Cryptographic Simulation Techniques with Applications to Quantum Zero-Knowledge and Copy-Protection

by

Rolando L. La Placa Massa

B.A., Harvard University (2014)

Submitted to the Department of Physics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author .....
Department of Physics
May 21, 2021

Certified by .....
Aram W. Harrow
Associate Professor
Thesis Supervisor

Accepted by .....
Deepto Chakrabarty
Associate Department Head

Cryptographic Simulation Techniques with Applications to Quantum Zero-Knowledge and Copy-Protection

by

Rolando L. La Placa Massa

Submitted to the Department of Physics
on May 21, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Bob is stuck doing a crossword puzzle and is starting to think that the puzzle is impossible to complete. Alice assures Bob that the puzzle can be solved, but she wants to prove it without revealing a single entry of the puzzle. Their cryptographer friend, Eve, tells them that Alice can prove it by using a *zero-knowledge* (ZK) protocol. These protocols are a cornerstone of modern cryptography, yet most of the work has been limited to the classical setting. Since Bob has a quantum computer, Alice needs to be careful choosing the right protocol to make sure it is a *quantum zero-knowledge* (QZK) protocol, guaranteeing that quantum Bob cannot learn anything about the puzzle except that it has a solution.

Proving the security of ZK protocols comes with additional hurdles when adversaries are quantum capable, in part because the main tool used in the classical setting, *rewinding*, has additional limitations in the quantum case. While one version of quantum rewinding introduced by Watrous has been successfully used to construct QZK protocols, most of the classical ZK results have been challenging to port to the quantum setting. Ideally, we want quantum secure protocols with the same desirable properties that have been achieved in the classical literature, like concurrent security or low-round complexity. In this thesis, we introduce new quantum simulation techniques and apply them to construct the following QZK protocols assuming the quantum hardness of learning with errors (QLWE).

- $O(1)$**-round black-box QZK classical argument system for NP**: We use techniques developed in the context of 'tests of quantumness' to obtain an extraction mechanism that can be leveraged to construct a QZK simulator.
- **Public coin bounded concurrent black-box QZK proof system for NP and QMA**: We introduce the technique of *block rewinding* and use it to obtain a concurrent QZK simulator.
- **Simulatable and extractable quantum proofs of knowledge for NP**: We construct QPoK with desirable properties needed for *composability*. The technique combines Watrous' rewinding with a recently studied cryptographic tool, statistical receiver-private oblivious transfer. This is the first construction of QPoK with the desired composability features.

We also introduce a new non-black-box knowledge extraction technique using quantum fully homomorphic encryption (QFHE) and lockable obfuscation. One of our main results is that we can adapt this non-black-box technique to the setting of quantum copy-protection to prove that it is impossible to quantum copy-protect arbitrary unlearnable functions. This resolves a long-standing open problem in the negative, assuming QLWE and the existence of QFHE.

Our impossibility result states that we can't construct quantum copy-protection for arbitrary functions. However, we can hope to do it for restricted families of functions like point functions or compute-and-compare functionalities. While this remains an interesting and challenging open question, we show that provable secure constructions in a standard model (without oracles) are possible if we consider weaker security guarantees from those of quantum copy-protection. For this purpose, we introduce the notion of Secure Software Leasing (SSL), and construct an SSL scheme for a general class of evasive circuits.

Thesis Supervisor: Aram W. Harrow
Title: Associate Professor

To my parents, Gladys I. Massa and Rolando R. La Placa.
and
to the memory of my grandfather, Enrique R. La Placa.

Acknowledgments

I am lucky to have Aram Harrow as my PhD advisor. Aram has been incredibly supportive of my research interests, always willing to discuss research ideas and help me figure out the right approaches to take. He always pushed me to pursue my own ideas, encouraged me to reach out to others, to travel and go to conferences, and gave me good advice on any topic I approached him with. I hope that his problem-solving approach, his attitude towards research, his clear communication ability, his patience and dedication, among other things, will stay with me long after grad school. He has been a great mentor to me.

Speaking of great mentors, I will be forever indebted to my biggest collaborator, Prabhanjan Ananth, who hosted me at UCSB during Fall of 2019. Not only did I learn a great amount of cryptography thanks to him, I also became a better researcher, improved my technical writing, learned to persevere when tackling challenging problems, and discovered how to enjoy the research process while never forgetting about the curiosity that led me to research in the first place. It has been a pleasure working with Prabhanjan, and I am grateful for having him as a mentor and friend.

I am thankful for all my other collaborators during my time at MIT: Alex Dalzell, John Napp, Fernando Brandão, Dax Koh, and Kai-Min Chung. I am especially grateful for my friendship with Alex and John, and I have to thank them for the countless times I bothered them with research questions.

I wouldn't have gotten to the stage of writing a PhD thesis if it wasn't for all the amazing mentors I had before grad school. I started along this path thanks to my high school teachers Rafael Mirabal and Judith Martinez, who introduced me to math and physics as well as motivate me to learn as much as possible. Héctor Jiménez and Raúl Portuondo gave me the opportunity and helped me train to compete in physics olympiads, which cemented my interest in pursuing physics in college. I went into grad school inspired by everything I learned from Subir Sachdev in college. To all of you, thank you.

My time at MIT has been shared with many smart and wonderful people who I had many discussions with. At the risk of missing someone, Nilin Abrahamsen, Eric Anshuetz, Srinivasan Arunachalam, Shankar Balasubramanian, Shalev Ben-David, Adam Bene Watts, Matt Hagan, Nicole Yunger Halpern, Linghang Kong, Zi-Wen Liu, Guang Hao Low, Saeed Mehraban, Anand Natarajan, Elina Sendonaris, Mehdi Soleimanifar, Ryuji Takagi, Annie Wei, John Wright, and Elton Zhu, it was a joy having you all around! Thank you to my thesis committee: Isaac Chuang, Aram Harrow, Yael Kalai, and Peter Shor. It is an honor to have you all in my committee.

Thank you to all of my Boston friends (Ali, Sebi, Paola, David, Jorge, Ana, Kidhanis, Alan, George, Diego, Ale, Dio, Kathy, Bryan, Pacheco, Jean, Iulia, Sergio, Bruno, Ricky, Suzy, Barbara, Nick, Gabe, Sophia, Isa, Denise, Carlos...) who have been there for me through my grad school years. I have been the luckiest grad student through all these years thanks to my best friend and fiancée, Carolina Fejgielman. Thank you for believing in me, for all your support, for distracting me from research, and for indulging me in all my different hobbies. Finally, thank you to my family, Rebecca, Gladys and Rolando. Mom and dad, the next pages are for you.

Financial support. During the work presented in this thesis, I was supported by NSF grant CCF-1729369, the MIT Physics department, and the MIT EECS department.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis lies at the intersection of quantum physics and cryptography. The idea to exploit quantum mechanical laws of nature to perform interesting cryptographic tasks without classical counterparts predates the main advent of quantum computation by a few decades. In a manuscript from 1968 [Bra05, Wie83], Wiesner proposed to use quantum mechanics to prepare banknotes that are secure against forgery, i.e. unforgeable banknotes or *quantum money*. It is impossible to guarantee unforgeability solely using classical resources. In principle, classical states (or information stored in a classical fashion) can be copied; thus, given a $1 dollar bill, it is conceivable that Bob is able to replicate the exact same bill provided that he has the appropriate tools. In contrast, the quantum mechanical laws of nature do not let us copy generic quantum states. This is called the *No-Cloning Theorem*, and it is the fundamental fact of quantum mechanics that lets us dream of constructing cryptographic primitives that, like quantum money, wouldn't otherwise exist in a purely classical world.

The term *quantum cryptography* was coined more than a decade later in a paper from 1982 by Bennett, Brassard, Breidbart, and Wiesner himself [BBBW83]. Not long after, Bennett and Brassard published their groundbreaking work introducing another cryptographic notion, *quantum key distribution* (QKD), whose existence crucially relies on quantum mechanics [BB83, BB14]. Their protocol, now called the BB84 protocol, allows for two parties, Alice and Bob, to randomly generate a shared secret key even in the presence of an eavesdropper, Eve. Classically, this would be impossible to achieve, as any message that Alice communicates to Bob can be intercepted and copied by Eve without either Alice or Bob detecting her. QKD is arguably the best known example of a cryptographic task that can be achieved thanks to quantum effects.

The story of quantum cryptography took an unexpected turn in the 90's, when Peter Shor published his seminal algorithm to factor integers [Sho94]. This algorithm made it clear to cryptographers that quantum mechanics is both a blessing and a woe. Unfortunately, the security of a lot of public-key cryptosystems rely on computational assumptions that are broken by Shor's algorithm. This means that their security cannot be guaranteed against quantum capable adversaries, which becomes an increasing threat with the growth of quantum computing. While assuming that it is computationally hard to factor integers is not useful in the quantum case, we

still have candidate post-quantum secure computational assumptions, the flagship being the *Learning with Errors (LWE)*[1] assumption [Reg05, Reg, Pei16]. Nowadays, many cryptographers (certainly quantum cryptographers!) seek to construct protocols whose security can be proven if the LWE assumption holds also against quantum computers.

From its inception, most work on quantum cryptography including the work presented in this thesis, aims to shine light on either of the following questions:

> ***(Q1)*** *Can we construct classical protocols – i.e. protocols that can be implemented using only classical computation – that are secure against quantum capable adversaries? Are existing constructions already quantum secure?*
> ***(Q2)*** *Can we use quantum mechanics to achieve new cryptographic protocols (possibly without classical counterparts like QKD or quantum money)?*

The goal of this thesis is to study both of these question with respect to two cryptographic primitives. Specifically, we study **(Q1)** the quantum security of *zero-knowledge* (ZK) protocols, and **(Q2)** the feasibility of *quantum copy-protection*. While ZK protocols and quantum copy-protection are generally unrelated topics, we use similar techniques to obtain most of our results. In particular, the main underlying technique is that of *extraction*, whose usefulness is better understood in the context of ZK protocols and the Ideal/Real world paradigm of cryptography discussed in the next section.

1.1 Zero-knowledge protocols

1.1.1 Zero-knowledge and the simulation paradigm

Introduced by Goldwasser, Micali, and Rackoff [GMR85], zero-knowledge protocols are a cornerstone of modern cryptography, where they are not only interesting in their own right but also serve as a basic ingredient in the construction of more advanced primitives. Roughly speaking, a ZK protocol is a protocol performed by two parties, a prover, P, and a verifier, V, that allows P to prove to V the validity of a statement *without revealing anything else*. For example, P can convince V that the statement 'this crossword puzzle can be completed correctly' is true without revealing a single entry of the puzzle. Another concrete example is the following scenario. Alice might have \$10,000 in the bank account, and she needs to provide a proof to Bob that she has more than \$1,000. However, Alice does not want to reveal to Bob the exact amount she owns. To achieve this, Alice and Bob could execute a ZK protocol taking the roles of P and V respectively, then Bob would be convinced that she indeed owns more than \$1,000, but he would not learn the exact amount. If Bob trusted Alice in the first place, then there would be no need for a ZK protocol, as Alice could just tell Bob: "I have more than \$1,000 in the bank". The power of ZK protocols comes from

[1] We will denote by QLWE the assumption that LWE is computationally hard also for efficient quantum algorithms.

the fact that the parties involved do not have to trust each other. At the end of the protocol, there is no way for Alice to convince Bob that she owns more than $1,000, unless she actually does. Similarly, even if Bob tried to get more information from Alice besides whether or not she has more than $1,000, he would fail.

This particular example does have a simple zero-knowledge protocol. Alice could give Bob $1,000.01, and that would be enough to convince Bob. However, this is zero-knowledge only if we assume that $0.01 is the minimum unit of currency available, because Bob knowing that Alice has more than $1,000 is the same as Bob knowing that Alice has at least $1,000.01. Yet, a direct solution like this one does not exist for most problems, and the goal of cryptographers is to build ZK protocols for as many cases as possible.

To make the discussion above more rigorous, we can consider the complexity class NP. A decision problem[2] is in NP, if for any YES instance of the problem, a prover that knows a proof (also called a witness) that the answer is YES is able to convince a polynomial-time verifier. In contrast, for any NO instance of the problem, any prover will fail to convince the verifier. In essence, NP is a class that formalizes and captures computational problems similar to problems like "Does this crossword puzzle have a solution?" or "Does Alice have more than $1,000 in the bank?" that have a YES/NO answer and for which a proof can be efficiently verified. *A priori* it is not guaranteed that an NP proof does not reveal more than a simple YES or NO to the verifier, i.e. the validity of the statement. For example, Alice could use her bank account password as a proof to Bob, but then Bob would learn the exact amount that Alice has in the bank. It is clear that this 'proof' would not be a ZK proof. In general, if Alice's only option is to provide a direct proof to Bob, she would not be able to guarantee that Bob gains zero-knowledge. The key to achieving ZK protocols is to allow interaction between P and V. In fact, Goldwasser, Micali, and Wigderson showed that every language in NP has an interactive ZK protocol [GMW86].

Interactive proofs. Before formally defining zero-knowledge protocols, we need to understand the concept of interactive proofs that was introduce by Goldwasser, Micali, and Rackoff also in [GMR85]. The decision problems found in NP can be solved non-interactively (one-way communication) by definition. In the YES instance case, there exists a proof that P can send to V, and there is no need for additional communication. An interactive proof is a generalization of this scenario to the case where P and V are allowed multiple rounds of interaction. If they take turns exchanging messages a total of k times[3], we say that P and V are executing a k-round protocol. In this terminology, NP contains all the decision problems that have a 1-round protocol. We formalize the definition of an interactive proof below.

Definition 1 (Interactive Proof System [GMR85])**.** *A prover P and a PPT verifier V are said to be performing an* **interactive proof system** *for a language $\mathcal{L}$ with completeness c and soundness s if:*

[2] A problem whose answer is either YES or NO.

[3] We always assume that P sends the k^{th} message (last message).

- **Completeness**: *For every $x \in \mathcal{L}$, it holds that*

$$\Pr\left[1 \leftarrow \langle P, V(x)\rangle\right] \geq c$$

- **Soundness**: *For every $x \notin \mathcal{L}$, and for any prover P^*, it holds that*

$$\Pr\left[1 \leftarrow \langle P^*, V(x)\rangle\right] \leq s$$

Where $\langle P, V(x)\rangle$ denotes V's output after interacting with P on input x.

While interactive proofs capture all the possible efficient (polynomial sized) protocols between P and V, interactive proofs are not inherently cryptographic. In other words, interactive proofs do not necessarily provide any sort of security or privacy to either of the parties besides completeness and soundness. A zero-knowledge proof system (or ZK protocol) is an interactive proof system with the additional cryptographic property called zero-knowledge. When we say that a protocol is a zero-knowledge, we mean that P has the guaranteed that, after the execution of the protocol, V did not learn anything else besides whether the answer to the problem is YES or NO. The best way to formalize the zero-knowledge property is with the simulation paradigm.

The simulation paradigm [Lin17, Gol07, Gol09]. Many security notions in cryptography (e.g. semantic security, multi-party computation, ZK) can be stated using the 'Real/Ideal' world paradigm, which is best understood by example. Alice wants to encrypt a one-bit message, $m \in \{0,1\}$, using a particular encryption scheme, $(\mathsf{Enc}, \mathsf{Dec})$, and would like to make sure that Bob cannot recover m. We can use the 'Real/Ideal' world paradigm to define what it means for the encryption to be secure. In the real world, Alice samples a random secret key, sk, and computes a ciphertext, $c \leftarrow \mathsf{Enc}_{\mathsf{sk}}(m)$, which is then sent to Bob. In an ideal world, when Bob asks Alice for an encryption of m, Alice would encrypt a random bit $b \in \{0,1\}$ instead. We can say that the encryption scheme is secure if Bob cannot distinguish whether he is in the real world or if he is in the ideal world. This is because in the ideal world, m is completely hidden from Bob – the encryption Alice sends is independent of m. Indistinguishability of ideal and real worlds would guarantee that m is also hidden from Bob in the real world (otherwise he could distinguish the two worlds).

