L}$ *has private register of size* $|\mathsf{R}_{\mathsf{Verifier}^*}| = p(|\mathbf{z}|)$, *there exist a QPT* Sim *such that the following two collections of quantum channels are quantum computationally indistinguishable,*
  - $\{\mathsf{Sim}(\mathbf{z}, \mathsf{Verifier}^*, \cdot)\}_{\mathbf{z} \in \mathcal{L}}$

- $\{\mathsf{View}_{\mathsf{Verifier}^*}(\langle \mathsf{Prover}(\mathbf{z}, \mathsf{aux}_1), \mathsf{Verifier}^*(\mathbf{z}, \cdot)\rangle)\}_{\mathbf{z} \in \mathcal{L}}$.

  In other words, that for every $\mathbf{z} \in \mathcal{L}$, for any bounded polynomial $q : \mathbb{N} \to \mathbb{N}$, for any QPT distinguisher $\mathcal{D}$ that outputs a single bit, and any $p(|\mathbf{z}|) + q(|\mathbf{z}|)$-qubits quantum state ρ,

$$\Big| \Pr \left[\mathcal{D} \left(\mathsf{Sim}(\mathbf{z}, \mathsf{Verifier}^*, \cdot) \otimes I)(\rho) \right) = 1 \right]$$
$$- \Pr \left[\mathcal{D} \left((\mathsf{View}_{\mathsf{Verifier}^*}(\langle \mathsf{Prover}(\mathbf{z}, \mathsf{aux}_1), \mathsf{Verifier}^*(\mathbf{z}, \cdot)\rangle) \otimes I)(\rho) \right) = 1 \right] \Big| \leq \epsilon(|\mathbf{z}|)$$

Witness-Indistinguishability against quantum verifiers. We also consider witness indistinguishable (WI) argument systems for NP languages secure against quantum verifiers. We define this formally below.

Definition 31 (Quantum WI for an $\mathcal{L} \in$ NP)**.** *A classical protocol* (Prover, Verifier) *is a* ***quantum witness indistinguishable argument system*** *for an NP language $\mathcal{L}$ if both of the following hold.*

- (Prover, Verifier) *is a classical argument for $\mathcal{L}$ (Definition 29).*
- **Quantum WI***: Let $p : \mathbb{N} \to \mathbb{N}$ be any polynomially bounded function. For every $\mathbf{z} \in \mathcal{L}$, for any two valid witnesses $\mathbf{w}_1$ and $\mathbf{w}_2$, for any QPT* Verifier* *that on instance $\mathbf{z}$ has private quantum register of size $|\mathsf{R}_{\mathsf{Verifier}^*}| = p(|\mathbf{z}|)$, we require that*

$$\mathsf{View}_{\mathsf{Verifier}^*}(\langle \mathsf{Prover}(\mathbf{z}, \mathbf{w}_1), \mathsf{Verifier}^*(\mathbf{z}, \cdot)\rangle) \approx_Q \mathsf{View}_{\mathsf{Verifier}^*}(\langle \mathsf{Prover}(\mathbf{z}, \mathbf{w}_2), \mathsf{Verifier}^*(\mathbf{z}, \cdot)\rangle).$$

If (Prover, Verifier) *is a quantum proof system (sound against unbounded provers), we say that* (Prover, Verifier) *is a* ***quantum witness indistinguishable proof system*** *for $\mathcal{L}$.*

Instantiation. By suitably instantiating the constant round WI argument system of Blum [Blu86] with perfectly binding quantum computational hiding commitments, we achieve a constant round quantum WI classical argument system assuming quantum hardness of learning with errors.

4.1.1 Construction

We present a construction of constant round quantum zero-knowledge classical argument system for NP.

Tools.

- Perfectly-binding and quantum-computational hiding non-interactive commitments Comm (see Section 2.3).
- Quantum extraction protocol secure against classical adversaries cQEXT = (S, R) associated with the relation $\mathcal{R}_{\mathrm{EXT}}$ as constructed in Section 5. More generally, cQEXT could be any quantum extraction protocol secure against classical adversaries as long as it satisfies Claim 28.
- Quantum witness indistinguishable classical argument of knowledge system $\Pi_{\mathsf{WI}} = (\Pi_{\mathsf{WI}}.\mathsf{Prover}, \Pi_{\mathsf{WI}}.\mathsf{Verifier})$ for the relation $\mathcal{R}_{\mathsf{wi}}$ (Definition 31).

Instance: $\left(\mathbf{z}, \mathsf{td}, \left\{(\mathbf{c}_0^{(j)})^*, (\mathbf{c}_1^{(j)})^*\right\}_{j\in[k]}\right)$

Witness: $\left(\mathbf{w}, \left\{(sh_0^{(j)}, \mathbf{d}_0^{(j)}, sh_1^{(j)}, \mathbf{d}_1^{(j)})\right\}_{j\in[k]}\right)$

NP verification: Accept if one of the following two conditions are satisfied:

- $(\mathbf{z}, \mathbf{w}) \in \mathcal{R}$.
- If for every $j \in [k]$, it holds that

$$\left((\mathbf{c}_0^{(j)})^* = \mathsf{Comm}(1^\lambda, sh_0^{(j)}; \mathbf{d}_0^{(j)})\right) \bigwedge \left((\mathbf{c}_1^{(j)})^* = \mathsf{Comm}(1^\lambda, sh_1^{(j)}; \mathbf{d}_1^{(j)})\right) \bigwedge \left(\mathsf{td} = sh_0^{(j)} \oplus sh_1^{(j)}\right).$$

Figure 3: Relation $\mathcal{R}_{\mathsf{wi}}$ associated with Π_{WI}.

Construction. Let $\mathcal{L}$ be an NP language. We describe a classical interactive protocol $(\mathsf{Prover}, \mathsf{Verifier})$ for $\mathcal{L}$ in Figure 4.

Lemma 32. *The classical interactive protocol* $(\mathsf{Prover}, \mathsf{Verifier})$ *is a quantum zero-knowledge, classical prover, argument system for NP.*

Proof. The completeness is straightforward. We prove soundness and zero-knowledge next.

Soundness. Let Prover^* be a classical PPT algorithm. We prove that $\mathsf{Prover}^*(1^\lambda, \mathbf{z}, \mathsf{aux})$, for $\mathbf{z} \notin \mathcal{L}$ and auxiliary information aux, can convince $\mathsf{Verifier}(1^\lambda, \mathbf{z})$ with only negligible probability. Consider the following hybrids.

Hyb_1: The output of this hybrid is the view of the prover $\mathsf{View}_{\mathsf{Prover}^*}(\langle \mathsf{Prover}^*(1^\lambda, \mathbf{z}, \mathsf{aux}), \mathsf{Verifier}(1^\lambda, \mathbf{z})\rangle)$ along with the decision bit of $\mathsf{Verifier}$.

Hyb_2: We consider the following hybrid verifier $\mathsf{Hyb}_2.\mathsf{Verifier}$ which executes the trapdoor commitment phase and the trapdoor extraction phase with Prover^* honestly. It then receives $\{((\mathbf{c}_0^{(j)})^*, (\mathbf{c}_1^{(j)})^*))\}_{j\in[k]}$ from the prover. $\mathsf{Hyb}_2.\mathsf{Verifier}$ sends random bits $\{b^{(j)}\}_{j\in[k]}$ to Prover^* and it then receives $(sh_{b^{(j)}}^{(j)}, \mathbf{d}_{b^{(j)}}^{(j)})$. At this point, $\mathsf{Hyb}_2.\mathsf{Verifier}$ will rewind until it can extract td^* from the commitments; if it extracted multiple values or it didn't extract any value, set $\mathsf{td}^* = \perp$. This is done similarly to the cQEXT case and the argument from [PW09].

The output distribution of this hybrid is identical to the output distribution of Hyb_1.
The following holds:

- **Trapdoor Committment Phase**: Verifier: sample $\mathsf{td} \leftarrow \{0,1\}^{\lambda}$. Compute $\mathbf{c} \leftarrow \mathsf{Comm}(1^{\lambda}, \mathsf{td}; \mathbf{d})$, where $\mathbf{d} \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}$ is the randomness used in the commitment. Send $\mathbf{c}$ to Prover.