The ideal world is an example of *simulation*. In fact, we could rephrase our example as follows: the encryption scheme is said to be secure if there exists a simulator that, without knowing m, simulates Alice's behavior such that Bob cannot distinguish Alice from the simulator. In the scenario above, the simulator generates the ciphertext of a random bit. If Bob cannot distinguish Alice from the simulator, the message m is hidden from him.

The same simulation concept, albeit in the more complex interactive setting, is used to define zero-knowledge. Consider any NP language, $\mathcal{L} \in \text{NP}$. In the real world, Alice and Bob exchange messages while executing a particular ZK protocol, Π_{ZK}. In an ideal world, if $x \in \mathcal{L}$, Alice could give a witness w to a trusted third party, who in turn would tell Bob that "Alice proved to me that $x \in \mathcal{L}$ ". This ideal

scenario captures exactly what we would like from a zero-knowledge protocol – all Bob learns is that someone he trusts tells him that $x \in \mathcal{L}$. Unlike in the encryption example above, Bob has a simple way to distinguish whether he is in the real or in the ideal world. In the real world, he had to engage in an interactive protocol, but in the ideal world someone just told him that $x \in \mathcal{L}$. In this case, we cannot get away by arguing that the messages he received in the real world are indistinguishable from the messages he received in the ideal world; however, it suffices to argue that anything that Bob can compute in the real world, he can also compute in the ideal world. In other words, that anything that Bob can compute by interacting with Alice in the real world, he can compute if he just trusts that $x \in \mathcal{L}$. We say that the interactive protocol Π_{ZK} is zero-knowledge if Bob in the ideal world can simulate the Bob from the real one. If by Bob's knowledge we mean 'anything that Bob can compute', then all the knowledge he gains from his interaction with Alice, he also gains if he is *only* told that $x \in \mathcal{L}$. This justifies the term zero-knowledge, as he doesn't gain any more knowledge by performing the protocol Π_{ZK} with Alice if he already knows that $x \in \mathcal{L}$.

What does it mean for Bob in the ideal world to simulate Bob from the real world? We mean that Bob in the ideal world (the simulator) outputs samples from a distribution that is either perfectly, statistically or computationally indistinguishable from the distribution that Bob outputs in the real world. If such a simulator exists, we say that the protocol is perfect, statistical, or computational zero-knowledge respectively. From now on, unless otherwise stated, zero-knowledge will mean computational zero-knowledge.

Before formally defining ZK proofs, there are a few points worth mentioning. First, we have implicitly assumed that the simulator (Bob in the ideal world) and the verifier (Bob in the real world) have the same computational complexity – that they are probabilistic polynomial-time (PPT) machines. In contrast, the original ZK definition given by Goldwasser, Micali, and Rackoff allows the simulator to run in expected polynomial-time instead. While this is not necessary, it is convenient as it allows the construction of simpler protocols[4].

Second, while there are no restrictions on the computational power of the prover (honest and malicious provers are allowed to be unbounded) in the definition of interactive proofs, for practical purposes it is more relevant for the honest parties involved in a protocol to be efficient. For example, while the soundness guarantee can still hold against unbounded malicious provers, we would like the honest prover to run efficiently. In order for this to make sense, we have to assume that the honest prover is given a witness as auxiliary-information. Without the auxiliary-information, an efficient P might not be able to convince V over NP instances unless NP problems can be solved by PPT machines. If we also restrict the computational power of the malicious provers by requiring soundness to only hold against malicious PPT provers, the resulting protocol is called a *argument system* instead of a proof system.

Third, in cryptography we consider PPT machines that also have *auxiliary* inputs. This is needed because protocols are not necessarily being performed in isolation. The

[4]Constant round and black-box ZK protocols for NP can be constructed with expected PPT simulators, but not with strict polynomial-time simulation [BL04].

parties involved in a protocol might have additional information before engaging in the protocol, so we generally assume that parties are non-uniform PPT machines.

Finally, as discussed before, the goal of Bob in the ideal world (the simulator) is to simulate the output of Bob in the real world. While Bob in the real world does play the role of V in the protocol, he can also privately compute anything he wants. Any such computation done, the simulator has to mimic. For this reason, we always assume that V has a private state, and that its output at the end of the computation is not just the decision 'accept' or 'reject'. Whatever V outputs besides accepting or rejecting we call the view of V in the protocol (denoted by View_V). Without loss of generality, we can assume that the view includes the transcript of the protocol as well as the verifier's private state at the end of the protocol.

We present the following definition with these modifications incorporated.

Definition 2 (Zero-Knowledge Proof (Argument) System for NP [GMR85])**.** *Let $\mathcal{L} \in$ NP, and $\mathcal{R}(\mathcal{L})$ be the associated* NP *relation*[5]*. A PPT prover P and a PPT verifier V are said to be performing a* **zero-knowledge proof (resp. argument) system** *for $\mathcal{L}$ with completeness c and soundness s if all the following hold.*

- **Completeness***: For every $(x, w) \in \mathcal{R}(\mathcal{L})$, it holds that*

$$\Pr\left[1 \leftarrow \langle P(x,w), V(x)\rangle\right] \geq c$$

- **Soundness (resp. computational soundness)***: For every $x \notin \mathcal{L}$, and for any prover (resp. PPT prover) P^*, it holds that*

$$\Pr\left[1 \leftarrow \langle P^*(x), V(x)\rangle\right] \leq s$$

- **Computational Zero-Knowledge***: For every $x \in \mathcal{L}$, for any auxiliary input $z \in \{0,1\}^*$, and for any PPT verifier V^*, there exists a probabilistic machine (the simulator),* Sim*, running in expected polynomial-time in $|x|$ such that the following holds:*

$$\mathsf{Sim}(V^*, x, z) \approx_c \mathsf{View}_{V^*}\langle P(x,w), V^*(x,z)\rangle$$

where $\approx_c$ denotes computationally indistinguishable. That is, for every PPT distinguisher D (polynomial-time in $|x|$), the following holds:

$$\left|\Pr\left[D\left(x, z, \mathsf{Sim}\left(V^*, x, z\right)\right) = 1\right] - \Pr\left[D\left(x, z, \mathsf{View}_{V^*}\langle P(x,w), V^*(x,z)\rangle\right) = 1\right]\right| \leq \mathit{negl}(|x|)$$

where negl is a negligible function.

Remark 3. *We say that a simulator* Sim *is a* black-box *simulator if it only uses V^* as a subroutine. In other words, it only needs input-output (query) access to the next message function of V^*. If* Sim *uses the code of V^* and not just its input-output behavior, we say that* Sim *is a* non-black-box *simulator.*

[5] $x \in \mathcal{L}$ iff there exists w s.t. $(x, w) \in \mathcal{R}(\mathcal{L})$.

The rewinding technique. *Rewinding* is the main technique used in the classical setting to construct ZK simulators. Unlike the prover, the simulator can rewind the verifier. Recall that the simulator is supposed to capture computation that Bob, who is playing the role of V in the actual ZK protocol, could have done by himself (without interaction with P), and Bob can execute the protocol all by himself by pretending to also be P. Indeed, the role of the simulator is to run a pretend protocol, in a way that is indistinguishable from the actual protocol. But if it is a pretend protocol, the simulator is playing the role of both P and V in the protocol, which means that it can decide to restart the pretend protocol from a previous round (rewind). In contrast, P in the actual protocol can't ever ask V to go back. The computational asymmetry between the simulator and the prover that rewinding provides is enough to construct ZK protocols.

Example: ZK protocol for Graph Isomorphism. The protocol shown in Figure 1-1 is a ZK protocol for the Graph Isomorphism problem. Given two graphs G_0 and G_1, (G_0, G_1) is a YES instance if G_0 and G_1 are isomorphic, i.e. there is an isomorphism π such that $\pi(G_0) = G_1$. Otherwise, (G_0, G_1) is a NO instance.

Initial input of prover P: (G_0, G_1) and π if YES instance.
Initial input of verifier V: (G_0, G_1).

- $\underline{P \to V}$: P chooses a random isomorphism σ and a random bit $a \in \{0,1\}$ and sends $H = \sigma(G_a)$ to V.
- $\underline{V \to P}$: V responds with a random bit $b \in \{0,1\}$.
- $\underline{P \to V}$: P computes and sends an isomorphism ϕ such that $\phi(G_b) = H$.
- V accepts if $\phi(G_b) = H$; otherwise, it rejects.

Figure 1-1: ZK protocol for Graph Isomorphism

If (G_0, G_1) is a YES instance, then the prover can always compute ϕ because H is isomorphic to both G_0 and G_1, hence the verifier will always accept (completeness). If (G_0, G_1) is a NO instance, the prover will only be able to convince the verifier to accept when $a = b$, which happens with probability $\frac{1}{2}$ (soundness).

To prove that the protocol is zero-knowledge consider the following simulator. Let V^* be any malicious verifier.

$\mathsf{Sim}(V^*, G_0, G_1)$**:**

1. Choose a random isomorphism σ and a random bit $a \in \{0,1\}$. Set $H = \sigma(G_a)$.

2. Compute $b \leftarrow V^*(G_0, G_1, H)$.

3. If $a = b$, output (H, b, σ).

4. If $a \neq b$, restart from Step 1.

Suppose that (G_0, G_1) is a YES instance, then H is independent of a. From this, we have that V^* can only output a with probability $\frac{1}{2}$, i.e. that

$$\Pr\left[b = a : \begin{array}{c} a \xleftarrow{\$} \{0,1\} \\ H \leftarrow \sigma(G_a) \\ b \leftarrow V^*(G_0, G_1, H) \end{array}\right] = \frac{1}{2}.$$

This means that the simulator runs in expected polynomial-time, as it will rewind once in expectation. Finally, to show that the protocol is (perfect) zero-knowledge, we have to argue that the output distribution of Sim is exactly the same as the view of V^*. To see why this is the case, note that the decision to rewind in Step 4 does not depend on either H or b, because the probability that P computed H from G_b is $\frac{1}{2}$ regardless of H and b. But this just means that the triplet (H, b, σ) that Sim computes in Step 3 is sampled from the same distribution whether there was rewinding or not; furthermore, (H, b, σ) is distributed the same as the view of the verifier.

Remark 4. *The simulator described above is an example of a black-box simulator.*

Intuitively, the protocol from Figure 1-1 is ZK because when (G_0, G_1) is a YES instance, it is not possible to determine whether H was computed from G_0 or from G_1. Furthermore, the isomorphism ϕ is randomly distributed as it is either σ, $\sigma \circ \pi$, or $\sigma \circ \pi^{-1}$. This means that the bit b computed by V is independent from the bit a computed by either P or Sim. As a consequence, V cannot distinguish whether $a = b$ or $a \neq b$. In the real world, P answers regardless of whether $a = b$ or $a \neq b$. Now consider an alternate world where P has magical powers and can predict the future, but it does not know an isomorphism between G_0 and G_1. This magical prover can predict what b will be and set $a = b$. Since V cannot distinguish the case when $a = b$ from the case $a \neq b$, it can't distinguish whether it is interacting with the real prover from when it is interacting with the magical prover that always guesses b ahead of time. Because the magical prover never used an isomorphism from G_0 to G_1 (it doesn't even know one), V did not learn an isomorphism from G_0 to G_1 either. While this argument shows why V did not learn an isomorphism between G_0 and G_1, it falls shorts in arguing that V didn't learn *anything*. After all, we just argued that, by interacting with P, it learns as much as it would have learn if it interacted with the magical prover with supernatural foresight! Fortunately, the ability to predict the future is not needed to behave as the magical prover would. A simulator can try to guess b, and if it fails, it can restart the protocol. Conditioning on it obtaining $a = b$, which happens on expectation after two tries, it outputs messages that are indistinguishable from messages that the magical prover would have sent. The conclusion is that by interacting with P, the verifier V learned as much as it would have learned if it interacted with this final simulator. The key idea behind zero-knowledge is that this final simulation – guessing b when computing a, outputting

the resulting execution if $a = b$, and restarting otherwise – is something that V could have done by itself.

Extraction via rewinding. One way in which rewinding is used to construct simulators is to rewind V in order to extract secrets from it. Consider the useless but instructive protocol shown in Figure 1-2. Any party playing the role of P in this protocol would not be able to retrieve sk. Regardless of whether it asks for b_0 or for b_1, the response will be a uniformly random bit. On the other hand, a simulator that is allowed to rewind V can first ask for b_0, restart the protocol, ask for b_1, and recover $\mathsf{sk} = b_0 \oplus b_1$.

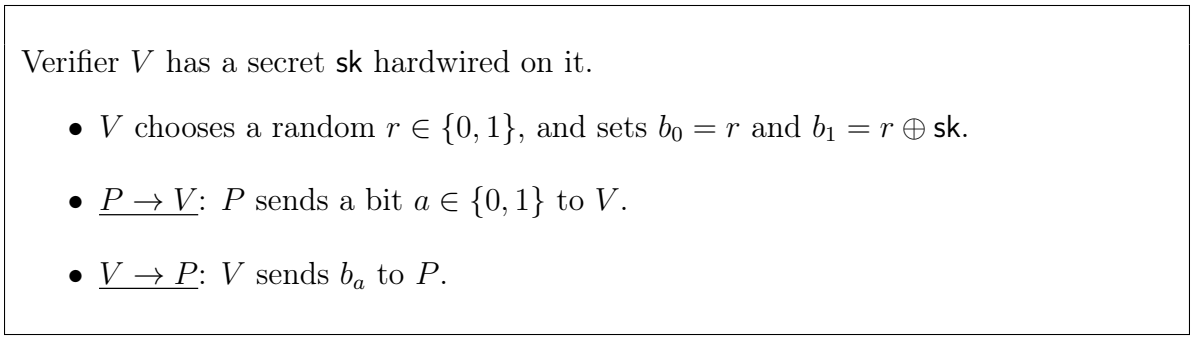

Figure 1-2: Extraction example

Why would we want a simulator to extract secrets from V? Leveraging knowledge extraction in order to construct ZK protocols goes back to the work of Feige, Lapidot and Shamir [FLS99]. At a high level, the more we know about V's behavior, the more we can say about what it's learning or not. While the simulator is given access to V in order to run a pretend protocol, this does not guarantee it has enough information about V for us to argue that V does not learn anything. For example, V could be an obfuscated program or a black-box that only allows access to its input-output behavior. In such an instance, even having access to the code of V, there is no way to guarantee that information like a hard-coded secret sk can be found. In order to say anything at all about what V learned, it is useful to extract additional information about V (from just its input-output behavior). The simulators for the different ZK protocols presented in this thesis extract secrets from V in one form or another.

Extraction and proofs of knowledge. While extraction is not always necessary for ZK simulators, it is necessary to show that some interactive protocols satisfy a desirable and stronger soundness[6] guarantee called *proof of knowledge* (PoK) [GMR85, BG92]. Proofs of knowledge are widely used in the construction of more advanced primitives. Informally, an interactive proof satisfies the PoK property if for any P that convinces V to accept, there is a simulator (with access to P this time) that can extract a proof/witness from P. In some sense, this property says that if P gets

[6] PoK supersedes soundness.

V to accept in the protocol, then P must know a witness. Unlike ZK simulators, simulators that extract are inherent in the definition of PoK.

1.1.2 Challenges in the quantum setting

There are additional hurdles present when studying ZK protocols in the quantum setting. In this setting, it is assumed that a malicious verifier has access to a quantum computer. Since quantum computers are believed to be computationally more powerful than classical computers, this means that we cannot rule out the possibility that a malicious quantum verifier engaging in a classical ZK protocol learns more than it should, i.e. breaks the zero-knowledge property. Similar to the classical zero-knowledge property discussed before, if quantum adversaries do not learn anything from a protocol except for the validity of the statement being proven, we say that the protocol is *quantum zero-knowledge* (QZK) [Wat09]. The formal definition of QZK is a direct quantum analogue of classical ZK (see Section 2.2.1), and the biggest difference is that malicious verifiers are now quantum polynomial-time (QPT). As a consequence, the QZK simulators are also QPT.

The biggest challenge in constructing QZK protocols is that rewinding does not always work in the quantum setting. While classically the simulator can take a snapshot of the state of the computation/protocol at a particular point in time, and then return back to it later on as needed, this is impossible in the quantum setting due to the No-Cloning Theorem. Furthermore, if a quantum simulator measures a message sent by V, it could perturb the private state of V, thereby also eliminating the possibility to rewind to a previous stage of the computation. This is not to say that quantum rewinding is impossible, but that we have to be more careful. Watrous [Wat09] was the first person to introduce and to study the notion of quantum zero-knowledge, as well as to point out sufficient conditions for quantum rewinding to work. While not as general as rewinding in the classical setting, Watrous' rewinding has proven to be quite useful. Besides Watrous' work, there have been many works [ARU14, BJSW16, BG20, BS20, ALP20, VZ20, ABG+20] that consider the notion of quantum zero-knowledge, and most of them use Watrous' rewinding in one way or another.

Quantum Proofs of Knowledge. Quantum rewinding also presents a challenge when studying PoK's in the quantum setting. Unruh [Unr12] showed how to use rewinding in specific cases to construct *quantum proofs of knowledge* (QPoK) albeit with a few restrictions that limit their *composability*, their usage as an ingredient in other protocols. In particular, Unruh's QPoK satisfies a weak version of *extractability* – the probability that the extractor succeeds is not negligibly close to the acceptance probability – and more importantly, it does not satisfy *simulatability* – the prover's state after extraction is not statistically close to the prover's state after interacting with the actual verifier. There have been other works that present constructions that satisfy both the above conditions, but the extraction is only against computationally bounded adversaries [HSS11, BS20, ALP20]. It has been an important open

problem to design quantum proofs of knowledge (i.e. extraction against unbounded adversaries) satisfying both of the above conditions.

1.1.3 Questions explored in this thesis

Much of the ZK work in the classical setting aims to construct ZK protocols with additional properties beyond the basic ones or to optimize their computational resources like their round complexity. Protocols with minimal round complexity and negligible soundness are desirable for practical reasons. Given a ZK protocol, its soundness can be improved by sequential repetition, but this incurs a polynomial overhead in the number of rounds. Optimal round complexity for protocols with negligible soundness have already been studied in the classical setting [BCPR16, BBK+16, BKP18, BKP19]; meanwhile, little is known about the round complexity of quantum zero-knowledge[7]. This motivates the following question:

> ***(Q1.1)*** *Are there constant round QZK protocols for* NP *(with negligible soundness)?*

Besides optimal round complexity, another desirable feature we might want our ZK protocols to have is *concurrency*. The basic ZK property guarantees security against a single verifier, but it says nothing about the security of the protocol in the more realistic scenario when a prover P performs the protocol concurrently with multiple verifiers, $V_1, \ldots, V_n$. While *concurrent zero-knowledge* can still be defined using the simulation paradigm, the simulators of concurrent protocols are more elaborate as they have to account for new challenges like the possibility that the multiple verifiers are colluding with each other and taking turns engaging with P in whatever order they want. Concurrency has been widely studied in the classical literature [DS98, DCO99, Can01, CLOS02, CF01, RK99, BS05, DNS04, PRS02, Lin03, Pas04, PV08, PTV14, GJO+13, CLP15, FKP19], but none of those simulators or proof techniques seem quantum friendly. Thus, constructing a concurrent QZK protocol seems to require new techniques, which leads us to the next question:

> ***(Q1.2)*** *Are there concurrent QZK protocols for* NP*?*

Both **(Q1.1)** and **(Q1.2)** arise in part due to the difficulty of rewinding in the quantum setting, which makes it trickier to construct QZK simulators. This affects the construction of not only QZK protocols, but also QPoK protocols, where an extractor has to be exhibited. As discussed in the previous subsection, Unruh has shown a QPoK protocol for NP. However, his protocol does not satisfy simulatability – it does not compose well. It has been an important open problem to design quantum proofs of knowledge satisfying both of the above conditions.

> ***(Q1.3)*** *Are there quantum proofs of knowledge for* NP *that compose well? I.e. are there QPoKs for* NP *that satisfy extractability and simulatability?*

In this work, we make significant progress towards answering these three questions. We present our results in Section 1.3.

[7] In a recent and concurrent work [BS20], constant round QZK arguments for NP and for QMA were given using similar techniques as those presented in this thesis.

1.2 Quantum copy-protection

One of the main results in this thesis is that extraction techniques developed in the context of zero-knowledge can be used to answer a longstanding open question about the feasibility of quantum copy-protection. Quantum copy-protection was introduced by Aaronson [Aar09] to exploit the No-Cloning Theorem in order to formally study and tackle the problem of software piracy. Roughly speaking, quantum copy-protection says that given a quantum state computing a function f, the adversary cannot produce two quantum states (possibly entangled) such that each of the states individually computes f. While ad hoc solutions exist in the real world, achieving this security guarantee against piracy is impossible from classical software alone – any classical code that computes f can be copied, even if the code is obfuscated.

Quantum copy-protection is one of the "best" quantum cryptographic primitive we could dream to obtain using the No-Cloning Theorem. It would let us embed *functionality* in quantum states allowing us to make any functionality unclonable. For example, by copy-protecting a decryption circuit $\mathsf{Dec}_{\mathsf{sk}}$, we could obtain unclonable decryption circuits where only a single party is guaranteed to be able to decrypt messages being broadcasted. Quantum mechanics would not only give us unclonable states, but unclonable states that can be functionally used!

We present the formal definition as given by Aaronson.

Definition 5 (Quantum Copy-Protection [Aar09])**.** *Let $\mathcal{F}_n$ be a family of Boolean functions $f : \{0,1\}^n \to \{0,1\}$, where each $f \in \mathcal{F}_n$ is associated with a unique "description" $d_f \in \{0,1\}^m$. A* **quantum copy-protection** *is a tuple of QPT algorithms $(\mathsf{Vendor}, \mathsf{Run})$ as follows:*

- $\mathsf{Vendor}(d_f)$*: takes as input the description d_f of a function $f \in \mathcal{F}_n$ and outputs a state ρ_f.*
- $\mathsf{Run}(\rho_f, x)$*: takes as input a state ρ_f and x and attempts to output $f(x)$.*

Correctness: *We say that $(\mathsf{Vendor}, \mathsf{Run})$ satisfies ε-**correctness** if for all $f \in \mathcal{F}_n$ and all $x \in \{0,1\}^n$,*

$$\Pr_{\rho_f \leftarrow \mathsf{Vendor}(d_f)} [\mathsf{Run}(\rho_f, x) = f(x)] \geq 1 - \varepsilon.$$

Security: *Let $\mathcal{D}$ be a distribution over $\mathcal{F}_n \times \{0,1\}^n$. We say that $(\mathsf{Vendor}, \mathsf{Run})$ satisfies δ-security against $\mathcal{D}$ if for any QPT algorithms $\mathcal{P}$ and $\mathcal{R}$ and any $k, r = \mathrm{poly}(n, m)$, the following holds.*

- $\mathcal{P}(\rho_f^{\otimes k})$*: outputs a state σ_f on $k + r$ registers.*
- *For all $i \in [k + r]$, let $\sigma_f^i = Tr_{\bar{i}}(\sigma_f)$. In other words, σ_f^i is the state on the i^{th} register.*

- *For all $i \in [k+r]$, let X_i be an indicator random variable defined as follows:*

$$\mathsf{X}_i = \begin{cases} 1, & \text{if } \mathcal{R}(\sigma_f^i, x) = f(x) \\ 0, & \text{otherwise} \end{cases}$$

Then, averaging over (f, x) drawn from $\mathcal{D}$, we have that

$$\mathop{\mathbb{E}}_{(f,x)\leftarrow\mathcal{D}} \left[\sum_{i=1}^{k+r} \mathsf{X}_i \right] \leq k + (1-\delta)r.$$

Remark 6. *Aaronson [Aar09] points out some subtleties encountered when defining copy-protection which partly explain why he introduced the notion as in the definition above. Recent work [ALL+20] introduces a more general definition based on the projective implementation framework introduced by Zhandry [Zha20]. Our impossibility result applies to both of these notions.*

Unlearnable functions. As Aaronson pointed out, copy-protection only makes sense for *unlearnable* functions. If a description of a function f can be learned from its input-output behavior, then f cannot be copy-protected. Any pirate that is given a quantum state that successfully computes f a polynomial number of times, can learn a description for f and make as many copies from this description as it wants. While Aaronson considered a more relaxed definition of quantum unlearnability, we present here the definition of quantum unlearnability that we will be using in this thesis.

Definition 7 (Quantum unlearnability)**.** *Let $\mathcal{C}$ be a family of Boolean circuits associated to a distribution $\mathcal{D}_\mathcal{C}$. We say that $(\mathcal{C}, \mathcal{D}_\mathcal{C})$ is ν-**quantum unlearnable** with security parameter λ if for any QPT adversary $\mathcal{A}$, the following holds:*

$$\mathsf{Pr}\left[\forall x, \mathsf{Pr}[U^*(\rho^*, x) = C(x)] \geq \nu \ : \ \begin{matrix} C\leftarrow\mathcal{D}_\mathcal{C} \\ (U^*,\rho^*)\leftarrow\mathcal{A}^{C(\cdot)}(1^\lambda) \end{matrix}\right] \leq negl(\lambda)$$

In [Aar09], Aaronson proved that there is a copy-protection scheme for every unlearnable function relative to a quantum oracle. He also gave two heuristic candidate schemes to copy-protect point functions. More recently, Aaronson, Liu, Liu, Zhandry, and Zhang [ALL+20] improved upon the quantum oracle result obtaining a similar result but using a classical oracle instead. At the time the work presented in this thesis was done, despite a decade since the introduction of copy-protection, these were the only known results regarding copy-protection. The following question remained open:

> ***(Q2.1)*** *Is there a quantum copy-protection scheme for **every** quantum unlearnable function in a standard model (without oracles)?*

In this work we will show that the answer is a (conditional) no. Under reasonable cryptographic assumptions, there are families of unlearnable functions that cannot be quantum copy-protected.

While it is impossible to copy-protect every unlearnable function, we can ask whether we can copy-protect restricted families of functions, e.g. point functions.

> ***(Q2.2)*** *Is there a quantum copy-protection scheme for* ***some*** *quantum unlearnable functions in a standard model (without oracles)?*

Unfortunately, constructing provable copy-protection schemes in the standard model remains challenging even for restricted families. In this thesis, we make partial progress in this direction by showing that if we weaken the notion of copy-protection, then we can achieve a provable construction for restricted families. This new notion is called *Secure Software Leasing (SSL)*, and while weaker than copy-protection, it still captures the spirit of using the No-Cloning Theorem to tackle software piracy.

1.3 Our results

In this thesis, we use new simulation techniques to construct new QZK protocols providing partial answers to **(Q1.1)**, **(Q1.2)**, and **(Q1.3)** discussed in Section 1.1.3. The ideas developed are then adapted to the setting of quantum copy-protection in order to (conditionally) answer **(Q2.1)** from Section 1.2 in the negative. Finally, we introduce a new quantum cryptographic notion, Secure Software Leasing (SSL), and show how to construct it for a restricted family of circuits making partial progress towards answering **(Q2.2)**. While SSL is weaker than quantum copy-protection, this construction is the first provably secure construction in a standard model in the topic of quantum copy-protection.

1.3.1 Results regarding QZK protocols

Recall that our goal is to study classical protocols that hold their security guarantees even against malicious quantum adversaries. We desire protocols that can be implemented today, but will remain secure in the future if quantum computation becomes a widespread reality.

Remark 8. *All the QZK protocols for* NP *presented in this thesis are classical protocols. They can be performed by purely classical parties.*

A summary of the classical protocols (with quantum security) constructed in this thesis as well as the new ideas behind the constructions are shown in Table 1.1 and Table 1.2.
We describe our results in more detail below.

#1 Constant round QZK classical argument system for NP

Our first construction is a QZK classical argument system, i.e. its soundness guarantee only holds against bounded classical (PPT) provers. Ideally we want a construction that guarantees security against unbounded provers (or even against quantum provers), but by considering classical argument systems we can construct a protocol

Protocol	Tools	Assumption	Rounds
QZK classical argument (Section 4.2)	•Trapdoor claw-free functions •Secure function evaluation •Perf. binding commitments •WI arguments	QLWE	$O(1)$
Non-black-box quantum extraction (Section 3.3)	•Lockable obfuscation •QFHE •Perf. binding commitments •Secure function evaluation	QLWE QFHE	$O(1)$
Bounded concurrent QZK (Section 4.3)	•Stat. binding commitments •WI proofs	OWF	$O(n^c)$
Bounded concurrent QZK with QPoK (Section 5.3)	•Stat. receiver-private OT •Bounded concurrent QZK	QLWE	$O(n^c)$
Stat. receiver-private OT (Section 5.2)	•Stat. sender-private OT •Stat. ZK quantum argument	QLWE	$O(n^c)$

Table 1.1: Summary of classical protocols

Protocol	New Simulation or Extraction Mechanism
QZK classical argument (Section 4.2)	Simulators succeed if they can pass a 'test of quantumness'
Non-black-box quantum extraction (Section 3.3)	Non-black-box use of QFHE to obtain encryption of secret followed by decryption using lockable obfuscation
Bounded concurrent QZK (Section 4.3)	Block rewinding
Bounded concurrent QZK with QPoK (Section 5.3)	Combine block rewinding with statistical receiver-private OT
Stat. receiver-private OT (Section 5.2)	Use Watrous' rewinding to argue post-quantum sender security

Table 1.2: Summary of new simulation methods

with a few interesting features. First, the protocol solely relies on the QLWE assumption. Second, it satisfies *quantum-lasting* security. Quantum-lasting security [Unr13] is the guarantee that a classical protocol executed today that is secure against classical adversaries remains secure if an adversary obtains a quantum computer long after the execution of the protocol. In other words, quantum-lasting security says that transcripts of protocols executed today in a classical world would still be secure in a quantum future. Third, the protocol satisfies black-box QZK– it is the first constant round black-box QZK protocol based solely on QLWE. Finally, the protocol is

conceptually interesting as it is an application of *tests of quantumness*.