- **Trapdoor Extraction Phase**: Prover and Verifier run the quantum extraction protocol cQEXT with Verifier taking the role of the sender cQEXT.S and Prover taking the role of the receiver cQEXT.R. The input of cQEXT.S is $(1^{\lambda}, \mathbf{c}, \mathbf{d}; \mathbf{r}_{\mathrm{qext}})$ and the input of cQEXT.R is $\left(1^{\lambda}, \mathbf{c}\right)$, where $\mathbf{r}_{\mathrm{qext}}$ is the randomness used by the sender in cQEXT. Let the transcript generated during the execution of cQEXT be $\mathcal{T}_{\mathsf{Verifier} \to \mathsf{Prover}}$. *The trapdoor extraction phase will be used by the simulator, while proving zero-knowledge, to extract the trapdoor from the malicious verifier.*

- Let $k = \lambda$. For every $j \in [k]$, Prover sends $(\mathbf{c}_0^{(j)})^* = \mathsf{Comm}(1^{\lambda}, sh_0^{(j)}; \mathbf{d}_0^{(j)})$ and $(\mathbf{c}_1^{(j)})^* = \mathsf{Comm}(1^{\lambda}, sh_1^{(j)}; \mathbf{d}_1^{(j)})$, where $sh_0^{(j)}, sh_1^{(j)} \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$.

- For every $j \in [k]$, Verifier sends bit $b^{(j)} \xleftarrow{\$} \{0,1\}$ to Prover.

- Prover sends $(sh_{b^{(j)}}^{(j)}, \mathbf{d}_{b^{(j)}}^{(j)})$ to Verifier.

- Verifier sends $\mathbf{r}_{\mathrm{qext}}, \mathbf{d}, \mathsf{td}$ to Prover. Then Prover checks the following:
  - Let $\mathcal{T}_{\mathsf{Verifier} \to \mathsf{Prover}}$ be $(m_1^S, m_1^R, \ldots, m_{t'}^S, m_{t'}^R)$, where the message m_i^R (resp., m_i^S) is the message sent by the receiver (resp., sender) in the i^{th} round [9] and t' is the number of rounds of cQEXT. Let the message produced by $\mathsf{S}\left(1^{\lambda}, \mathbf{c}, \mathbf{d}; \mathbf{r}_{\mathrm{qext}}\right)$ in the i^{th} round be $\widetilde{m}_i^S$.
  - If for any $i \in [t']$, $\widetilde{m}_i^S \neq m_i^S$ then Prover aborts If $\mathbf{c} \neq \mathsf{Comm}(1^{\lambda}, \mathsf{td}; \mathbf{d})$ then abort.

- **Execute Quantum WI**: Prover and Verifier run Π_{WI} with Prover taking the role of Π_{WI} prover Π_{WI}.Prover and Verifier taking the role of Π_{WI} verifier Π_{WI}.Verifier. The input to Π_{WI}.Prover is the security parameter 1^{λ}, instance $\left(\mathbf{z}, \mathsf{td}, \left\{(\mathbf{c}_0^{(j)})^*, (\mathbf{c}_1^{(j)})^*\right\}_{j \in [k]}\right)$ and witness $(\mathbf{w}, \perp)$. The input to Π_{WI}.Verifier is the security parameter 1^{λ} and instance $\left(\mathbf{z}, \mathsf{td}, \left\{(\mathbf{c}_0^{(j)})^*, (\mathbf{c}_1^{(j)})^*\right\}_{j \in [k]}\right)$.

- **Decision step**: Verifier computes the decision step of Π_{WI}.Verifier.

Figure 4: (Classical Prover) Quantum Zero-Knowledge Argument Systems for NP.

$$\begin{aligned}
\mathsf{Pr}\left[1 \leftarrow \langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_2.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle\right] &= \mathsf{Pr}\left[\begin{matrix} 1\leftarrow\langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_2.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle \\ \bigwedge \\ (\mathsf{td}^*=\mathsf{td} \bigvee \mathsf{td}^*\neq\mathsf{td}) \end{matrix} \;:\; \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right] \\
&\leq \underbrace{\mathsf{Pr}\left[\begin{matrix} 1\leftarrow\langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_2.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle \\ \bigwedge \\ (\mathsf{td}^*=\mathsf{td}) \end{matrix} \;:\; \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right]}_{\varepsilon_1} \\
&\quad \underbrace{+\,\mathsf{Pr}\left[\begin{matrix} 1\leftarrow\langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_2.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle \\ \bigwedge \\ (\mathsf{td}^*\neq\mathsf{td}) \end{matrix} \;:\; \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right]}_{\varepsilon_2}
\end{aligned}$$

We prove the following claims.

Claim 33. $\varepsilon_1 \leq \mathsf{negl}(\lambda)$, *for some negligible function* negl.

Proof. Consider the following hybrids.

$\underline{\mathsf{Hyb}_3}$: We define a hybrid verifier $\mathsf{Hyb}_3.\mathsf{Verifier}$ that performs the trapdoor commitment phase honestly. In the trapdoor extraction phase, it executes $\mathsf{QEXT}_1.\mathsf{Sim}(1^\lambda)$, instead of $\mathsf{QEXT}_1.\mathsf{S}(1^\lambda,\mathbf{c},\mathbf{d})$, while interacting with Prover^*. The rest of the steps of $\mathsf{Hyb}_3.\mathsf{Verifier}$ is as defined in $\mathsf{Hyb}_2.\mathsf{Verifier}$.

Let td^* be the trapdoor extracted as before. From the zero-knowledge property of cQEXT, the following holds:

$$\varepsilon_1 \leq \mathsf{Pr}\left[\begin{matrix} 1\leftarrow\langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_3.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle \\ \bigwedge \\ (\mathsf{td}^*=\mathsf{td}) \end{matrix} \;:\; \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right] + \mathsf{negl}(\lambda) \tag{1}$$

$\underline{\mathsf{Hyb}_4}$: We define the hybrid verifier $\mathsf{Hyb}_4.\mathsf{Verifier}$ that performs the same steps as $\mathsf{Hyb}_3.\mathsf{Verifier}$ execpt that it computes $\mathbf{c}$ as $\mathsf{Comm}(1^\lambda,\mathbf{0};\mathbf{d})$ instead of $\mathsf{Comm}(1^\lambda,\mathsf{td};\mathbf{d})$, where $\mathbf{0}$ is a λ-length string of all zeroes.

Let td^* be the trapdoor extracted as before. From the quantum hiding property of Comm, the following holds:

$$\mathsf{Pr}\left[\begin{matrix} 1\leftarrow\langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_3.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle \\ \bigwedge \\ (\mathsf{td}^*=\mathsf{td}) \end{matrix} \;:\; \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right] \tag{2}$$

$$\leq \mathsf{Pr}\left[\begin{matrix} 1\leftarrow\langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_4.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle \\ \bigwedge \\ (\mathsf{td}^*=\mathsf{td}) \end{matrix} \;:\; \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right] + \mathsf{negl}(\lambda) \tag{3}$$

$\underline{\mathsf{Hyb}_5}$: We define the hybrid verifier $\mathsf{Hyb}_5.\mathsf{Verifier}$ that performs the same steps as $\mathsf{Hyb}_4.\mathsf{Verifier}$ except that it samples td *after* it completes its interaction with the Prover^*.

Note that the output distributions of Hyb_4 and Hyb_5 are identical. Moreover, the probability that $\mathsf{Hyb}_5.\mathsf{Verifier}$ accepts and $\mathsf{td}^* = \mathsf{td}$ is at most $\frac{1}{2^\lambda}$. Thus we have,

$$\begin{aligned} & \Pr\left[\begin{array}{c} 1 \leftarrow \langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_4.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle \\ \bigwedge \\ (\mathsf{td}^*=\mathsf{td}) \end{array} : \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda, \rho_{\mathsf{aux}})\right] \\ = \; & \Pr\left[\begin{array}{c} 1 \leftarrow \langle P^*(1^\lambda,\mathbf{z},\mathsf{aux}),\mathsf{Hyb}_5.\mathsf{Verifier}(1^\lambda,\mathbf{z})\rangle \\ \bigwedge \\ (\mathsf{td}^*=\mathsf{td}) \end{array} : \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda, \rho_{\mathsf{aux}})\right] \\ \leq \; & \mathsf{negl}(\lambda) \end{aligned}$$

From the above hybrids, it follows that $\varepsilon_1 \leq \mathsf{negl}(\lambda)$.