Theorem 9 (Constant Round Quantum ZK with Classical Soundness; Informal Version of Lemma 69)**.** *Assuming quantum hardness of learning with errors (QLWE), there exists a constant round black-box quantum zero-knowledge system with negligible soundness against classical PPT algorithms.*

Application: Authorization with Quantum Cloud. Soundness holding only against classical malicious provers is restricting, but as the following example shows, there are applications for such protocols in the near-term of quantum computation where we expect only a few big players to have access to quantum resources. Suppose Eve wants to convince the IBM cloud service that she has the authorization to access a document residing in the cloud. Since the authorization information could leak sensitive information about Eve, she would rather use a zero-knowledge protocol to prove to the cloud that she has the appropriate authorization. While we currently don't have scalable implementations of quantum computers, this could change in the future when organizations like IBM could be the first ones to develop a quantum computer. They could in principle then use this to break the zero-knowledge property of Eve's protocol and learn sensitive information about her. In this case, it suffices to use a QZK protocol but only requiring soundness against malicious classical users; it is reasonable to assume that even if IBM gets to develop a full-fledged quantum computer, in the nearby future, it'll take a while before every day users will have access to one. Everyday users can then take the role of the provers in protocols where soundness is only guaranteed against classical parties.

#2 Non-black-box quantum extraction

The underlying idea behind the construction of Theorem 9 is to first construct an extraction protocol (defined in Chapter 3), and then leverage it to obtain full-fledged QZK. While we fall short of carrying the same program to obtain a QZK argument system with soundness against bounded quantum provers, we show how to achieve a constant round extraction protocol against such provers. To do this, we introduce a new non-black-box extraction technique in the quantum setting building upon a classical non-black-box extraction technique of [BKP19].

Theorem 10 (Non-black-box quantum extraction; Informal Version of Lemma 63)**.** *Assuming quantum hardness of learning with errors (QLWE) and a quantum fully homomorphic encryption scheme (QFHE) (for arbitrary poly-time computations)*[8]*, 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 receivers.*

We clarify what we mean by perfect correctness. For every public key, every valid fresh ciphertext of a classical message can always be decrypted correctly. Moreover,

[8] As against leveled quantum FHE, which can be based on QLWE.

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, Bra18] give candidates for quantum fully homomorphic encryption schemes satisfying the properties needed for Theorem 10.

We view identifying the appropriate classical non-black-box technique to also be a contribution of our work. *A priori* it should not be clear whether classical non-black-box techniques are useful in constructing their quantum analogues. For instance, it is unclear how to utilize the well known non-black-box technique of Barak [Bar01]; at a high level, the idea of Barak [Bar01] 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.

Non-black-box extraction overcomes the disadvantage quantum rewinding poses in achieving constant round extraction; the quantum rewinding employed by [Wat09] requires polynomially many rounds (due to sequential repetition) or constant rounds with non-negligible gap between extraction and verification error [Unr12].

This technique was concurrently developed by Bitansky and Shmueli [BS20] (see Section 1.5) and they critically relied upon this to construct a constant-round zero-knowledge argument system for NP and QMA, thus resolving a long-standing open problem in the round complexity of quantum zero-knowledge.

#3 Bounded concurrent QZK proof systems for NP and QMA

In our next QZK protocol, we initiate a formal study of concurrent composition in the quantum setting. In this setting, the prover P can interact with many verifiers $V_1, \ldots, V_m$ while preserving the ZK property. Security is expected to hold against a malicious quantum adversary V^* who controls the behavior of any subset of the verifiers. This adversary can entangle the private states of all the verifiers, as well as decide in which order the verifiers will send their messages (decide their *scheduling*). As we will discuss in Chapter 4, quantum rewinding these type of adversaries is harder than in the standalone setting because the rewinding cannot depend on the scheduling of the verifiers. Nonetheless, we introduce the technique of *block rewinding*, which allows us to use Watrous' rewinding as long as we restrict ourselves to the *bounded* concurrent setting – the prover interacts only with a bounded number of verifiers where this bound is fixed at the time of protocol specification. This setting has been well studied in the classical concurrency literature [Lin03, PR03, Pas04, PTW09]. Moreover, we note that the only other existing work that constructs quantum zero-knowledge against multiple verifiers (in the parallel composition setting), namely [ABG+20], also works in the bounded setting.

Theorem 11 (Bounded concurrent QZK for NP; Informal Version of Theorem 77)**.** *Assuming the existence of post-quantum one-way functions*[9]*, there exists a bounded*

[9]That is, one-way functions secure against (non-uniform) quantum polynomial-time algorithms.

concurrent quantum zero-knowledge proof system for NP. *Additionally, our protocol is a public coin proof system.*

Our construction satisfies quantum black-box zero-knowledge. We note that achieving public-coin *unbounded* concurrrent ZK is impossible [PTW09] in the classical setting. The construction from Theorem 11 is similar from that of [PTW09]; however, we need to instantiate the protocol with different parameters and to construct a new QZK simulator.

While we see the construction from Theorem 11 as our main contribution in the concurrent setting, we also show how to obtain bounded concurrent QZK for QMA by following the framework introduced by Broadbent, Ji, Song, and Watrous (BJSW) [BJSW16] and using our bounded concurrent QZK protocol for NP.

Theorem 12 (Bounded concurrent QZK for QMA; Informal Version of Theorem 94)**.** *Assuming post-quantum one-way functions, there exists a bounded concurrent quantum zero-knowledge proof system for QMA.*

#4 Statistical receiver-private oblivious transfer

The main ingredient in our construction of quantum proofs of knowledge is statistical receiver-private oblivious transfer introduced in [GJJM20, DGH+20]. An oblivious transfer (OT) protocol is a protocol between a sender S and a receiver R. The input to S is a pair of messages (m_0, m_1), and the input to R is a bit b. The goal of OT is for R to obtain m_b with the following security guarantees. S does not want to reveal both of the messages, and R does not want S to learn which message it is retrieving, i.e. does not want to reveal b. For the purpose of obtaining QPoKs, we desire an OT protocol that is secure against computationally unbounded malicious senders and secure against malicious QPT receivers. The protocols presented in [GJJM20, DGH+20] are statistical receiver-private, that is, they are secure against unbounded senders. They are also secure against PPT receivers; however, the sender's security proofs do not directly work in the quantum setting. Starting from the ideas in [GJJM20, DGH+20], we show how to construct a statistical receiver-private OT with post-quantum security against receivers.

Theorem 13 (Post-quantum statistical receiver-private OT; Informal (Section 5.2.3))**.** *Assuming quantum hardness of learning with errors (QLWE), there is a statistical receiver-private oblivious transfer protocol that is secure against malicious QPT receivers.*

#5 Quantum proofs of knowledge

Our bounded concurrent construction only satisfies the standard soundness guarantee. A more desirable property is quantum proof of knowledge. Roughly speaking, proof of knowledge states the following: suppose a malicious (computationally unbounded) prover can convince a verifier to accept an instance x with probability ε. Let the state of the prover at the end of interaction with the verifier be $|\Psi\rangle$. Then there exists an

efficient extractor, with black-box access to the prover, that can output a witness w for x with probability δ. Additionally, it also outputs a quantum state $|\Phi\rangle$. Ideally, we require the following two conditions to hold: (i) $|\varepsilon - \delta|$ is negligible and, (ii) the states $|\Psi\rangle$ and $|\Phi\rangle$ are close in trace distance; this property is also referred to as simulatability property. Unruh [Unr12] presented a construction of quantum proofs of knowledge; their construction satisfies (i) but not (ii). Indeed, the prover's state, after it interacts with the extractor, could be completely destroyed. Condition (ii) is especially important if we were to use quantum proofs of knowledge protocols as a sub-routine inside larger protocols, for instance in secure multiparty computation protocols.

We prove the following theorem by combining our bounded concurrent QZK protocol with our construction of QPoK from statistical receiver-private OT.

Theorem 14 (Bounded concurrent QZK with QPoK; Informal (Section 5.3.3))**.** *Assuming the quantum hardness of learning with errors (QLWE), there exists a bounded concurrent quantum zero-knowledge proof system for NP satisfying quantum proofs of knowledge property.*

1.3.2 Results regarding quantum copy-protection

We show how the non-black-box extraction techniques, like those used in Theorem 10, introduced in seemingly different contexts – proving impossibility of obfuscation [BGI+01, BP16, AF16] and constructing zero-knowledge protocols [BKP19, BS20, ALP20] – are relevant to proving the impossibility of copy-protection.

To demonstrate our impossibility result, we identify a class of classical circuits $\mathcal{C}$ that we call a *de-quantumizable* circuit class. This class has the property that given *any* QPT U_C and any auxiliary state ρ_C such that for all x, it holds that $U_C(\rho_C, x) = C(x)$ for a circuit $C \in \mathcal{C}$, we can efficiently 'de-quantumize' to obtain a classical circuit $C' \in \mathcal{C}$ that has the same functionality as C. We call (U_C, ρ_C) an *efficient quantum implementation* of the circuit C. In other words, a de-quantumizable circuit is one for which there is no cryptographic advantage to providing a quantum implementation instead of a classical description of the circuit. If $\mathcal{C}$ is learnable then, from the definition of learnability, there could be a QPT algorithm that finds C'. To make the notion interesting and non-trivial, we add the additional requirement that this class of circuits is quantum unlearnable. Our main result is the existence of de-quantumizable circuits under cryptographic assumptions.

Proposition 15 (Existence of de-quantumizable circuits; Informal Version of Theorem 114)**.** *Assuming the quantum hardness of learning with errors (QLWE), and asssuming the existence of quantum fully homomorphic encryption*[10] *(QFHE), there exists a de-quantumizable class of circuits.*

It is easy to see why de-quantumizable circuits cannot be copy-protected. If a pirate is given any quantum implementation (U_C, ρ_C) of the circuit C, it can recover

[10]We need additional properties from the quantum fully homomorphic encryption scheme but these properties are natural and satisfied by existing schemes [Mah18a, Bra18].

a classical description of the circuit and copy it. We obtain the desired impossibility result as a corollary.

Corollary 16 (Impossibility of copy-protection; Informal Version of Corollary 119)**.** *Assuming the quantum hardness of learning with errors (QLWE), and asssuming the existence of quantum fully homomorphic encryption (QFHE), there exists a class of quantum unlearnable circuits $\mathcal{C}$ that cannot be quantum copy-protected.*

We will see that our impossibility result is actually stronger than Corollary 16. We show that even achieving the weaker notion that we will introduce, SSL, is impossible for arbitrary quantum unlearnable circuits. While the reason is similar, there is no SSL scheme for our family of de-quantumizable circuits, the formal argument is a bit more subtle. We defer this discussion until we have defined SSL in Chapter 7. Our strongest impossibility result is the following.

Theorem 17 (Impossibility of SSL; Informal Version of Theorem 131)**.** *Assuming the quantum hardness of learning with errors (QLWE), and asssuming the existence of quantum fully homomorphic encryption (QFHE), there exists a class of quantum unlearnable circuits $\mathcal{C}$ such that there is no SSL for $\mathcal{C}$.*

Impossibility of Quantum VBB with single unclonable state. One could hope to use quantum resources to obfuscate classical circuits. In some sense, quantum states are already obfuscated – if we do not know enough about a particular state, any measurement we do on it might destroy it. This means that we could hope to somehow embed a Boolean circuit C in a quantum state for the purpose of hiding the circuit, and perhaps achieve the strongest notion of circuit obfuscation there is, virtual black-box obfuscation (VBB). The notion of using quantum resources to achieve VBB was termed quantum virtual black-box obfuscation (QVBB) by Alagic and Fefferman [AF16]. They also showed how to extend classical impossibility results [BGI+01] to the quantum setting . However, the possibility of using a single unclonable state to achieve QVBB was left open. Our techniques also rule out the possibility of QVBB for classical circuits.

Proposition 18 (Impossibility of QVBB; Informal Version of Proposition 120)**.** *Assuming the quantum hardness of learning with errors (QLWE) and assuming the existence of quantum fully homomorphic encryption (QFHE), there exists a circuit class $\mathcal{C}$ such that any quantum VBB for $\mathcal{C}$ is insecure.*

1.3.3 New notion: Secure Software Leasing

While the impossibility result rules out copy-protection for arbitrary functions, it might still be possible to obtain copy-protection for restricted family of functions like point-functions or compare-and-compute families. Unfortunately, even for simple families like point-functions, we do not know how to construct provably secure quantum copy-protection in the standard model. This remains a challenging and interesting open problem. Nevertheless, we show that at least if we weaken the notion of quantum copy-protection, then there are constructions for restricted families.

Our first observation is that there are scenarios in which full-blown copy-protection might be an overkill. For example, Alice might want to *lend* a program C to Bob with the expectation that sometime in the future he will return it. Alice might want the security guarantee that Bob cannot keep a copy of the program after he returned the original copy. Another example would be the following scenario. Alice wants to pirate and sell (illegal) copies of a program C; however, not everyone is willing to buy illegal copies. Bob wants to buy an *authenticated* (or *official*) copy. In this scenario, it is enough to have the security guarantee that Alice cannot forge authenticated copies. She might be able to make pirated copies, but she will not be able to fool Bob into buying them. We introduce the notion of *Secure Software Leasing* (SSL) to capture these scenarios. The syntax of SSL is similar to copy-protection, except that the security guarantee only rules out the possibility of a pirate making two copies *both* authenticated that evaluate the protected circuit correctly under a fixed Run algorithm. We use the terminology of leasing, since we see Run as a proprietary fixed algorithm (like an operating system) for which an authority can lease software. Furthermore, the leasing terminology also lets us capture the case where the lessor requires the original software to be returned. We define two security notions associated to SSL, finite-term and infinite-term security, corresponding to whether the lessor expects the initial state back or not.

We show that there is an SSL scheme for a general class of evasive circuits[11]. We need a few properties from a circuit family $\mathcal{C}$ in order to be able to construct SSL. First, $\mathcal{C}$ has to be *searchable* – given a circuit $C \in \mathcal{C}$, it is possible to find an accepting input. For example, if $\mathcal{C}$ is a family of point-functions where each circuit is described by the accepting point. I.e. the string $z \in \mathcal{C}$ represents the Boolean circuit C_z where $C_z(x) = 1$ if and only if $x = z$. This would be a searchable class. We emphasize that searchability is a property of the description of the circuits. Furthermore, a description of a circuit C does not necessarily lets you find accepting inputs (e.g. evasive circuits). Searchability is a natural property and is implicit in the description of the existing constructions of copy-protection by Aaronson [Aar09]. Second, we need to be able to obfuscate the accepting inputs of $\mathcal{C}$. Specifically, we need an algorithm that takes as input a circuit $C \in \mathcal{C}$, and outputs a different description, $\widetilde{C}$, of the same circuit that does not reveal the accepting inputs. This notion is called input-hiding obfuscation [BBC+14]. If $\mathcal{C}$ is searchable, and it can be obfuscated with an input-hiding obfuscator, then we show how to construct SSL for $\mathcal{C}$.

Theorem 19 (SSL for General Evasive Circuits; Informal (Section 7.5))**.** *Let $\mathcal{C}$ be a searchable class of circuits. Assuming the existence of: (a) quantum-secure input-hiding obfuscators [BBC+14] for $\mathcal{C}$, (b) quantum-secure subspace obfuscators [Zha19] and, (c) learning with errors (QLWE) secure against sub-exponential quantum algorithms, there exists an SSL scheme in the common reference string model for $\mathcal{C}$.*

Remark 20. *The common reference string (CRS) model means that the honest parties have access to a common input produced by a trusted setup. The impossibility result also holds in this model.*

[11] Boolean circuits for which it is hard to find an accepting input.