□

Claim 34. $\varepsilon_2 \leq \mathsf{negl}(\lambda)$, *for some negligible function* negl.

Proof. Since the trapdoor td^* extracted from Prover^* is not equal to td, this means that there is a $j \in [k]$ s.t. $sh_0^{(j)} \oplus sh_1^{(j)} \neq \mathsf{td}$, where $sh_0^{(j)}$ and $sh_1^{(j)}$ are the values committed to in $(\mathbf{c}_0^{(j)})^*$ and $(\mathbf{c}_1^{(j)})^*$ respectively. Moreover, from the perfect binding property of Comm, there does not exist shares $(sh_0^{(j)})', (sh_1^{(j)})'$ such that $(sh_0^{(j)})' \neq sh_0^{(j)}$ or $(sh_1^{(j)})' \neq sh_1^{(j)}$ such that the commitments of $(sh_0^{(j)})'$ and $(sh_1^{(j)})'$, for some fixed random strings, are $(\mathbf{c}_0^{(j)})^*$ and $(\mathbf{c}_1^{(j)})^*$ respectively. □

Zero-Knowledge. Let $\mathsf{Verifier}^*$ be the malicious QPT verifier. We describe the simulator Sim as follows.

- It receives $\mathbf{c}$ from $\mathsf{Verifier}^*$.
- Suppose Ext be the extractor of cQEXT associated with $\mathsf{cQEXT.S}^*$, where $\mathsf{cQEXT.S}^*$ is the adversarial sender algorithm computed by $\mathsf{Verifier}^*$. Compute $\mathsf{Ext}(1^\lambda, \mathsf{cQEXT.S}^*, \cdot)$ to obtain td^*. At any time, if $\mathsf{Verifier}^*$ aborts, Sim also aborts with the output, the current private state of $\mathsf{Verifier}^*$.
- For every $j \in [k]$, it samples $sh_0^{(j)}, sh_1^{(j)}$ uniformly at random subject to $sh_0^{(j)} \oplus sh_1^{(j)} = \mathsf{td}^*$. It then computes $(\mathbf{c}_0^{(j)})^* = \mathsf{Comm}(1^\lambda, sh_0^{(j)}; \mathbf{d}_0^{(j)})$ and $(\mathbf{c}_1^{(j)})^* = \mathsf{Comm}(1^\lambda, sh_1^{(j)}; \mathbf{d}_1^{(j)})$ and sends $((\mathbf{c}_0^{(j)})^*, (\mathbf{c}_1^{(j)})^*)$ to $\mathsf{Verifier}^*$.
- It receives bits $\{b^{(j)}\}_{j \in [k]}$ from $\mathsf{Verifier}^*$.
- It sends $(sh_{b^{(j)}}^{(j)}, \mathbf{d}_{b^{(j)}}^{(j)})$ from $\mathsf{Verifier}^*$.
- It receives $(\mathbf{r}_{\mathrm{qext}}, \mathbf{d}, \mathsf{td})$ from $\mathsf{Verifier}^*$. It then checks the following:
  - Let $\mathcal{T}_{\mathsf{Verifier} \rightarrow \mathsf{Prover}}$ be $(m_1^S, m_1^R, \ldots, m_{t'}^S, m_{t'}^R)$, where the message m_i^R (resp., m_i^S) is the message sent by the receiver (resp., sender) in the i^{th} round[10] and t' is the number of rounds of cQEXT. Let the message produced by $\mathsf{cQEXT.S}\left(1^\lambda, \mathbf{c}, \mathbf{d}; \mathbf{r}_{\mathrm{qext}}\right)$ in the i^{th} round be $\widetilde{m}_i^S$.
  - If for any $i \in [t']$, $\widetilde{m}_i^S \neq m_i^S$ then Sim aborts. If $\mathsf{td} \neq \mathsf{td}^*$ then Sim aborts.

[10] We remind the reader that in every round, only one party speaks.

- Sim executes Π_{WI} with Verifier* on input instance $\left(\mathbf{z}, \mathsf{td}, \left\{(\mathbf{c}_0^{(j)})^*, (\mathbf{c}_1^{(j)})^*\right\}_{j\in[k]}\right)$. The witness Sim uses in Π_{WI} is $\left(\perp, \left\{(sh_0^{(j)}, \mathbf{d}_0^{(j)}, sh_1^{(j)}, \mathbf{d}_1^{(j)})\right\}_{j\in[k]}\right)$. If Verifier aborts at any point in time, Sim also aborts and outputs the current state of the verifier.
- Otherwise, output the current state of the verifier.

We prove the indistinguishability of the view of the verifier when interacting with the honest prover versus the view of the verifier when interacting with the simulator. Consider the following hybrids.

Hyb$_1$: The output of this hybrid is the view of Verifier* when interacting with Prover. That is, the output of the hybrid is $\mathsf{View}_{\mathsf{Verifier}^*}\left(\langle \mathsf{Prover}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{Verifier}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right)$.

Hyb$_2$: We define a hybrid prover Hyb$_2$.Prover as follows: it first receives $\mathbf{c}$ from Verifier*. It computes $\mathsf{Ext}(1^\lambda, \mathsf{cQEXT.S}^*, \cdot)$ to obtain td*. It then sends $(\mathbf{c}_0^{(j)})^*$ and $(\mathbf{c}_1^{(j)})^*$, where $(\mathbf{c}_0^{(j)})^*$ and $(\mathbf{c}_1^{(j)})^*$ are commitments of $sh_0^{(j)}, sh_1^{(j)}$ respectively and $sh_0^{(j)}, sh_1^{(j)}$ are sampled uniformly at random. It receives b from Verifier*. It then sends $(sh_b^{(j)}, \mathbf{d}_b^{(j)})$ to Verifier*. It then receives $(\mathbf{r}_{\mathsf{qext}}, \mathbf{d}, \mathsf{td})$ from Verifier*. It then checks the following:

- Let $\mathcal{T}_{\mathsf{Verifier}\rightarrow\mathsf{Prover}}$ be $(m_1^S, m_1^R, \ldots, m_{t'}^S, m_{t'}^R)$, where the message m_i^R (resp., m_i^S) is the message sent by the receiver (resp., sender) in the i^{th} round and t' is the number of rounds of cQEXT. Let the message produced by $\mathsf{cQEXT.S}\left(1^\lambda, \mathbf{c}, \mathbf{d}; \mathbf{r}_{\mathsf{qext}}\right)$ in the i^{th} round be $\widetilde{m}_i^S$.
- If for any $i \in [t']$, $\widetilde{m}_i^S \neq m_i^S$ then Sim aborts. If $\mathsf{td} \neq \mathsf{td}^*$ then Sim aborts.

Hyb$_2$.Prover finally executes Π_{WI} with Verifier*; it still uses $\mathbf{w}$ in Π_{WI}.

The following holds from the semi-malicious extractability property of cQEXT:

$$\mathsf{View}_{\mathsf{Verifier}^*}\left(\langle \mathsf{Prover}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{Verifier}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) \approx_Q \mathsf{View}_{\mathsf{Verifier}^*}\left(\mathsf{Hyb}_2.\mathsf{Prover}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{Verifier}^*(1^\lambda, \mathbf{z}, \cdot)\right)$$

This is because either cQEXT.S* is not semi-malicious in which case, simulator aborts and hence, conditioned on the event that indeed cQEXT.S* is semi-malicious, Ext can extract the witness with probability negligibly close to 1. Moreover, at any point in time if cQEXT.S* aborts, by Claim 28, we have that the state of Verifier* output by Sim is indistinguishable from the state of Verifier* when interacting with the honest prover.

Hyb$_3$: We define a hybrid prover Hyb$_3$.Prover as follows: it behaves exactly like Hyb$_2$.Prover except that it computes the commitments $(\mathbf{c}_0^{(j)})^*$ and $(\mathbf{c}_1^{(j)})^*$ as commitments of $sh_0^{(j)}$ and $sh_1^{(j)}$, where $sh_0^{(j)} \oplus sh_1^{(j)} = \mathsf{td}$.

The following holds from the quantum-computational hiding property of Comm following the same argument as [PW09]:

$$\mathsf{View}_{\mathsf{Verifier}^*}\left(\langle \mathsf{Hyb}_2.\mathsf{Prover}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{Verifier}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) \approx_Q \mathsf{View}_{\mathsf{Verifier}^*}\left(\mathsf{Hyb}_3.\mathsf{Prover}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{Verifier}^*(1^\lambda, \mathbf{z}, \cdot)\right)$$

$\underline{\mathsf{Hyb}_4}$: We define a hybrid prover $\mathsf{Hyb}_4.\mathsf{Prover}$ as follows: it behaves exactly like $\mathsf{Hyb}_3.\mathsf{Prover}$ except that it uses the witness $(\bot, (sh_0^{(j)}, \mathbf{d}_0^{(j)}, sh_1^{(j)}, \mathbf{d}_1^{(j)}))$ in Π_{WI} instead of $(\mathbf{w}, \bot)$. Note that the description of $\mathsf{Hyb}_4.\mathsf{Prover}$ is identical to the description of Sim.

The following holds from the quantum witness indistinguishability property of Π_{WI}:

$$\begin{aligned}\mathsf{View}_{\mathsf{Verifier}^*}\left(\langle \mathsf{Hyb}_3.\mathsf{Prover}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{Verifier}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) &\approx_Q \mathsf{View}_{\mathsf{Verifier}^*}\left(\mathsf{Hyb}_4.\mathsf{Prover}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{Verifier}^*(1^\lambda, \mathbf{z}, \cdot)\right) \\ &\equiv \mathsf{Sim}(1^\lambda, \mathbf{z}, \cdot)\end{aligned}$$

□

5 QEXT Secure Against Quantum Adversaries

5.1 Construction of QEXT

We present a construction of quantum extraction protocols secure against quantum adversaries, denoted by qQEXT. First, we describe the tools used in this construction.

Tools.

- Quantum-secure computationally-hiding and perfectly-binding non-interactive commitments Comm (see Section 2.3).
- Quantum fully homomorphic encryption scheme with some desired properties, (qFHE.Gen, qFHE.Enc, qFHE.Dec, qFHE.Eval).
  - It admits homomorphic evaluation of arbitrary computations,
  - It admits perfect correctness,
  - The ciphertext of a classical message is also classical.

  We show in Section 2.5 that there are qFHE schemes satisfying the above properties.
- Quantum-secure two-party secure computation SFE with the following properties (see Section 2.6):
  - Only one party receives the output. We designate the party receiving the output as the receiver SFE.R and the other party to be SFE.S.
  - Security against quantum passive senders.
  - IND-Security against quantum malicious receivers.
- Quantum-secure lockable obfuscation **LObf** = (Obf, ObfEval) for $\mathcal{C}$, where every circuit $\mathbf{C}$, parameterized by $(r, k, \mathsf{SK}_1, \mathsf{CT}^*)$, in $\mathcal{C}$ is defined in Figure 5. Note that $\mathcal{C}$ is a compute-and-compare functionality (see Section 2.7).

C

Input: CT
Hardwired values: $\mathbf{r}, \mathbf{k}, \mathsf{SK}_1, \mathsf{CT}^*$.

- $\mathsf{SK}_2' \leftarrow \mathsf{qFHE.Dec}(\mathsf{SK}_1, \mathsf{CT})$
- $\mathbf{r}' \leftarrow \mathsf{qFHE.Dec}(\mathsf{SK}_2', \mathsf{CT}^*)$
- If $\mathbf{r}' = \mathbf{r}$, output $\mathbf{k}$. Else, output $\perp$.

Figure 5: Circuits used in the lockable obfuscation

f

Input of sender: $(\mathsf{td}, \mathbf{c}, \mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*, \mathsf{SK}_2)$
Input of receiver: $(\mathbf{d}, r_1, \ldots, r_\ell)$

- If $\left(\mathbf{c} \leftarrow \mathsf{Comm}\left(1^\lambda, (r_1, \ldots, r_\ell); \mathbf{d}\right)\right) \bigwedge \left(\forall i \in [\ell], \mathbf{c}_i^* \leftarrow \mathsf{Comm}\left(1^\lambda, \mathsf{td}_i\right)\right)$, output SK_2.
- Otherwise, output $\perp$.

Figure 6: Description of the function f associated with the SFE.

Construction. We construct a protocol (S, R) in Figure 7 for a NP language $\mathcal{L}$, and the following lemma shows that (S, R) is a quantum extraction protocol.

Lemma 35. *Assuming the quantum security of* Comm, SFE, qFHE *and* (S, R) *is a quantum extraction protocol for* $\mathcal{L}$ *secure against quantum adversaries.*

Proof.

Quantum Zero-Knowledge. Let $(\mathbf{z}, \mathbf{w}) \in \mathcal{R}$, and let R^* be a QPT malicious receiver. Associated with R^* is the QPT algorithm Sim – in fact, Sim is a classical PPT algorithm that only uses R^* as a black-box – defined below.

Description of Sim**.**

- It first receives $\mathbf{c}$ from R^*. It performs the following operations:
  - Compute the $\mathsf{qFHE.Setup}$ to obtain $(\mathsf{PK}_1, \mathsf{SK}_1)$.

Input of sender: $(\mathbf{z}, \mathbf{w})$.
Input of receiver: $\mathbf{z}$

- R: sample $(r_1, \ldots, r_\ell) \xleftarrow{\$} \{0,1\}^{\ell \cdot \mathrm{poly}(\lambda)}$. Compute $\mathbf{c} \leftarrow \mathsf{Comm}\left(1^\lambda, (r_1, \ldots, r_\ell; \mathbf{d})\right)$, where $\ell = \lambda$ and $\mathbf{d}$ is the randomness used to compute $\mathbf{c}$. Send $\mathbf{c}$ to S.
- S:
  - Compute the $\mathsf{qFHE.Setup}$ twice; $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{qFHE.Setup}(1^\lambda)$ for $i \in \{1,2\}$.
  - Compute $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, (\mathsf{td}||\mathbf{w}))$, where $\mathsf{td} \xleftarrow{\$} \{0,1\}^\lambda$.
  - Compute $\widetilde{\mathbf{C}} \leftarrow \mathsf{Obf}(1^\lambda, \mathbf{C}[\mathbf{r}, \mathbf{k}, \mathsf{SK}_1, \mathsf{CT}^*])$, where $\mathbf{r} \xleftarrow{\$} \{0,1\}^\lambda$ and $\mathbf{k} \xleftarrow{\$} \{0,1\}^\lambda$, CT^* is defined below and $\mathbf{C}[\mathbf{r}, \mathbf{k}, \mathsf{SK}_1, \mathsf{CT}^*]$ is defined in Figure 5.
    - $\mathsf{CT}^* \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_2, \mathbf{r})$

  Send $\mathsf{msg}_1 = \left(\mathsf{CT}_1, \widetilde{\mathbf{C}}, \mathsf{otp} := \mathbf{k} \oplus \mathsf{SK}_1\right)$.
- R: compute $\mathbf{c}_i^* \leftarrow \mathsf{Comm}\left(1^\lambda, 0; r_i\right)$ for $i \in [\ell]$. Send $(\mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*)$ to S.
- S and R run SFE, associated with the two-party functionality f defined in Figure 6; S takes the role of $\mathsf{SFE.S}$ and R takes the role of $\mathsf{SFE.R}$. The input to $\mathsf{SFE.S}$ is $(\mathsf{td}, \mathbf{c}, \mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*, \mathsf{SK}_2)$ and the input to $\mathsf{SFE.R}$ is $(\mathbf{d}, r_1, \ldots, r_\ell)$.