A class of circuits that satisfies our requirements is searchable compute-and-compare circuits. A circuit $D_{C,\alpha}$ in this class, parametrized by a circuit C and a lock α, is defined as follows:

$$D_{C,\alpha}(x) = \begin{cases} 1 & \text{if } C(x) = \alpha \\ 0 & \text{otherwise} \end{cases}$$

This circuit class has been studied in the cryptography literature in the context of constructing program obfuscation [WZ17, GKW17]. Restricting Theorem 19 to compute-and-compare circuits, we obtain the following result.

Theorem 21. *Assuming the existence of: (a) quantum-secure subspace obfuscators [Zha19] and, (b) learning with errors (QLWE) secure against sub-exponential quantum algorithms, there exists an SSL scheme in the common reference string model for searchable compute-and-compare circuits.*

1.4 Technical overview

Our goal in this section is to give a high level, but still technical, overview of the ideas and proofs presented in the thesis. Each chapter has a corresponding technical overview with more details than those presented here.

Technical background: the FLS Paradigm. One powerful framework for constructing zero-knowledge protocols is the FLS technique [FLS99]. Suppose we want to design a protocol for an NP language $\mathcal{L}$. There are two ideas behind the FLS technique. First, let $\mathcal{L}' \in \text{NP}$ be a language that we will fix later ($\mathcal{L}'$ will be chosen so that the rest of the FLS plan works out). Instead of designing a protocol for $\mathcal{L}$, we design a protocol for the composite language $\mathcal{L}''$ defined as follows: $(x, x') \in \mathcal{L}''$ if and only if either $x \in \mathcal{L}$ **or** $x' \in \mathcal{L}'$. For any $\mathcal{L}'$, completeness is unchanged by this transformation, i.e. if a prover can prove that $x \in \mathcal{L}$, it can also prove that $(x, x') \in \mathcal{L}''$. For soundness to still hold, a malicious prover shouldn't be able to prove that $x' \in \mathcal{L}'$; otherwise, a malicious prover can make the verifier accept despite $x \notin \mathcal{L}$. This means we need to fix $\mathcal{L}'$ and design a protocol in such a way that a malicious prover cannot use a witness w' to $x' \in \mathcal{L}'$. Nevertheless, it should be possible for a *simulator* to use a witness w' to $x' \in \mathcal{L}'$. For example, if the only way to obtain (x', w') is by rewinding the verifier, then a malicious prover wouldn't be able to break soundness, yet the simulator is able to convince the verifier. Or maybe, unlike the malicious prover, the simulator has non-black-box access to the verifier, which allows it to recover (x', w').

The first step discuss above is essentially creating a backdoor for the simulator to use. While the malicious prover cannot use the language $\mathcal{L}'$ to break soundness, by having more power via rewinding or non-black-box access, the simulator can use $\mathcal{L}'$. This is not enough for zero-knowledge. Consider a standard NP protocol for $\mathcal{L}''$, where P just sends a witness to V. Then, a witness to $(x, x'') \in \mathcal{L}''$ would be either a witness to $x \in \mathcal{L}$ or a witness to $x' \in \mathcal{L}'$. Unfortunately, V would be able to distinguish the

simulator from the prover, because only the simulator sends a witness to $x' \in \mathcal{L}'$. The second ingredient needed for the FLS trick is a *witness-indistinguishable* (WI) proof system. We say that a proof system for a language L is witness-indistinguishable if for any $x \in L$, and any pair of witness (w, w'), it is not possible to distinguish whether P used w or w' in the protocol. This notion is weaker than zero-knowledge. If we require P and V to perform a WI proof for $\mathcal{L}''$, then V wouldn't be able to distinguish when a witness for $x \in \mathcal{L}$ is used from when a witness $x' \in \mathcal{L}'$ is used. In other words, V cannot distinguish whether it interacted with the simulator or the real prover. To summarize the discussion until now: the FLS idea is to design a protocol between P and V where instead of just proving that $x \in \mathcal{L}$, the verifier will accept if either $x \in \mathcal{L}$ or $x' \in \mathcal{L}'$. The real prover will use a witness to $x \in \mathcal{L}$ while *only* the simulator can use a witness to $x' \in \mathcal{L}'$. Finally, WI proofs guarantee that V will not be able to distinguish whether it interacted with prover or simulator.

Let's look at the FLS trick through an example. Consider the argument (soundness against bounded provers) protocol in Figure 1-3. If $x \in \mathcal{L}$, then P will be able to make V accept in the WI protocol. If $x \notin \mathcal{L}$, then a malicious bounded prover will not be able to prove that $x \in \mathcal{L}$, and by security of the PRG, it will also not find the seed s satisfying $G(s) = y$. This means that soundness will still hold against bounded malicious provers. On the other hand, suppose that $x \in \mathcal{L}$, and suppose that a simulator Sim is capable of extracting s from $G(s)$ – perhaps Sim is computationally more powerful than the malicious P. Then, Sim will be able to use s as a witness in the WI protocol. By security of WI, the verifier wouldn't be able to distinguish whether a witness w to $x \in \mathcal{L}$ (the real prover) was used or whether the seed s (the simulator) was used in the WI.

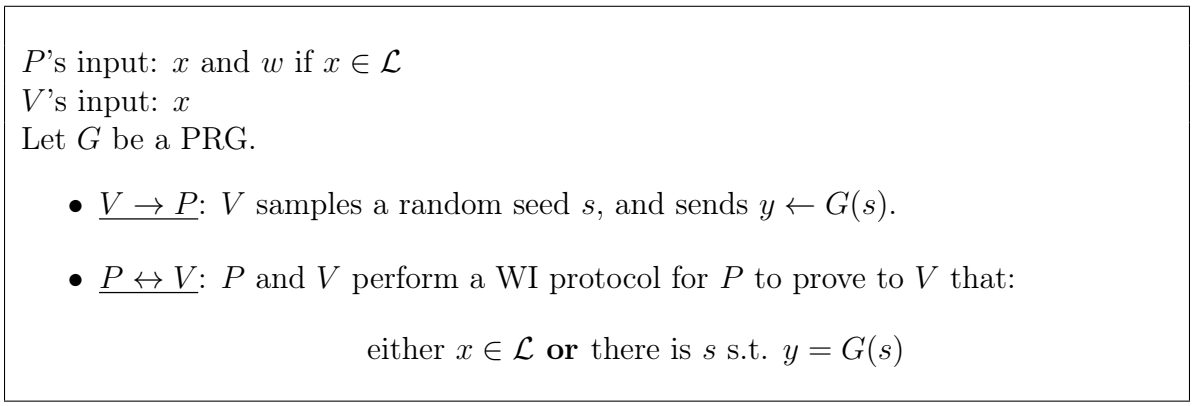

Figure 1-3: FLS Example

This example falls short of zero-knowledge because we haven't fully specified the simulator. In particular, to extract the seed s, the simulator might have to run in exponential time and not expected polynomial-time. Nevertheless, this example shows the main idea behind the FLS trick and behind our QZK constructions. The goal in constructing QZK protocols starting from the FLS paradigm is to instantiate the first step appropriately, i.e. to find the right primitives and protocols that allow a

simulator to extract the seed s while satisfying the desired zero-knowledge properties like expected polynomial run time.

1.4.1 QZK protocols

The constructions of our two main QZK protocols, (1) the classical argument system for NP, and (2) the bounded concurrent protocol, follow the FLS paradigm.

QZK classical argument system for NP

As discussed in the previous section, the FLS paradigm relies on the simulator's ability to do something that the malicious prover can't. One way to achieve this is to restrict to the setting where malicious provers are classical (PPT) while simulators are quantum (QPT). This exact situation arises when considering QZK protocols where soundness only has to hold against PPT machines. At a high level, our construction is an extension of the FLS trick to this setting, where our goal is to take advantage of the fact that the simulators are QPT. Instead of designing a protocol where P proves to V that $x \in \mathcal{L}$ or $x' \in \mathcal{L}'$, we design a protocol where the P proves that

$$\text{either } x \in \mathcal{L} \text{ or } P \text{ has quantum capabilities.}$$

For example, we could instantiate the protocol in Figure 1-3 with a PRG that is quantum *in-secure* but classically secure, then a QPT simulator would be able to extract the seed s. We can construct such protocols from quantum in-secure assumptions like Decisional Diffie-Hellman (DDH) or Factoring, but such protocols wouldn't satisfy the strongest security guarantee we would want in this setting. If we restrict any malicious party to be classical, at the very least we can hope to prove that the scheme remains secure if a malicious party has access to a quantum computer *after* executing the protocol. In other words, that no additional information will be obtained from a transcript of the execution if later everyone has quantum computers. This is a relevant scenario for the current state of affairs and the near future of quantum computation. If we use a quantum in-secure assumption, the malicious PPT prover will not break security of the protocol now, but in a quantum future, it will be able to get more information from the transcript. Going back to the example in Figure 1-3, if G is quantum in-secure, then the prover in the quantum future will be able to obtain the seed s. While this doesn't seem important in the standalone setting, this type of 'future' leakage could affect composition of protocols. Maybe the verifier had committed to using this same seed in a future execution of some other protocol, and by the time this other protocol gets executed, the malicious prover found access to a quantum computer and broke the transcript to get s.

The challenge is then to construct a classical argument system for NP solely from a quantum secure assumption like QLWE. Our main idea is to use QLWE-based noisy trapdoor claw-free functions (NTCFs), which have been used to construct 'tests of quantumness' [BCM+18]. NTCFs provide the computational asymmetry between classical and quantum computation. Assuming the security of NTCFs, there is a

computational task that a quantum simulator will be able to complete, while the malicious PPT prover will not be able to. In more details, an NTCF is a pair of functions (f_0, f_1) such that it is computationally hard to find x_0 and x_1 satisfying $f_0(x_0) = f_1(x_1)$ (a claw). The computational task that a quantum computer can do is the following. It can first compute a string y, and later it can do either of the following things: (1) provide a pre-image x_b s.t. $f_b(x_b) = y$ or (2) provide a string z s.t. $z \cdot (x_0 \oplus x_1) = 0$. A PPT machine cannot do this – it has to choose whether to compute (1) or (2) at the time it computes y. The ability to defer whether to compute (1) or (2) when computing y is a quantum capability. Using NTCFs, we can construct a protocol that lets P prove to V that:

(*) either $x \in \mathcal{L}$ or P can do the NTCF task.

Recall that the FLS trick is to first have a protocol for a composite (conjunction) NP statement, and then use a WI proof system. We can't use a WI proof yet because the statement (*) above is not an NP statement. To get an NP statement, we can replace the "P can do the NTCF task" part in (*) by an NP statement for which P can find a witness to if it can do the NTCF task. The transformation is as follows: V will commit to a trapdoor td, then V will reveal td to P if it passes the NTCF task. Finally, P and V will engage in a WI proof for P to prove that

(**) either $x \in \mathcal{L}$ or P knows td.

We are almost done. The issue with the prescription above is that the verifier will know whether the prover passes the NTCF challenge or not. But the real prover cannot pass the NTCF challenge, only the simulator can. This means that the verifier will know it is interacting with the simulator when it sees that the NTCF challenge was successfully completed. To prevent this from happening, we use *secure function evaluation* (SFE). Using an SFE scheme, P and V can evaluate a functionality that will output td to the P if and only if P passes the NTCF challenge. At the same time, SFE will guarantee the secrecy of P's input, so V will not be able to know whether the NTCF task was passed or not.

Quantum non-black-box extraction. The novelty in our approach above is to combine SFE with NTCFs to obtain a constant round extraction mechanism. This extraction mechanism allows the simulator to extract a trapdoor from the verifier, and this lets us use WI proof system to obtain the desired QZK protocol. We emphasize that this technique only works because the malicious prover is restricted to be PPT. Nevertheless, porting classical non-black-box techniques from [BKP19] to the quantum setting, we show how to obtain a constant round extraction mechanism in the full quantum setting (against malicious QPT parties). The high level idea of the non-black-box technique is shown Figure 1-4. The first step is to not worry about extracting td, but to extract an encryption of td instead. In the protocol shown in Figure 1-4, the only way for a QPT P to obtain an encryption of td is to receive td directly from V and then to encrypt it. But it will only receive td from V if it can find

a first. By the security of the encryption scheme, it cannot find a. In contrast, a simulator (extractor) that has non-black-box access to V will be able to homomorphically evaluate V on ct to obtain an encryption of the trapdoor, $\mathsf{QFHE.Enc}_{\mathsf{pk}}(\mathsf{td})$.

- $\underline{V \to P}$: V samples a trapdoor td, samples a random string a, and sends $\mathsf{ct} \leftarrow \mathsf{QFHE.Enc}_{\mathsf{pk}}(a)$.
- $\underline{P \to V}$: P sends b.
- $\underline{V \to P}$: If $a = b$, sends td. Othwerwise, send $\perp$.

Figure 1-4: High level idea behind non-black-box extraction

The final step in the extraction mechanism is to design a way to let the extractor decrypt the QFHE encryption of the trapdoor. This can be achieved by using *lockable obfuscation* [WZ17, GKW17]. We defer the rest of the technical details to Chapter 3.

Bounded concurrent QZK for NP

Our bounded concurrent QZK construction is similar to that of [PTW09], which is also based on the FLS technique. The construction in [PTW09] is as follows. Let x be the input to the protocol, and let $M = \text{poly}(|x|)$ and $T < M$ be parameters fixed ahead of time. P and V perform the protocol in Figure 1-5 below. Each execution of Steps (1) and (2) is called a *slot*. We say that the i^{th} slot *matched* if $a_i = b_i$. After performing the M slots, P and V engage in a WI protocol in order for P to prove that either $x \in \mathcal{L}$ **or** that at least T slots matched. If M and T are chosen appropriately, the probability that a prover matches at least T slots is negligible. On the other hand, a simulator that is allow to rewind after each slot will be able to match more than T slots.

For $i \in [M]$:

1. $\underline{P \to V}$: P commits to a random bit a_i.
2. $\underline{V \to P}$: V sends b_i.

Figure 1-5: Sequential execution of M slots

It was shown in [PTW09] that if the number of verifiers is known before the specification of the protocol, then M and T can be chosen appropriately so that this

scheme is a concurrent ZK protocol. The additional challenge in the concurrent setting is that different verifiers can send their messages at different times, and in particular, the ordering can be different after each rewinding of the simulator. By specifying the correct rewinding strategy, [PTW09] exhibited a concurrent ZK simulator. However, their simulator does not work to show that the protocol is also a QZK protocol, because the rewinding strategy depends on the scheduling (order of messages) which in turn depends on the verifiers' private states. This means that we cannot use Watrous' rewinding.

Our idea is to choose different M and T parameters such that it is possible to use Watrous' rewinding even in the concurrent setting. We introduce the concept of *block rewinding*, and show how block rewinding is the right rewinding strategy to construct a QZK simulator. Roughly speaking, block rewinding means that the simulator rewinds after a fixed number of messages (a block of messages) regardless of who sends those message. First, we divide all the messages sent between all the verifiers and the prover into L blocks, in a way that guarantees that each block will contain at least one slot. This means that L is chosen so that it is guaranteed that at least one verifier will receive the commitment to a_i and will respond with b_i within a single block of messages. Then, we rewind each block in order to match one slot within the rewound block. We show that choosing M, T and L appropriately that a block rewinding simulator will match at least T slots for all the verifiers. Furthermore, we can use Watrous' rewinding because the rewinding strategy is oblivious to the scheduling – the rewinding strategy is fixed ahead of time.