Figure 7: Quantum Extraction Protocol (S, R)

  - Compute $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, \perp)$.
  - Compute the obfuscated circuit $\widetilde{\mathbf{C}} \leftarrow \mathsf{LObf.Sim}\left(1^\lambda, 1^{|\mathbf{C}|}\right)$.
  - Sample $\mathsf{otp} \xleftarrow{\$} \{0,1\}^{|\mathsf{SK}_1|}$.

  Send $(\mathsf{CT}_1, \widetilde{\mathbf{C}}, \mathsf{otp})$.

- It then receives $(\mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*)$ from the receiver.
- It executes SFE with R^*; Sim takes the role of $\mathsf{SFE.S}$ with the input $\perp$.
- Finally, it outputs the final state of R^*.

We show below that the view of R^* when interacting with the honest sender is indistinguishable, by a QPT distinguisher, from the output of Sim. Consider the following hybrids:

$\underline{\mathsf{Hyb}_1}$: In this hybrid, R^* is interacting with the honest sender S. The output of this hybrid is the output of R^*.

$\underline{\mathsf{Hyb}_2}$: In this hybrid, we define a hybrid sender, denoted by $\mathsf{Hyb}_2.\mathsf{S}$: it behaves exactly like S except that in SFE, the input of $\mathsf{SFE.S}$ is $\bot$.

Consider the following claim.

Claim 36. $\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) \approx_Q \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_2.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right).$

Proof. To prove this claim, we first need to show that the probability that the receiver R^* commits to $\mathbf{w}$ is negligible. Consider the following claim.

Claim 37. *Assuming the quantum security of* Comm, $\mathbf{LObf}$ *and* qFHE, *the following holds:*

$$\Pr\left[\begin{array}{c} \exists r_1,\ldots,r_\ell,\mathbf{d}, \\ \big(\mathbf{c}{=}\mathsf{Comm}\big(1^\lambda,(r_1,\ldots,r_\ell);\mathbf{d}\big)\big) \\ \bigwedge \\ \big(\forall i{\in}[\ell],\mathbf{c}_i^*{=}\mathsf{Comm}\big(1^\lambda,\mathsf{td}_i;r_i\big)\big){=}1 \end{array} : \begin{array}{c} \mathbf{c}{\leftarrow}\mathsf{R}^*(1^\lambda,\mathbf{z},\cdot) \\ \mathsf{td}\xleftarrow{\$}\{0,1\}^\lambda \\ (\mathsf{PK}_i,\mathsf{SK}_i){\leftarrow}\mathsf{qFHE.Setup}(1^\lambda),\forall i{\in}\{1,2\} \\ \mathsf{CT}_1{\leftarrow}\mathsf{qFHE.Enc}(\mathsf{PK}_1,(\mathsf{td}||\mathbf{w})) \\ \mathbf{r}\xleftarrow{\$}\{0,1\}^\lambda \\ \mathbf{k}\xleftarrow{\$}\{0,1\}^{|\mathsf{SK}_1|} \\ \mathsf{CT}^*{\leftarrow}\mathsf{qFHE.Enc}(\mathsf{PK}_2,\mathbf{r}) \\ \widetilde{\mathbf{C}}{\leftarrow}\mathsf{Obf}(1^\lambda,\mathbf{C}[\mathbf{r},\mathbf{k},\mathsf{SK}_1,\mathsf{CT}^*]) \\ \mathsf{otp}{=}\mathbf{k}{\oplus}\mathsf{SK}_1 \\ (\mathbf{c}_1^*,\ldots,\mathbf{c}_\ell^*){\leftarrow}\mathsf{R}^*(1^\lambda,\mathbf{z},\cdot) \end{array}\right] \le \mathsf{negl}(\lambda),$$

for some negligible function negl.

Proof. We define the event BAD_1 as follows:

$\mathsf{BAD}_1 = 1$ if there exists $r_1, \ldots, r_\ell, \mathbf{d}$ such that

$$\left(\mathbf{c} = \mathsf{Comm}\left(1^\lambda, (r_1, \ldots, r_\ell); \mathbf{d}\right)\right) \bigwedge \left(\forall i \in [\ell], \mathbf{c}_i^* = \mathsf{Comm}\left(1^\lambda, \mathsf{td}_i; r_i\right)\right) = 1,$$

where:

- $\mathbf{c} \leftarrow \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$,
- $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, (\mathsf{td}||\mathbf{w}))$, where $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{qFHE.Setup}(1^\lambda), \forall i \in \{1, 2\}$ and $\mathsf{td} \xleftarrow{\$} \{0, 1\}^\lambda$,
- $\widetilde{\mathbf{C}} \leftarrow \mathsf{Obf}(1^\lambda, \mathbf{C}[\mathbf{r}, \mathbf{k}, \mathsf{SK}_1, \mathsf{CT}^*])$, where $\mathbf{r} \xleftarrow{\$} \{0, 1\}^\lambda$, $\mathbf{k} \xleftarrow{\$} \{0, 1\}^{|\mathsf{SK}_1|}$ and $\mathsf{CT}^* \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_2, \mathbf{r})$,
- $\mathsf{otp} = \mathbf{k} \oplus \mathsf{SK}_1$ and,
- $\mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$ on input $(\mathsf{CT}, \widetilde{\mathbf{C}}, \mathsf{otp})$ outputs $(\mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*)$.

Otherwise, $\mathsf{BAD}_1 = 0$.

Define p_1 to be $\mathsf{p}_1 = \Pr[\mathsf{BAD}_1 = 1]$.

We define a hybrid event $\mathsf{BAD}_{1.1}$ as follows:

$\mathsf{BAD}_{1.1} = 1$ if there exists $r_1, \ldots, r_\ell, \mathbf{d}$ such that

$$\left(\mathbf{c} = \mathsf{Comm}\left(1^\lambda, (r_1, \ldots, r_\ell); \mathbf{d}\right)\right) \bigwedge \left(\forall i \in [\ell], \mathbf{c}_i^* = \mathsf{Comm}\left(1^\lambda, \mathsf{td}_i; r_i\right)\right) = 1,$$

where:

- $\mathbf{c} \leftarrow \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$,
- $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, (\mathsf{td}||\mathbf{w}))$, where $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{qFHE.Setup}(1^\lambda), \forall i \in \{1, 2\}$ and $\mathsf{td} \xleftarrow{\$} \{0, 1\}^\lambda$,
- $\widetilde{\mathbf{C}} \leftarrow \mathsf{Obf}(1^\lambda, \mathbf{C}[\mathbf{r}, \mathbf{k}, \mathsf{SK}_1, \mathsf{CT}^*])$, where $\mathbf{r} \xleftarrow{\$} \{0, 1\}^\lambda$, $\mathbf{k} \xleftarrow{\$} \{0, 1\}^{|\mathsf{SK}_1|}$ and $\underline{\mathsf{CT}^* \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_2, \perp),}$
- $\mathsf{otp} = \mathbf{k} \oplus \mathsf{SK}_1$ and,
- $\mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$ on input $(\mathsf{CT}, \widetilde{\mathbf{C}}, \mathsf{otp})$ outputs $(\mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*)$.

Otherwise, $\mathsf{BAD}_{1.1} = 0$.

We define $\mathsf{p}_{1.1}$ as $\mathsf{p}_{1.1} = \Pr[\mathsf{BAD}_{1.1} = 1]$.

From the quantum security of qFHE, it holds that $|\mathsf{p}_1 - \mathsf{p}_{1.1}| \leq \mathsf{negl}(\lambda)$ for some negligible function negl. Note that we crucially rely on the fact that SFE, that requires the sender to input SK_2, is only executed after the receiver sends $(\mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*)$.