Quantum proofs of knowledge

Using rewinding to match commitments like in the protocol of Figure 1-5 is widely used to construct ZK or QZK simulator. We show that if we use an oblivious transfer protocol instead of commitments, then we can also extract. Recall that the aim in QPoK is to extract from the prover. For this purpose, suppose we only want to extract a single secret bit s. Figure 1-6 shows our extraction idea. The real verifier will only obtain s with probability $\frac{1}{2}$ [12]. On the other hand, a simulator that is allowed to rewind the prover will be able to rewind until it matches $c = b$, and obtain s with probability negligibly close to 1.

Because we want to use OT in order to extract from an unbounded prover, we require the OT to have security against malicious unbounded senders. This is called statistical receiver-private OT [GJJM20, DGH+20]. We need this security guarantee to make sure that the prover does not know which bit the verifier is retrieving, thus it will not be able to detect when it is interacting with the simulator that always retrive the secret s. This is what gives us the desired *simulability* property, because the prover cannot detect whether it is interacting with an extractor (that always guess correctly $c = b$) or with a verifier (that only guesses $c = b$ with probability $\frac{1}{2}$). Finally, to make sure that a malicious QPT verifier cannot obtain s with probability better than $\frac{1}{2}$, we need the OT scheme to be secure against malicious QPT receivers.

[12] We will later reduce this to be negligible.

Input of P: s

- P chooses random bits a and b. If $b = 0$, it sets $(m_0, m_1) = (s, a)$. If $b = 1$, it sets $(m_0, m_1) = (a, s)$.
- V chooses a random bit c.
- $\underline{P \leftrightarrow V}$: P and V perform an OT protocol where P takes the role of the sender and V takes the role of the receiver. The input of P in the OT protocol is (m_0, m_1) and the input of V is c.
- $\underline{P \rightarrow V}$: P sends b.

Figure 1-6: Simple OT extraction

In Chapter 4, we show how to construct an OT protocol with the desired security guarantees starting from the ideas in [GJJM20, DGH+20]. We also show how to use block rewinding in order to combine QPoK with our bounded concurrent QZK simulator to obtain bounded concurrent QZK with the additional QPoK property.

1.4.2 Impossibility of quantum copy-protection

We have seen in the previous section that extraction is the main underlying theme in our QZK results. In the context of QZK, we designed ways to extract information from QPT adversaries either in the black-box setting (via rewinding or NTCFs) or in the non-black-box setting (via QFHE). Our main observation is that we can extend these techniques to the setting of copy-protection. Specifically, we notice that if adversaries can extract useful information from any QPT implementation (U_C, ρ_C) of a Boolean circuit C, then they might be able to generate many copies of C. This would be the case if the circuit is de-quantumizable, i.e. if the information extracted is already a classical description of C. Our starting point for constructing de-quantumizable circuits is similar to the starting point for the non-black-box technique (Figure 1-4).

Construction of de-quantumizable circuits. Consider the circuit in Figure 1-7. Given any QPT implementation (U_C, ρ_C), we can evaluate $C(0 \dots 0)$ to obtain $\mathsf{ct} = \mathsf{QFHE.Enc}_{\mathsf{pk}}(a)$. If (U_C, ρ_C) is reusable (i.e. if we can use it to evaluate C many times), then we can also homomorphically evaluate the circuit on ct to obtain a ciphertext $\mathsf{QFHE.Enc}_{\mathsf{pk}}(b)$. If we are additionally given a black-box $\mathcal{O}_{\mathsf{sk},b}$ that on a ciphertext of b, outputs the secret key sk, then we can recover a complete classical description of C – i.e., we can recover a, b and ct. With the help of $\mathcal{O}_{\mathsf{sk},b}$, this would be a de-quantumizable circuit.

In the meantime, it can be shown that C is quantum unlearnable. To show this, we

$C_{a,b,\mathsf{ct}}(x)$:

- If $x = 0 \dots 0$, output $\mathsf{ct} := \mathsf{QFHE.Enc}_{\mathsf{pk}}(a)$.
- If $x = a$, output b.
- Otherwise, output $0 \dots 0$

Figure 1-7: An almost de-quantumizable circuit

first argue that with only oracle access to C, it is not possible to obtain a ciphertext of b. If C is not quantum unlearnable, and we can compute a QPT implementation (U_C, ρ_C) just from oracle access to C, then we would be able to obtain a ciphertext of b. This would be a contradiction. Our goal then is to show that from oracle access alone it is not possible to find a ciphertext of b. But with oracle access alone, it is not possible to homomorphically evaluate C, which means that the only way to compute anything about b is by querying the oracle on a. However, security of QFHE guarantees that a is hidden from us.

The final step is to instantiate the black-box circuit $\mathcal{O}_{\mathsf{sk},b}$ which we do with lockable obfuscation. We also let C output $\mathcal{O}_{\mathsf{sk},b}$ as well as $\mathsf{QFHE.Enc}_{\mathsf{pk}}(a)$ when evaluated at $x = 0 \dots 0$.

1.4.3 Construction of SSL

One reason for why copy-protection is hard to provably construct even for restricted families of circuits is because of *malleability*. To explain what we mean by malleability being challenging, consider the setting of quantum money. Given a state $|\$_s\rangle$ along with a serial number s, we can check whether $|\$_s\rangle$ is valid with respect to the serial number s. Suppose the correspondence between states and serial numbers is one-to-one, that is, for each serial number there is a unique state corresponding to it. An adversary that wants to attack this scheme would need to be able to produce two identical copies $|\$_s\rangle$ or to generate a completely new state $|\$_{s'}\rangle$ corresponding to a different serial number s'. The latter it cannot do because only the bank is allowed to generate new valid states, and the former it cannot do because of the No-Cloning Theorem. If there were more valid states associated to a serial number s, we would need to argue that the adversary cannot *maul* $|\$_s\rangle$ into two valid states $|\$'_s\rangle \otimes |\$''_s\rangle$ also with serial number s. This means that the No-Cloning Theorem wouldn't be enough to argue security. The same scenario arises in copy-protection where we not only need to argue that the pirate cannot clone the initial state, but also that it cannot maul it into any two states that evaluate the same circuit.

Our main insight is that we can use classical cryptographic primitives to tackle the challenge of malleability, at least in the weaker setting of SSL. The idea is as

follows. In the SSL setting, our goal is to guarantee that a pirate cannot produce two states that both correctly evaluate C under a fixed QPT algorithm Run. By fixing the Run algorithm, we can restrict the type of states that are useful to evaluate correctly, e.g. the Run algorithm can first check that the input state satisfies certain properties before evaluating the circuit C. This authentication performed by Run restricts the type of mauling that a pirate can do. Once we have restricted the states that a pirate has to prepare in order to break the security, we can use classical non-malleable primitives to argue that such mauling cannot happen. This would mean that the only way the pirate succeeds is by making two identical copies of a state of the initial state, which it cannot do due to the No-Cloning Theorem.

We design the Run algorithm so that it expects states of the form $(|\psi_s\rangle, d_s, s, C)$, where $|\psi_s\rangle$ is some state associated with a string s, and d_s is a classical string also associated with s, and C is the Boolean circuit being SSL protected. There are two properties needed: (1) d_s is non-malleable, i.e. it is not possible for an adversary to output $d_{s'}$ for some other $s' \neq s$[13], and (2) we can check that both $|\psi_s\rangle$ and d_s correspond to s (similar to quantum money). Condition (1) will protect us against maulers, and condition (2) will protect us against cloners[14]. Suppose that given a state ρ_s, a pirate outputs two states σ_1 and σ_2 both of the form that Run expects. Then there are two cases, either both states have the same string s as ρ_s (pirate is a cloner), or at least one of the copies, say σ_2, have a different $s' \neq s$ (pirate is a mauler). By making sure that the d_s part of the input is non-malleable, the latter case cannot happen, because the pirate cannot prepare a different $d_{s'}$. We deal with the former case by choosing our states $(|\psi_s\rangle, s)$ the same way Zhandry does in [Zha19] to construct public-key quantum money from indistinguishability obfuscation. By [Zha19], an adversary will not be able to clone these states, so the pirate cannot be a cloner either.

1.5 Related work

The following is an overview of the literature that is closely related to this thesis. A general survey of quantum cryptography can be found in [BS16]. See [VW16] for a general treatment on quantum proofs and quantum zero-knowledge.

1.5.1 (Computational) Quantum zero-knowledge

Quantum zero-knowledge was first studied by Watrous [Wat09]. He proved how to achieve QZK for all of NP using quantum concealing commitments. His protocol does not have negligible soundness, so we have to sequentially repeat it in order to improve the soundness. The resulting protocol would be a polynomial round QZK protocol with negligible soundness. The main idea behind his proof is the use of Watrous' rewinding. Soon after, some classical ZK results were generalized to the

[13] You can think of d_s here as a non-malleable signature of s.

[14] Maulers are adversaries that intend to make a new copy with a different s' from the original s. Cloners or duplicators are adversaries that intend to make copies with the same s as the original.

quantum setting. Kobayashi [Kob08] showed that some of the known classical ZK results also hold in the quantum setting: (a) any honest verifier QZK protocol can be made into a malicious verifier QZK protocol, (b) any QZK protocol can be made into a public-coin QZK protocol, (c) any QZK protocol can be made to have perfect completeness, and (d) any QZK protocol can be made into a 3-round public-coin QZK with perfect completeness (but not negligible soundness). Jain, Kolla, Midrijanis, and Reichardt [JKMR06] showed how to generalize the classical impossibility result of Goldreich and Krawczyk [GK96b] to the quantum setting. They showed that there are no 3-round or constant round public-coin black-box QZK proof systems for NP unless BQP $\subseteq$ NP.

A different type of quantum rewinding was introduced by Unruh in [Unr12], where he introduced quantum proofs of knowledge and showed that under certain conditions ('special' and 'strict' soundness) his rewinding can be applied to get QPoK for NP from existing Σ-protocols. Unfortunately, his QPoK protocol does not satisfy extractability – there is no extractor that is also a simulator for a malicious prover, so it cannot be generally used inside other protocols. Simulability and extractabily was achieved against *bounded* quantum adversaries in follow-up work [HSS11, LN11] using stronger assumptions (mixed commitments) than those used by Unruh. QFHE-based non-black-box extraction techniques [BS20, ALP20] could be used to achieve extractability, but they would also work only against bounded quantum adversaries. Unruh's techniques were extended to get arguments of knowledge based on a collapse binding commitments in [Unr16].

The inherent difficulties of applying Watrous' or Unruh's rewinding to show that existing classical protocols are quantum secure were studied in [ARU14]. Relative to an oracle, they showed that many classical protocols are actually quantum insecure, and that in some sense, the conditions needed to apply Watrous' or Unruh's rewinding seem necessary to argue quantum security of existing ZK or PoK protocols.

The first constant round QZK argument system for NP (and for QMA) was constructed by Bitansky and Shmueli [BS20] assuming both QLWE and QFHE. They concurrently developed a similar non-black-box technique (using QFHE and lockable obfuscation) to the one we present in Chapter 3. While we only show how to get an extraction protocol (where in particular, the malicious sender is semi-malicious) using the non-black-box techniques, they show how to deal with malicious, possibly aborting, verifiers in order to get full QZK. They make crucial use of Watrous' rewinding to do this.

Constant round ZK protocols have been known in the classical setting since the work of Goldreich and Kahan [GK96a]. Unlike the QZK protocol of [BS20], Goldreich and Kahan's protocol is a *black-box* ZK protocol constructed from collision-resistant hash functions. A quantum version of Goldreick and Kahan was developed by Chia, Chung, and Yamakawa [CCY20] using collapse binding hash functions, albeit for a weaker notion called quantum ϵ-ZK. It turns out that the existence of constant round black-box ZK protocols appears to only hold in the classical setting. Recently, Chia, Chung, Liu, and Yamakawa [CCLY21] showed that there are no constant round

black-box QZK proofs or quantum arguments[15] for NP unless NP $\subseteq$ BQP.

QZK protocols with different security properties beyond the basic QZK definition have been recently developed. Extending the non-black-box techniques of [BS20, ALP20], Agarwal, Bartusek, Goyal, Khurana, and Malavolta [ABG+20] construct constant round QZK protocols that are secured under parallel composition in the bounded setting, i.e. where the prover is interacting in parallel with a number of quantum verifiers that is fixed before the protocol's specification. Another notion that have been studied in the classical setting, resettably-sound ZK protocols [BGGL01], have been extended to the QZK setting by Bitansky, Kellner, and Shmueli [BKS21]. Starting from the non-black-box QZK protocol [BS20], they construct resettably-sound QZK quantum arguments for NP. They also show an alternate proof of the impossibility of result of [JKMR06]. Curiously enough, they also show that the existence of a resettably-sound QZK protocol for NP implies the impossibility of quantum VBB, providing an alternate proof of our impossibility result presented in Chapter 6 and independently obtained by [ABDS20].

QZK for QMA. The first QZK protocol for QMA was constructed by Broadbent, Ji, Song, and Watrous [BJSW16]. Their main idea is to use a quantum authentication scheme that allows them to reduce the construction of a QZK protocol for QMA to using a QZK protocol for NP. A different QZK protocol for QMA that has the commit-and-open structure found in the classical ZK protocols and a non-interactive QZK protocol in the secret parameter setting were recently constructed in [BG20]. Concurrently, Coladangelo, Vidick, and Zhang [CVZ20] also constructed a non-interactive QZK protocol for QMA but in the pre-processing model. Both [BG20, CVZ20] defined and constructed proofs of quantum knowledge for QMA (PoQK). These are analogous to QPoKs except that the witness is a QMA witness so the extractor has the task of extracting a quantum state. Their PoQK protocols have similar limitations to Unruh's QPoK. Combining the techniques develop in the context of verification of quantum computation [Mah18b] with the BJSW framework [BJSW16], Vidick and Zhang showed how to construct a classical ZK argument system for QMA where the verification is fully classical [VZ20]. Their argument system was made non-interactive via a Fiat-Shamir type of transformation in the quantum random oracle model by Alagic, Childs, Grilo, and Hung [ACGH20]. While other flavors of non-interactive QZK for QMA have been studied: (1) multi-theorem designated verifiers [Shm20], and (2) dual-mode classically verifiable with preprocessing [MY21], we still do not know how to achieve NIQZK solely from QLWE.

1.5.2 Unclonable Primitives, Copy-Protection, and SSL

The study of unclonable primitives has picked up paced over the last decade. A few constructions [Aar09, LAF+09, Gav12, FGH+12, AC12] achieved quantum money with various features and very recently, in a breakthrough work, Zhandry [Zha19] shows how to construct publicly-verifiable quantum money from cryptographic as-

[15] QPT provers

sumptions. Zhandry also introduced a stronger notion of quantum money, which he coined quantum lightning, and constructed it from cryptographic assumptions.

Unclonability has also been studied in the context of encryption schemes. The work of Gottesman [Got03] studies the problem of quantum tamper detection. Alice can use a quantum state to send Bob an encryption of a classical message m with the guarantee that any eavsdropper could not have cloned the ciphertext. After Bob receives the ciphertext, he can check if the state has been tampered with, and if this is not the case, he would know that a potential eavsdropper did not keep a copy of the ciphertext. In recent work, Broadbent and Lord [BL19] introduced the notion of unclonable encryption. Roughly speaking, an unclonable encryption allows Alice to give Bob and Charlie an encryption of a classical message m, in the form of a quantum state $\sigma(m)$, such that Bob and Charlie cannot 'split' the state among them. Ananth and Kaleoglu [AK21] extended the construction of unclonable encryption to the setting where the encryption key can be re-used multiple times. On the other hand, Majenz, Schaffner, and Tahmasbi [MST21] studied the limitations of unclonable encryption in the information-theoretic setting. The related notion of unclonable decryption keys was studied in [GZ20].