We define a hybrid event $\mathsf{BAD}_{1.2}$ as follows:

$\mathsf{BAD}_{1.2} = 1$ if there exists $r_1, \ldots, r_\ell, \mathbf{d}$ such that

$$\left(\mathbf{c} = \mathsf{Comm}\left(1^\lambda, (r_1, \ldots, r_\ell); \mathbf{d}\right)\right) \bigwedge \left(\forall i \in [\ell], \mathbf{c}_i^* = \mathsf{Comm}\left(1^\lambda, \mathsf{td}_i; r_i\right)\right) = 1,$$

where:

- $\mathbf{c} \leftarrow \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$,
- $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, (\mathsf{td}||\mathbf{w}))$, where $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{qFHE.Setup}(1^\lambda), \forall i \in \{1, 2\}$ and $\mathsf{td} \xleftarrow{\$} \{0, 1\}^\lambda$,
- $\underline{\widetilde{\mathbf{C}} \leftarrow \mathsf{LObf.Sim}\left(1^\lambda, 1^{|\mathsf{C}|}\right),}$
- $\mathsf{otp} = \mathbf{k} \oplus \mathsf{SK}_1$ and,
- $\mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$ on input $(\mathsf{CT}, \widetilde{\mathbf{C}}, \mathsf{otp})$ outputs $(\mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*)$.

Otherwise, $\mathsf{BAD}_{1.2} = 0$.

We define $\mathsf{p}_{1.2}$ as $\mathsf{p}_{1.2} = \Pr[\mathsf{BAD}_{1.2} = 1]$. From the quantum security of **LObf**, it follows that $|\mathsf{p}_{1.1} - \mathsf{p}_{1.2}| \leq \mathsf{negl}(\lambda)$. Note that we crucially use the fact that the lock r is uniformly sampled and independently of the function that is obfuscated.

We define a hybrid event $\mathsf{BAD}_{1.3}$ as follows:

$\mathsf{BAD}_{1.3} = 1$ if there exists $r_1, \ldots, r_\ell, \mathbf{d}$ such that

$$\left(\mathbf{c} = \mathsf{Comm}\left(1^\lambda, (r_1, \ldots, r_\ell); \mathbf{d}\right)\right) \bigwedge \left(\forall i \in [\ell], \mathbf{c}^*_i = \mathsf{Comm}\left(1^\lambda, \mathsf{td}_i; r_i\right)\right) = 1,$$

where:

- $\mathbf{c} \leftarrow \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$,
- $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, (\mathsf{td}||\mathbf{w}))$, where $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{qFHE.Setup}(1^\lambda), \forall i \in \{1, 2\}$ and $\mathsf{td} \xleftarrow{\$} \{0,1\}^\lambda$,
- $\widetilde{\mathbf{C}} \leftarrow \mathsf{LObf.Sim}\left(1^\lambda, 1^{|\mathsf{C}|}\right)$,
- $\underline{\mathsf{otp} \xleftarrow{\$} \{0,1\}^{|\mathsf{SK}_1|}}$ and,
- $\mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$ on input $(\mathsf{CT}, \widetilde{\mathbf{C}}, \mathsf{otp})$ outputs $(\mathbf{c}^*_1, \ldots, \mathbf{c}^*_\ell)$.

Otherwise, $\mathsf{BAD}_{1.3} = 0$.

We define $\mathsf{p}_{1.3}$ as $\mathsf{p}_{1.3} = \Pr[\mathsf{BAD}_{1.3} = 1]$. Observe that $\mathsf{p}_{1.2} = \mathsf{p}_{1.3}$.

We define a hybrid event $\mathsf{BAD}_{1.4}$ as follows:

$\mathsf{BAD}_{1.4} = 1$ if there exists $r_1, \ldots, r_\ell, \mathbf{d}$ such that

$$\left(\mathbf{c} = \mathsf{Comm}\left(1^\lambda, (r_1, \ldots, r_\ell); \mathbf{d}\right)\right) \bigwedge \left(\forall i \in [\ell], \mathbf{c}^*_i = \mathsf{Comm}\left(1^\lambda, \mathsf{td}_i; r_i\right)\right) = 1,$$

where:

- $\mathbf{c} \leftarrow \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$,
- $\underline{\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, \bot)}$, where $(\mathsf{PK}_i, \mathsf{SK}_i) \leftarrow \mathsf{qFHE.Setup}(1^\lambda), \forall i \in \{1, 2\}$ and $\mathsf{td} \xleftarrow{\$} \{0,1\}^\lambda$,
- $\widetilde{\mathbf{C}} \leftarrow \mathsf{LObf.Sim}\left(1^\lambda, 1^{|\mathsf{C}|}\right)$,
- $\mathsf{otp} \xleftarrow{\$} \{0,1\}^{|\mathsf{SK}_1|}$ and,
- $\mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)$ on input $(\mathsf{CT}, \widetilde{\mathbf{C}}, \mathsf{otp})$ outputs $(\mathbf{c}^*_1, \ldots, \mathbf{c}^*_\ell)$.

Otherwise, $\mathsf{BAD}_{1.4} = 0$.

We define $\mathsf{p}_{1.4}$ as $\mathsf{p}_{1.4} = \Pr[\mathsf{BAD}_{1.4} = 1]$. From the quantum security of qFHE, it follows that $|\mathsf{p}_{1.3} - \mathsf{p}_{1.4}| \leq \mathsf{negl}(\lambda)$. Moreover, note that $\mathsf{p}_{1.4} = 2^{-\lambda}$ since td is information-theoretically hidden from R^*. Thus, we have that $\mathsf{p}_1 \leq \mathsf{negl}(\lambda)$.

□

We now use Claim 37 to prove Claim 36. Conditioned on $\mathsf{BAD}_1 \neq 1$, it holds that the view of R^* after its interaction with S is indistinguishable (by a QPT algorithm) from the view of R^* after its interaction with $\mathsf{Hyb}_2.\mathsf{S}$; this follows from the IND-security of SFE against quantum receivers since $f((\mathsf{td}, \mathbf{c}, \mathbf{c}^*_1, \ldots, \mathbf{c}^*_\ell, \mathsf{SK}_2), (\mathbf{d}, r_1, \ldots, r_\ell)) = f((\bot), (\mathbf{d}, r_1, \ldots, r_\ell))$.

□

$\underline{\mathsf{Hyb}_3}$: We define a hybrid sender, denoted by $\mathsf{Hyb}_3.\mathsf{S}$: it behaves exactly like $\mathsf{Hyb}_2.\mathsf{S}$ except that CT^* in $\widetilde{\mathbf{C}}$ is generated as $\mathsf{CT}^* \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_2, \perp)$.

Assuming the quantum security of qFHE, we have:

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_2.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) \approx_Q \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_3.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right)$$

$\underline{\mathsf{Hyb}_4}$: We define a hybrid sender, denoted by $\mathsf{Hyb}_4.\mathsf{S}$: it behaves exactly like $\mathsf{Hyb}_3.\mathsf{S}$ except that $\widetilde{\mathbf{C}}$ is generated as $\widetilde{\mathbf{C}} \leftarrow \mathsf{LObf.Sim}\left(1^\lambda, 1^{|\mathbf{C}|}\right)$.

Assuming the quantum security of **LObf**, we have:

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_3.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) \equiv \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_4.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right)$$

$\underline{\mathsf{Hyb}_5}$: We define a hybrid sender, denoted by $\mathsf{Hyb}_5.\mathsf{S}$: it behaves exactly like $\mathsf{Hyb}_4.\mathsf{S}$ except that OT is generated uniformly at random.

The following holds unconditionally:

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_4.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) \equiv \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_5.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right)$$

$\underline{\mathsf{Hyb}_6}$: We define a hybrid sender, denoted by $\mathsf{Hyb}_6.\mathsf{S}$: it behaves exactly like $\mathsf{Hyb}_5.\mathsf{S}$ except that CT_1 is generated as $\mathsf{CT}_1 \leftarrow \mathsf{qFHE.Enc}(\mathsf{PK}_1, \perp)$.

Assuming the quantum security of qFHE, we have:

$$\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_5.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right) \approx_Q \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hyb}_6.\mathsf{S}(1^\lambda, \mathbf{z}, \mathbf{w}), \mathsf{R}^*(1^\lambda, \mathbf{z}, \cdot)\rangle\right)$$

Since $\mathsf{Hyb}_6.\mathsf{S}$ is identical to Sim, the proof of quantum zero-knowledge follows.