In a follow-up work, Broadbent and Islam [BI19], construct a one-time use encryption scheme with certifiable deletion. An encryption scheme has certifiable deletion, if there is an algorithm to check that a ciphertext was deleted. The security guarantee is that if an adversary is in possession of the ciphertext, and it then passes the certification of deletion, the issuer of the encryption can now give the secret key to the adversary. At this point, the adversary still can't distinguish which plaintext correspond to the ciphertext it was given.

One-shot or one-time primitives provide one of the strongest version of unclonability. Arguably, the most well-known one-time primitive is one-time programs [GKR08]. Quantum one-time programs, that use only quantum information, are not possible even under computational assumptions [BGS13]. This rules out the possibility of having a copy-protection scheme where a single copy of the software is consumed by the evaluation procedure. Despite the lack of quantum one-time programs, there are constructions of secure signature tokens and one-shot signatures in the oracle models [BDS16, AGKZ20]. A quantum token for signatures is a quantum state that would let anyone in possession of it to sign an arbitrary document, but only once. The token is destroyed in the signing process.

Quantum copy-protection was introduced by Aaronson [Aar09]. He constructed a copy-protection scheme for arbitrary unlearnable functions in the quantum oracle model, and he provided two heuristic constructions (i.e. without security proofs) for copy-protecting point functions. An earlier version of [ALL+20] improved the result to get copy-protection in the classical oracle model. They also proved that the existence of copy-protection implies public-key quantum money assuming QLWE.

Follow-Up Work. There has been many follow-ups since our introduction of SSL. While our construction of SSL makes it seem that you might need public-key quantum money to get SSL, Kitagawa, Nishimaki, and Yamakawa [KNY21] showed that

this is not the case. They introduce a relaxed version of quantum lightning, which they called two-tier quantum lightning, and show how to use this to achieve finite-term SSL for PRFs (under QLWE) and for a subclass of evasive functions (under sub-exponential QLWE). Coladangelo, Majenz, and Poremba [CMP20] showed how to obtain SSL for our same class of evasive circuits but in the quantum random oracle model. Recently, Broadbent, Jeffery, Lord, Podder, and Sundaram [BJL+21] showed how to achieve SSL for compute-and-compare circuits without any assumptions. Specifically, they define a stronger notion than SSL, honest-malicious copy-protection, where a pirate wins if it generates two copies that evaluated correctly and at least one copy is an authenticated copy (whereas in SSL both copies have to be authenticated). A similar notion to infinite-term SSL, called quantum copy-detection, was introduced in [ALL+20]. They show that this notion is achievable for circuits that can be watermarked [KW17, CHN+16].

Concurrent Work on QVBB. Our construction of de-quantumizable circuits also rules out the existence of quantum VBB for classical circuits assuming QFHE and QLWE; this was stated as an open problem by Alagic and Fefferman [AF16]. Concurrently, [ABDS20] also rule out quantum virtual black-box obfuscation under the assumption of QLWE; unlike our work they don't additionally assume the existence of QFHE.

In hindsight, it shouldn't be surprising that non-black box techniques developed in the context of quantum zero-knowledge [BS20, ALP20] are relevant to proving the impossibility of quantum obfuscation; the breakthrough work of Bitansky and Paneth [BP13] show how to construct (classical) zero-knowledge protocols with non-black box simulation using techniques developed in the context of (classical) obfuscation.

1.6 Organization and bibliographical information

This thesis is based on work done in collabration with Prabhanjan Ananth and already presented in the following papers: 'Secure Quantum Extraction Protocols' [ALP20][16] (in TCC '20), and 'Secure Software Leasing' [ALP21][17] (in EUROCRYPT '21 and QIP '21), and 'On the Concurrent Composition of Quantum Zero-Knowledge' [ACLP21][18] (in submission) also in collaboration with Kai-Min Chung.

- Chapter 2 contains all the preliminary definitions and conventions used throughout the thesis.
- Chapter 3 reproduces the definition and construction of quantum extraction protocols (QEXT) from [ALP20]. There are two types of QEXT protocols depending on whether the malicious receiver is classical (cQEXT) or quantum

[16] arXiv:1911.07672
[17] arXiv:2005.05289
[18] arXiv:2012.03139

(qQEXT). The construction of cQEXT is used to get the QZK classical argument system for NP in Chapter 4.

- Chapter 4 reproduces the work done in [ALP20, ACLP21]. This chapter contains the construction of $O(1)$-round QZK classical arguments for NP, and the bounded concurrent QZK proof systems for NP and QMA.
- Chapter 5 reproduces work in [ACLP21]. It contains a construction of post-quantum statistical receiver-private OT, which is then used to construct quantum proofs of knowledge. We also show how to extend QPoKs to the bounded concurrent setting.
- Chapters 6 and 7 reproduce the work done in [ALP21]. Chapter 6 contains the construction of de-quantumizable circuits and the impossibility results. Chapter 7 is fully dedicated to our new SSL notion.

Chapter 2

Preliminaries

2.1 Notation and conventions

We assume that the reader is familiar with basic cryptographic notions such as negligible functions and computational indistinguishability (see [G+05]). As much as possible, we will be consistent with the papers where the work in this thesis was presented, so sometimes a quantum-secure primitive X will be explicitly denoted by q-X. However, we assume that all the primitives are quantum secure unless otherwise stated.

The security parameter is always denoted by λ and we denote $\mathrm{negl}(\lambda)$ to be a negligible function in 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.

Whenever we talk about polynomial-time (PPT) or quantum polynomial-time (QPT) adversaries, we assume they take auxiliary inputs (i.e. they are non-uniform).

2.2 Quantum background

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, we call these density matrices, and use the notation ρ or σ to stand for density matrices when possible. 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. We use the trace distance, denoted by $\|\rho - \sigma\|_{\mathrm{tr}}$, as our distance measure on quantum states,

$$\|\rho - \sigma\|_{\mathrm{tr}} = \frac{1}{2}\mathrm{Tr}\left[\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\right]$$

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 $\{C_n\}_{n\in\mathbb{N}}$. As stated before, we always assume that the QPT adversaries are non-uniform – a QPT adversary $\mathcal{A}$ acting on n qubits could be given a quantum auxiliary state with $\text{poly}(n)$ qubits.

Quantum Computational Indistinguishability. When we talk about quantum distinguishers, we need the following definitions, which we take from [Wat09, BJSW16].

Definition 22 (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*

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

for some 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 23 (Indistinguishability of channels)**.** *Let I be an infinite subset $I \subset \{0,1\}^*$, let $p, q : \mathbb{N} \to \mathbb{N}$ 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 $t : \mathbb{N} \to \mathbb{N}$, any $p(|x|) + t(|x|)$-qubit quantum state ρ, and for all $x \in I$, we have that*

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

for some function $\epsilon : \mathbb{N} \to [0,1]$. 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.

Quantum Fourier Transform and Subspaces. Our main construction uses the same type of quantum states (superpositions over linear subspaces) considered by [AC12, Zha19] in the context of constructing quantum money.

We recall some key facts from these works relevant to our construction. Consider the field $\mathbb{Z}_q^\lambda$ where $q \geq 2$,and let FT denote the quantum fourier transfrom over $\mathbb{Z}_q^\lambda$.

For any linear subspace A, let $A^\perp$ denote its orthogonal (dual) subspace,

$$A^\perp = \{v \in \mathbb{Z}_q^\lambda | \langle v, a\rangle = 0\}.$$

Let $|A\rangle = \frac{1}{\sqrt{|A|}} \sum_{a\in A} |a\rangle$. The quantum fourier Transform, FT, does the following:

$$\mathsf{FT}|A\rangle = |A^\perp\rangle.$$

Since $(A^\perp)^\perp = A$, we also have $\mathsf{FT}|A^\perp\rangle = |A\rangle$.

Let $\Pi_A = \sum_{a\in A} |a\rangle\langle a|$, then as shown in Lemma 21 of [AC12],

$$\mathsf{FT}(\Pi_{A^\perp})\mathsf{FT}\Pi_A = |A\rangle\langle A|.$$

Almost As Good As New Lemma. We use the Almost As Good As New Lemma [Aar04], restated here verbatim from [AC12].

Lemma 24 (Almost As Good As New)**.** *Let ρ be a mixed state acting on $\mathbb{C}^d$. Let U be a unitary and $(\Pi_0, \Pi_1 = 1 - \Pi_0)$ be projectors all acting on $\mathbb{C}^d \otimes \mathbb{C}^d$. We interpret (U, Π_0, Π_1) as a measurement performed by appending an acillary system of dimension d' in the state $|0\rangle\langle 0|$, applying U and then performing the projective measurement $\{\Pi_0, \Pi_1\}$ on the larger system. Assuming that the outcome corresponding to Π_0 has probability $1 - \varepsilon$, i.e., $Tr[\Pi_0(U\rho \otimes |0\rangle\langle 0|U^\dagger)] = 1 - \varepsilon$, we have*

$$\|\rho - \widetilde{\rho}\|_{tr} \leq \sqrt{\varepsilon},$$

where $\widetilde{\rho}$ is state after performing the measurement and then undoing the unitary U and tracing out the ancillary system:

$$\widetilde{\rho} = Tr_{d'}\left(U^\dagger\left(\Pi_0 U\left(\rho \otimes |0\rangle\langle 0|\right)U^\dagger\Pi_0 + \Pi_1 U\left(\rho \otimes |0\rangle\langle 0|\right)U^\dagger\Pi_1\right)U\right)$$

We use this Lemma to argue that whenever a QPT algorithm $\mathcal{A}$ on input ρ, outputs a particular bit string z with probability $1-\varepsilon$, then $\mathcal{A}$ can be performed in a way that also lets us recover the initial state. In particular, given the QPT description for $\mathcal{A}$, we can implement $\mathcal{A}$ with an acillary system, a unitary, and only measuring in the computational basis after the unitary has been applied, similarly to Lemma 24. Then, it is possible to uncompute in order to also obtain $\widetilde{\rho}$.

Interactive Models. We model an interactive protocol between a prover, P, and a verifier, V, as follows. There are 2 registers R_P and R_V corresponding to the prover's and the verifier's private registers, as well as a message register, R_{M}, which is used by both P and V 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 have 3 different notions of interactive computation.

1. **Classical protocol:** An interactive protocol is classical if R_P, R_V, and R_{M} are classical, and P and V can only perform classical computation.
2. **Quantum protocol with classical messages:** An interactive protocol is quantum with classical messages if either one of R_P or R_V is a quantum register, and R_{M} is classical. P and V can perform quantum computations if the respective private register is quantum, but they can only send classical messages.
3. **Quantum protocol:** An interactive protocol is fully quantum if all the registers are quantum. P and V can perform quantum operations.

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 and Quantum View. For any protocol (P, V), the interaction between P and V on input x induces a quantum channel $\mathcal{E}_x$ acting on their private input states, ρ_P and σ_V. We denote the view of V when interacting with P by

$$\mathsf{View}_V \left\langle P\left(x, \rho_P\right), V\left(x, \sigma_V\right)\right\rangle,$$

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

$$\mathsf{View}_V \left\langle P\left(x, \rho_P\right), V\left(x, \sigma_V\right)\right\rangle := \mathrm{Tr}_{\mathrm{R}_P}\left[\mathcal{E}_x\left(\rho_P \otimes \sigma_V\right)\right].$$

From the verifier's point of view, the interaction induces the channel

$$\mathcal{E}_{x,V}(\sigma) = \mathrm{Tr}_{\mathrm{R}_P}\left[\mathcal{E}_x(\sigma \otimes \rho_P)\right]$$

on its private input state.

2.2.1 Quantum Zero-Knowledge (QZK)

Quantum zero-knowledge was initially defined by Watrous as follows.

Definition 25 (Quantum Zero-Knowledge [Wat09])**.** *An interactive proof system (P, V) for a promise problem $\mathcal{A} = \mathcal{A}_{yes} \cup \mathcal{A}_{no}$ is* **quantum computational zero-knowledge (QZK)** *if for any QPT verifier V^* with a private register of size $\text{poly}(|x|)$, there exists a QPT simulator Sim such that the following holds.*

$$\{\mathsf{View}_{V^*}\langle P, V^*(x,\cdot)\rangle\}_{x\in\mathcal{A}_{yes}} \approx_Q \{\mathsf{Sim}(V^*, x, \cdot)\}_{x\in\mathcal{A}_{yes}}$$

In other words, that the collection of channels induced on the private state of V^* by the interaction with P on input x is computationally indistinguishable from the collection of channels induced by the simulator Sim on input x.

When convenient in terms of exposition or writing, we will also use the following equivalent definition in terms of indistinguishability of quantum states.

Definition 26 (Alternative definition of QZK)**.** *An interactive proof system (P, V) for a promise problem $\mathcal{A} = \mathcal{A}_{yes} \cup \mathcal{A}_{no}$ is* **quantum computational zero-knowledge (QZK)** *if for any QPT verifier V^*, there exists a QPT simulator Sim such that the following holds. For all $x \in \mathcal{A}_{yes}$, for any $\text{poly}(|x|)$-qubits bipartite state, ρ_{AB}, on registers A and B,*

$$\mathsf{View}_{V^*}\langle P, V^*(x, \rho_{AB})\rangle \approx_Q \mathsf{Sim}(V^*, x, \rho_{AB})$$

where V^* and Sim only have access to register A. In other words, only the identity is performed on register B.

Remark 27. *We could restate these definitions using a security parameter λ instead. In that case, we would also have 1^λ as an input to V^*. The size of the advice state ρ would be $\text{poly}(\lambda)$ instead and the indistinguishability parameter would be a negligible function in the security parameter, $\epsilon(\lambda)$. The definitions would be the same whenever $\lambda = \text{poly}(|x|)$ as is usually assumed.*

2.2.2 Watrous Rewinding Lemma

We will make heavy use of the following lemma due to Watrous [Wat09].

Lemma 28 (Watrous Rewinding Lemma)**.** *Suppose Q be a quantum circuit acting on $n + k$ qubits such that for every n-qubit state $|\psi\rangle$, the following holds:*

$$Q|\psi\rangle|0^{\otimes k}\rangle = \sqrt{p(\psi)}\ |0\rangle|\phi_0(\psi)\rangle + \sqrt{1-p(\psi)}\ |1\rangle|\phi_1(\psi)\rangle$$

Let $p_0, p_1 \in (0,1)$ and $\varepsilon \in (0, 1/2)$ be real numbers such that:

- $|p(\psi) - p_1| \leq \varepsilon$
- $p_0(1-p_0) \leq p_1(1-p_1)$, *and*

- $p_0 \leq p(\psi)$

for all n-qubit states. Then there exists a general quantum circuit R of size $O\left(\frac{\log(1/\varepsilon)\mathsf{size}(Q)}{p_0(1-p_0)}\right)$ satisfying the following property:

$$\langle\phi_0(\psi)|\rho(\psi)|\phi_0(\psi)\rangle \geq 1 - 16\varepsilon\frac{\log^2(1/\varepsilon)}{p_0^2(1-p_0)^2}$$

In this case, we define R to be $\mathsf{Amplifier}(Q,\varepsilon)$. If ε is a negligible function in the security parameter, we omit this from the algorithm.

2.3 Learning with errors

We consider the decisional learning with errors (LWE) problem, introduced by Regev [Reg09]. We define this problem formally below.