Extractability. Let $\mathsf{S}^* = (\mathsf{S}_1^*, \mathsf{S}_2^*)$ be a semi-malicious QPT, where S_2^* is the QPT involved in SFE. Denote by $\mathsf{R} = (\mathsf{R}_1, \mathsf{R}_2, \mathsf{R}_3)$ the PPT algorithms of the honest receiver. In particular, R_3 is the algorithm that the receiver runs in SFE protocol. Let

$$\mathcal{E}_{\mathsf{SFE}}(\cdot\,; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*) := \left\langle \mathsf{R}_3(1^\lambda, \mathbf{d}, r_1, \ldots, r_\ell), \mathsf{S}_2^*(1^\lambda, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*, \cdot)\right\rangle$$

be the interaction channel induced on the private quantum input of S^* by the interaction with R in the SFE protocol for the functionality f with inputs $\mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*$. Without loss of generality, assume that this channel also outputs the classical message output of SFE.

Consider the following extractor Ext, that takes as input the efficient quantum circuit description of $\mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot)$, and the instance $\mathbf{z}$.

$\mathsf{Ext}(1^\lambda, S^*, \mathbf{z}, \cdot)$**:**

- Run R_1 to compute $\mathbf{c}$, $\mathbf{d}$, and $r_1, \ldots, r_\ell$.
- Apply the channel $S_1^*(1^\lambda, \mathbf{z}, \mathbf{w}, \mathbf{c}, \cdot)$.

- Let $(\mathsf{CT}_1, \widetilde{\mathbf{C}}, \mathsf{otp})$ denote the classical messages outputted by S_1^*, and let ρ denote the rest of the state.
- With CT_1, homomorphically commit to td, obtaining

$$\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathbf{c}^* := \mathsf{Comm}(1^\lambda, \mathsf{td}))$$

.

- Encrypt $(\mathbf{d}, \mathbf{c}, r_1, \ldots, r_\ell)$, and ρ, and homomorphically apply the channel $\mathcal{E}_{\mathsf{SFE}}(\cdot\, ; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*)$
- Let $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{SFE.Out} \otimes \rho')$ be the output of the previous step, where $\mathsf{SFE.Out}$ is the classical output of the SFE protocol.
- Apply $\widetilde{\mathbf{C}}$ to the qFHE encryption of $\mathsf{SFE.Out}$. Note that we are assuming that classical messages have classical ciphertexts, so this computation is a classical one. Let k be the output of $\widetilde{\mathbf{C}}\,(\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{SFE.Out}))$.
- Let $\mathsf{SK}_1 := k \oplus \mathsf{otp}$, and decrypt CT_1 with SK_1. If the decryption is successful and the message $\mathbf{w}$ is recovered, let Ext_2 output $\mathbf{w}$.
- Use SK_1 to decrypt the ciphertext $\mathsf{qFHE.Enc}(\mathsf{PK}_1, \mathsf{SFE.Out} \otimes \rho')$, and let Ext_1 output ρ'.

Claim 38. $\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{R}(1^\lambda, \mathbf{z})\rangle\right) \approx_Q \mathsf{Ext}_1\left(1^\lambda, \mathsf{S}^*, \mathbf{z}, \cdot\right)$

Proof. Let $\mathsf{R}_{\mathcal{D}}$ be the quantum register of a distinguisher $\mathcal{D}$. Let $\mathcal{F} : \mathsf{R}_{\mathcal{D}} \to \mathsf{R}_{\mathcal{D}}$ be the following channels, parametrized by $\mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*$,

$$\mathcal{F}(\rho; \mathbf{d}, r_1, ..., r_\ell, \mathbf{w}, \mathbf{c}, \mathbf{c}^*) := \left(\left[\mathcal{E}_{\mathsf{SFE}}(\cdot\, ; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*) \circ S_1^*(1^\lambda, \mathbf{z}, \mathbf{w}, \mathbf{c}, \cdot)\right] \otimes \mathsf{Id}\right)(\rho)\,.$$

The identity is acting on the distinguisher's private state, and the composition $\mathcal{E}_{\mathsf{SFE}}(\cdot\, ; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*) \circ S_1^*(1^\lambda, \mathbf{z}, \mathbf{w}, \mathbf{c}, \cdot)$ acts on the private state of S^*. We do not write td as a parameter to $\mathcal{F}$, because td is generated by S_1^* and assumed to be part of the sender's private state. We do add it as a parameter to $\mathcal{E}_{\mathsf{SFE}}$ to be consistent and to remind ourselves that the td is input into the SFE protocol.

Note that when $\mathbf{d}, r_1, \ldots, r_\ell, \mathbf{c}$ and $\mathbf{c}^*$ are generated by the honest R in the protocol, we have

$$\mathcal{F}(\rho; \mathbf{d}, r_1, ..., r_\ell, \mathbf{w}, \mathbf{c}, \mathbf{c}^*) = \left(\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, \mathbf{z}, \mathbf{w}, \cdot), \mathsf{R}(1^\lambda, \mathbf{z})\rangle\right) \otimes \mathsf{Id}\right)(\rho)$$

We will show that when $\mathbf{d}, r_1, \ldots, r_\ell, \mathbf{c}$ are generated the same way as the honest R would generate them in the first round R_1, but the commitment $\mathbf{c}^* = \mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*$ is a commitment to the witness, $\mathbf{w}$, instead, we have

$$\mathcal{F}(\rho; \mathbf{d}, r_1, ..., r_\ell, \mathbf{w}, \mathbf{c}, \mathbf{c}^*_{\mathbf{w}}) = \left(\mathsf{Ext}_1\left(1^\lambda, \mathsf{S}^*, \mathbf{z}, \cdot\right) \otimes \mathsf{Id}\right)(\rho)$$

Our goal is to show that these two cases, $\mathbf{c}^*$ and $\mathbf{c}^*_{\mathbf{w}}$, are quantum computationally indistinguishable.

To see why this last equation is true, we are using the perfect correctness of both the qFHE scheme and of the lockable obfuscator, as well as the fact that the S^* is semi-malicious, which means it has to follow the protocol. This means that when S_1^* outputs $(\mathsf{CT}_1, \widetilde{\mathbf{C}}, \mathsf{otp})$, the extractor has a valid ciphertext CT_1 encrypted with a key PK_1, which in turn is one-time padded, $\mathsf{SK}_1 \oplus k = \mathsf{otp}$. Furthermore, the one-time pad value k is the output of $\widetilde{\mathbf{C}}$ if an input releases the lock, and $\widetilde{\mathbf{C}}$ is a correct lockable obfuscation of the desired circuit.

After this, the extractor performed $\mathcal{E}_{\mathsf{SFE}}(\cdot\,; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*_{\mathbf{w}})$ homomorphically, which results in the extractor having an encryption of SK_2 under PK_1. This is true because the extractor is able to commit to the witness inside the encryption, and the semi-malicious sender has to engage correctly in the SFE. Since the extractor can now use the $\widetilde{\mathbf{C}}$ to obtain SK_1, we can summarize the whole operation of the extractor as follows. Let $(\mathsf{CT}_1, \widetilde{\mathbf{C}}, \mathsf{otp}) \otimes \rho'$ be the state of the distinguisher after S_1^*. Then, the extractor performs

$$((\mathsf{Dec}(\mathsf{SK}_1, \cdot) \circ \mathsf{Eval}\left(\mathcal{E}_{\mathsf{SFE}}\left(\cdot\,; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*_{\mathbf{w}}\right), \cdot\right) \circ \mathsf{Enc}(\mathsf{PK}_1, \mathbf{c}^*_{\mathbf{w}}, \cdot)) \otimes \mathsf{Id})\left(\rho'\right)$$

By correctness of the qFHE scheme, this is the same as the extractor performing

$$\left(\left[\mathcal{E}_{\mathsf{SFE}}(\cdot\,; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*_{\mathbf{w}}) \circ S_1^*(1^\lambda, \mathbf{z}, \mathbf{w}, \mathbf{c}, \cdot)\right] \otimes \mathsf{Id}\right)(\rho)$$

on the distinguisher's state.

To show that the view of the sender when interacting with the honest receiver is indistinguishable (against polynomial time quantum algorithms) from the view of the sender when interacting with the extractor.

<u>Hyb_1</u>: The output of this hybrid is the view of the sender when interacting with the honest receiver.

<u>Hyb_2</u>: We define a hybrid receiver $\mathsf{Hyb}_2.\mathsf{R}$ that behaves like the honest receiver except that the input of $\mathsf{Hyb}_2.\mathsf{R}$ in SFE is $\perp$. The output of this hybrid is the view of the sender when interacting with $\mathsf{Hyb}_2.\mathsf{R}$.

The quantum indistinguishability of Hyb_1 and Hyb_2 follows from the semantic security of SFE against quantum polynomial time adversaries.

<u>Hyb_3</u>: We define a hybrid receiver $\mathsf{Hyb}_3.\mathsf{R}$ that behaves like $\mathsf{Hyb}_2.\mathsf{R}$ except that it sets $\mathbf{c}$ to be $\mathbf{c} = \mathsf{Comm}(1^\lambda, 0; \mathbf{d})$. The output of this hybrid is the view of the receiver when interacting with $\mathsf{Hyb}_3.\mathsf{R}$.

The quantum indistinguishability of Hyb_2 and Hyb_3 follows from the quantum computational hiding of Comm.

<u>Hyb_4</u>: We define a hybrid receiver $\mathsf{Hyb}_4.\mathsf{R}$ that sets $\mathbf{c}^*_i = \mathsf{Comm}(1^\lambda, \mathsf{td}_i; r_i)$, for every $i \in [\ell]$.

The quantum indistinguishability of Hyb_3 and Hyb_4 follows from the quantum computational hiding of Comm.

<u>Hyb_5</u>: We define a hybrid receiver $\mathsf{Hyb}_5.\mathsf{R}$ that behaves as $\mathsf{Hyb}_4.\mathsf{R}$ except that it sets $\mathbf{c}$ to be $\mathbf{c} = \mathsf{Comm}(1^\lambda, (r_1, \ldots, r_\ell); \mathbf{d})$, where r_i is the randomness used in the commitment $\mathbf{c}^*_i$.

The quantum indistinguishability of Hyb_4 and Hyb_5 follows from the quantum computational hiding of Comm.

$\underline{\mathsf{Hyb}_6}$: The output of this hybrid is the output of the extractor.

The quantum indistinguishability of Hyb_5 and Hyb_6 follows from the semantic security of SFE against polynomial time quantum adversaries.

□

□

Acknowledgements

We are grateful to Kai-Min Chung for many clarifications regarding quantum zero-knowledge proof and argument systems. We thank Thomas Vidick and Urmila Mahadev for answering questions about noisy trapdoor claw-free functions. We thank Abhishek Jain for helpful discussions and pointing us to the relevant references.

References

[AJ17] Prabhanjan Ananth and Abhishek Jain. On secure two-party computation in three rounds. In *Theory of Cryptography Conference*, pages 612–644. Springer, 2017.

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

[Bar01a] Boaz Barak. How to go beyond the black-box simulation barrier. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 106–115. IEEE, 2001.

[Bar01b] Boaz Barak. How to go beyond the black-box simulation barrier. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 106–115. IEEE, 2001.

[BBK$^+$16] Nir Bitansky, Zvika Brakerski, Yael Kalai, Omer Paneth, and Vinod Vaikuntanathan. 3-message zero knowledge against human ignorance. In *Theory of Cryptography Conference*, pages 57–83. Springer, 2016.

[BCM$^+$18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 320–331. IEEE, 2018.

[BCPR16a] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. *SIAM Journal on Computing*, 45(5):1910–1952, 2016.

[BCPR16b] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. *SIAM Journal on Computing*, 45(5):1910–1952, 2016.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private ot from lwe. In *Theory of Cryptography Conference*, pages 370–390. Springer, 2018.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001.

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low t-gate complexity. In *Annual Cryptology Conference*, pages 609–629. Springer, 2015.

[BJSW16] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for qma. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 31–40. IEEE, 2016.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 671–684. ACM, 2018.

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1091–1102. ACM, 2019.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. Citeseer, 1986.

[BP16] Zvika Brakerski and Renen Perlman. Lattice-based fully dynamic multi-key fhe with short ciphertexts. In *Annual International Cryptology Conference*, pages 190–213. Springer, 2016.

[Bra18a] Zvika Brakerski. Quantum fhe (almost) as secure as classical. In *Annual International Cryptology Conference*, pages 67–95. Springer, 2018.

[Bra18b] Zvika Brakerski. Quantum fhe (almost) as secure as classical. In *Annual International Cryptology Conference*, pages 67–95. Springer, 2018.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

[CCKV08] André Chailloux, Dragos Florin Ciocan, Iordanis Kerenidis, and Salil Vadhan. Interactive and noninteractive zero knowledge are equivalent in the help model. In *Theory of Cryptography Conference*, pages 501–534. Springer, 2008.

[CM15] Michael Clear and Ciaran McGoldrick. Multi-identity and multi-key leveled fhe from learning with errors. In *Annual Cryptology Conference*, pages 630–656. Springer, 2015.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM Journal on Computing*, 29(1):1–28, 1999.

[G$^+$09] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In *Stoc*, volume 9, pages 169–178, 2009.

[GHKW17] Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In *Theory of Cryptography Conference*, pages 537–566. Springer, 2017.

[GHV10] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. i-hop homomorphic encryption and rerandomizable yao circuits. In *Annual Cryptology Conference*, pages 155–172. Springer, 2010.

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

[GKVW19] Rishab Goyal, Venkata Koppula, Satyanarayana Vusirikala, and Brent Waters. On perfect correctness in (lockable) obfuscation. 2019.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–621. IEEE, 2017.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 174–187. IEEE, 1986.

[HKSZ08] Sean Hallgren, Alexandra Kolla, Pranab Sen, and Shengyu Zhang. Making classical honest verifier zero knowledge protocols secure against quantum attacks. In *International Colloquium on Automata, Languages, and Programming*, pages 592–603. Springer, 2008.

[JKMR06a] Rahul Jain, Alexandra Kolla, Gatis Midrijanis, and Ben W Reichardt. On parallel composition of zero-knowledge proofs with black-box quantum simulators. *arXiv preprint quant-ph/0607211*, 2006.

[JKMR06b] Rahul Jain, Alexandra Kolla, Gatis Midrijanis, and Ben W Reichardt. On parallel composition of zero-knowledge proofs with black-box quantum simulators. *arXiv preprint quant-ph/0607211*, 2006.

[KK19] Yael Tauman Kalai and Dakshita Khurana. Non-interactive non-malleability from quantum supremacy. In *Annual International Cryptology Conference*, pages 552–582. Springer, 2019.

[Kob08] Hirotada Kobayashi. General properties of quantum zero-knowledge proofs. In *Theory of Cryptography Conference*, pages 107–124. Springer, 2008.

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. *IACR Cryptology ePrint Archive*, 2019:279, 2019.

[Mah18a] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 332–338. IEEE, 2018.

[Mah18b] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018.

[Mat06] Keiji Matsumoto. A simpler proof of zero-knowledge against quantum attacks using grover's amplitude amplification. *arXiv preprint quant-ph/0602186*, 2006.

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key fhe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 735–763. Springer, 2016.

[NC02] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 160–176. Springer, 2003.

[PS16] Chris Peikert and Sina Shiehian. Multi-key fhe from lwe, revisited. In *Theory of Cryptography Conference*, pages 217–238. Springer, 2016.

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *Theory of Cryptography Conference*, pages 403–418. Springer, 2009.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 135–152. Springer, 2012.

[Unr13] Dominique Unruh. Everlasting multi-party computation. In *Annual Cryptology Conference*, pages 380–397. Springer, 2013.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 497–527. Springer, 2016.

[VZ19] Thomas Vidick and Tina Zhang. Classical zero-knowledge arguments for quantum computations. *arXiv preprint arXiv:1902.05217*, 2019.

[Wat09a] John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

[Wat09b] John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under lwe. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611. IEEE, 2017.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.