> The problem (n, m, q, χ)-LWE, where $n, m, q \in \mathbb{N}$ and χ is a distribution supported over $\mathbb{Z}$, is to distinguish between the distributions $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{n\times 1}, \mathbf{e} \xleftarrow{\$} \chi^{m\times 1}$ and $\mathbf{u} \leftarrow \mathbb{Z}_q^{m\times 1}$.

The above problem has been believed to be hard against classical PPT algorithms – also referred to as LWE assumption – has had many powerful applications in cryptography. In this work, we conjecture the above problem to be hard even against QPT algorithms; this conjecture referred to as QLWE assumption has been useful in the constructions of interesting primitives such as quantum fully-homomorphic encryption [Mah18a, Bra18]. We refer to this assumption as QLWE assumption.

> **QLWE assumption**: This assumption is parameterized by λ. Let $n = \mathrm{poly}(\lambda)$, $m = \mathrm{poly}(n \cdot \log(q))$ and χ be a discrete Gaussian distribution[1] with parameter $\alpha q > 0$, where α can set to be any non-negative number.
>
> Any QPT distinguisher (even given access to polynomial-sized advice state) can solve (n, m, q, χ)-LWE only with probability $\mathrm{negl}(\lambda)$, for some negligible function negl.

Remark 29. *We drop the notation λ from the description of the assumption when it is clear.*

(n, m, q, χ)-LWE is shown [Reg09, PRSD17] to be as hard as approximating shortest independent vector problem (SIVP) to within a factor of $\gamma = \tilde{O}(n/\alpha)$ (where α is defined above). The best known quantum algorithms for this problem run in time $2^{\tilde{O}(n/\log(\gamma))}$.

For our construction of SSL, we require a stronger version of QLWE that is secure even against sub-exponential quantum adversaries. We state this assumption formally below.

[1] Refer [Bra18] for a definition of discrete Gaussian distribution.

T-Sub-exponential QLWE Assumption: This assumption is parameterized by λ and time T. Let $n = T + \text{poly}(\lambda)$, $m = \text{poly}(n \cdot \log(q))$ and χ be a discrete Gaussian distribution with parameter $\alpha q > 0$, where α can set to be any non-negative number.

Any quantum distinguisher (even given access to polynomial-sized advice state) running in time $2^{\widetilde{O}(T)}$ can solve (n, m, q, χ)-LWE only with probability $\text{negl}(\lambda)$, for some negligible function negl.

2.4 Cryptographic primitives

2.4.1 Noisy Trapdoor Claw-Free Functions (NTCF)

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].

Definition 30 (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 $\{f_{\mathbf{k},b} : \mathcal{X} \to D_{\mathcal{Y}}\}_{\mathbf{k}\in\mathcal{K}, b\in\{0,1\}}$ 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 $f'_{\mathbf{k},b} : \mathcal{X} \to D_{\mathcal{Y}}$ 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 $\mathsf{Chk} : \mathcal{K} \times \{0,1\} \times \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ 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* $w : \mathbb{N} \to \mathbb{N}$*, 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.* $\Pr_{d \leftarrow \{0,1\}^{w(\lambda)}} [d \notin G_{\mathbf{k},b,x}] \leq negl(\lambda)$*. 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* $J : \mathcal{X} \to \{0,1\}^{w(\lambda)}$*, that can be inverted efficiently in its range, and for which the following holds. Let*

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

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

  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.

2.4.2 Commitments

Perfectly Binding Commitments

A commitment scheme consists a classical PPT algorithm Comm that takes as input security parameter 1^λ, input message x and outputs the commitment $\mathbf{c}$. Typically, there is an opening algorithm associated with Comm, but we do not make use of this algorithm in our work.

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 31 (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 32 (Quantum-Computational Hiding/Concealing)**.** *A commitment scheme* Comm *is said to be* **quantum computationally hiding or concealing** *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.

Statistically Binding

We employ a two-message commitment scheme that satisfies the following two properties.

Definition 33 (Statistically Binding)**.** *A two-message commitment scheme between a committer,* Comm*, and a receiver* R*, both running in probabilistic polynomial time, is said to satisfy statistical binding property if the following holds for any adversary* $\mathcal{A}$*:*

$$\mathsf{Pr}\left[\begin{array}{c}(\mathbf{c},r_1,x_1,r_2,x_2)\leftarrow\mathcal{A}\\ \bigwedge\\ \mathsf{Comm}(1^\lambda,\mathbf{r},x_1;r_1)=\mathsf{Comm}(1^\lambda,\mathbf{r},x_2;r_2)=\mathbf{c}:\mathbf{r}\leftarrow\mathsf{R}(1^\lambda)\\ \bigwedge\\ x_1\neq x_2\end{array}\right]\leq negl(\lambda),$$

for some negligible function negl.

Definition 34 (Quantum-Computationall Hiding/Concealing)**.** *A two-message commitments schemes,*$(\mathsf{Comm},\mathsf{R})$*, is said to be* **quantum computational hiding or concealing** *if the following holds. Suppose* $\mathcal{A}$ *be a non-uniform QPT algorithm and let* $\mathbf{r}$ *be the message generated by* $\mathcal{A}(1^\lambda)$*. We require that* $\mathcal{A}$ *cannot distinguish the two distributions,* $\{\mathsf{Comm}(1^\lambda,\mathbf{r},x)\}$ *and* $\{\mathsf{Comm}(1^\lambda,\mathbf{r},x')\}$*, for any two inputs* x,x'*.*

Remark 35. *We only considered two message protocols in the above definition for simplicity in the constructions.*

Instantiation. We can instantiate statistically binding and quantum-concealing commitments from post-quantum one-way functions [Nao91].

2.4.3 Quantum Fully Homomorphic Encryption (QFHE)

A fully homomorphic encryption scheme allows for publicly evaluating an encryption of x using a function f to obtain an encryption of $f(x)$. Traditionally f has been modeled as classical circuits but in this work, we consider the setting when f is modeled as quantum circuits and when the messages are quantum states. This notion is referred to as quantum fully homomorphic encryption (QFHE). We state our definition verbatim from [BJ15].

Definition 36. *Let* $\mathcal{M}$ *be the Hilbert space associated with the message space (plaintexts),* $\mathcal{C}$ *be the Hilbert space associated with the ciphertexts, and* $\mathcal{R}_{evk}$ *be the Hilbert space associated with the evaluation key. A* **quantum fully homomorphic encryption scheme** *is a tuple of QPT algorithms* $\mathsf{QFHE}=(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec},\mathsf{Eval})$ *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.*

- $\mathsf{QFHE.Enc}(\mathsf{pk}, \cdot) : L(\mathcal{M}) \to L(\mathcal{C})$*: takes as input a state* ρ *and outputs a ciphertext* σ

- $\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot) : L(\mathcal{C}) \to L(\mathcal{M})$*: takes a quantum ciphertext* σ*, and outputs a state* ρ *in the message space* $L(\mathcal{M})$*.*

- $\mathsf{QFHE.Eval}(\mathcal{E}, \cdot) : L(\mathcal{R}_{evk} \otimes \mathcal{C}^{\otimes n}) \to L(\mathcal{C}^{\otimes m})$*: takes as input a quantum circuit* $\mathcal{E} : L(\mathcal{M}^{\otimes n}) \to L(\mathcal{M}^{\otimes m})$*, and a ciphertext in* $L(\mathcal{C}^{\otimes n})$ *and outputs a ciphertext in* $L(\mathcal{C}^{\otimes m})$*, 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 full definition. In our work, we only use QFHE to encrypt classical messages, so the security notion that is enough for us is the following:

Semantic security of classical plaintexts. We say that QFHE satisfies semantic security of classical messages if for any QPT adversary $\mathcal{A}$, and any messages m_0 and m_1, the following holds when $(\mathsf{pk}, \mathsf{sk}, \rho_{evk}) \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$,

$$|\Pr[\mathcal{A}(\mathsf{pk}, \rho_{evk}, \mathsf{QFHE.Enc}_{\mathsf{pk}}(m_0)) = 1] - \Pr[\mathcal{A}(\mathsf{pk}, \rho_{evk}, \mathsf{QFHE.Enc}_{\mathsf{pk}}(m_1))]| \leq \mathrm{negl}(\lambda).$$

We use QFHE in our non-black-box extraction construction as well as in our construction of de-quantumizable circuits. In both of these cases, we require additional properties from the QFHE schemes. In the non-black-box construction we require both of the properties described below to hold, while in the construction of de-quantumizable circuits, we only require the classical ciphertexts condition.

1. (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 $r_1, r_2 \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

- $\Pr[x \leftarrow \mathsf{QFHE.Dec}_{\mathsf{sk}}(\mathsf{QFHE.Enc}_{\mathsf{pk}}(x)) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QFHE.Gen}(1^\lambda)] = 1$

- $\Pr[\mathsf{QFHE.Dec}_{\mathsf{sk}}(\mathsf{QFHE.Eval}(\mathsf{pk}, \mathcal{E}, \mathsf{ct})) = \mathcal{E}(x)] \geq 1 - \mathrm{negl}(\lambda)$, for some negligible function negl, where: (1) $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QFHE.Setup}(1^\lambda; r_1)$ and, (2) $\mathsf{ct} \leftarrow \mathsf{QFHE.Enc}_{\mathsf{pk}}(x; r_2)$. The probability is defined over the randomness of the evaluation procedure.

2. Classical ciphertexts. We require a QFHE scheme where ciphertexts of classical plaintexts are also classical. Given any $x \in \{0,1\}$, we want $\mathsf{QFHE.Enc}_{\mathsf{pk}}(|x\rangle\langle x|)$ to be a computational basis state $|z\rangle\langle z|$ for some $z \in \{0,1\}^l$ (here, l is the length of ciphertexts for 1-bit messages). In this case, we write $\mathsf{QFHE.Enc}_{\mathsf{pk}}(x)$. We also want the same to be true for evaluated ciphertexts, i.e. if $\mathcal{E}(|x\rangle\langle x|) = |y\rangle\langle y|$ for some $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$, then

$$\mathsf{QFHE.Enc}_{\mathsf{pk}}(y) \leftarrow \mathsf{QFHE.Eval}(\rho_{evk}, \mathcal{E}, \mathsf{QFHE.Enc}_{\mathsf{pk}}(x))$$

is a classical ciphertext of y. Finally, it should be possible to decrypt classical ciphertexts with a classical circuit, i.e. it should be possible to compute $\mathsf{Dec}_{\mathsf{sk}}(c)$ with a classical circuit if c is a classical ciphertext. In the QFHE schemes from [Mah18a, Bra18], ciphertexts are of the form $\mathsf{QFHE.Enc}_{\mathsf{pk}}(|\psi\rangle) = \mathsf{FHE.Enc}_{\mathsf{pk}}(a,b) \otimes X^a Z^b |\psi\rangle$ where X and Z are the Pauli operations and FHE is a classical FHE scheme. In particular, when $|\psi\rangle$ is a computational basis state, we have:

$$\mathsf{QFHE.Enc}_{\mathsf{pk}}(x) = \mathsf{FHE.Enc}_{\mathsf{pk}}(a,b) \otimes |a \oplus x\rangle$$

Ciphertexts of this form have a classical description: $(\mathsf{QFHE.Enc}_{\mathsf{pk}}(a,b), a \oplus x)$. It is also clear that decryption of these ciphertexts can be done classically by using $\mathsf{FHE.Dec}_{\mathsf{pk}}$.

Instantiation. The works of [Mah18a, Bra18] give lattice-based candidates for quantum fully homomorphic encryption schemes; we currently do not know how to base this on QLWE alone[2]. The desirable properties required from the quantum FHE schemes are satisfied by both candidates [Mah18a, Bra18].

2.4.4 Cryptographic Obfuscation

In this work, we use different notions of cryptographic obfucation. We review all the required notions below, but first we recall the functionality of obfuscation.

Definition 37 (Functionality of Obfuscation)**.** *Consider a class of circuits $\mathcal{C}$. An obfuscator $\mathcal{O}$ consists of two PPT algorithms* Obf *and* Eval *such that the following holds: for every $\lambda \in \mathbb{N}$, circuit $C \in \mathcal{C}$, $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$, we have $C(x) \leftarrow \mathsf{Eval}(\widetilde{C}, x)$ where $\widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C)$.*

Lockable Obfuscation

In the impossibility result, we will make use of program obfuscation schemes that are (i) defined for 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 38 (Quantum-Secure Lockable Obfuscation)**.** *An obfuscation scheme* $(\mathsf{LO.Obf}, \mathsf{LO.Eval})$ *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 functionality of obfuscation.*

[2]Brakerski [Bra18] 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].

- **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* $\alpha \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$,

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

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

Instantiation. The works of [WZ17, GKW17, GKVW20] construct a lockable obfuscation scheme based on polynomial-security of learning with errors (see Section 2.3). Since learning with errors is conjectured to be hard against QPT algorithms, the obfuscation schemes of [WZ17, GKW17, GKVW20] are also secure against QPT algorithms.

q-Input-Hiding Obfuscators

One of the main tools used in our construction is q-input-hiding obfuscators. The notion of input-hiding obfuscators was first defined in the classical setting by Barak, Bitansky, Canetti, Kalai, Paneth, and Sahai [BBC+14]. We adopt the same notion except that we require the security of the primitive to hold against QPT adversaries.

The notion of q-input-hiding obfuscators states that given an obfuscated circuit, it should be infeasible for a QPT adversary to find an accepting input; that is, an input on which the circuit outputs 1. Note that this notion is only meaningful for the class of evasive circuits.

The definition below is suitably adapted from [BBC+14]; in particular, our security should hold against QPT adversaries.

Definition 39 (q-Input-Hiding Obfuscators [BBC+14])**.** *An obfuscator* $\mathsf{qIHO} = (\mathsf{Obf}, \mathsf{Eval})$ *for a class of circuits associated with distribution* $\mathcal{D}_{\mathcal{C}}$ *is* **q-input-hiding** *if for every non-uniform QPT adversary* $\mathcal{A}$, *for every sufficiently large* $\lambda \in \mathbb{N}$,

$$\Pr\left[C(x) = 1 \;:\; \begin{array}{c} C \leftarrow \mathcal{D}_{\mathcal{C}}(\lambda), \\ \widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C), \\ x \leftarrow \mathcal{A}(1^\lambda, \widetilde{C}) \end{array}\right] \leq negl(\lambda).$$

Subspace Hiding Obfuscators

Another ingredient in our construction is subspace hiding obfuscation. Subspace hiding obfuscation is a notion of obfuscation introduced by Zhandry [Zha19], as a tool to build pulic-key quantum money schemes. This notion allows for obfuscating a circuit, associated with subspace A, that checks if an input vector belongs to this subspace A or not. In terms of security, we require that the obfuscation of this circuit

is indistinguishable from obfuscation of another circuit that tests membership of a larger random (and hidden) subspace containing A.

Definition 40 ([Zha19]). *A* **subspace hiding obfuscator** *for a field $\mathbb{F}$ and dimensions d_0, d_1, λ is a tuple $(\mathsf{shO.Obf}, \mathsf{shO.Eval})$ satisfying:*

- $\mathsf{shO.Obf}(A)$*: on input an efficient description of a linear subspace $A \subset \mathbb{F}^\lambda$ of dimensions $d \in \{d_0, d_1\}$ outputs an obfuscator $\mathsf{shO}(A)$.*
- **Correctness:** *For any A of dimension $d \in \{d_0, d_1\}$, it holds that*

$$\Pr[\forall x, \mathsf{shO.Eval}(\mathsf{shO}(A), x) = \mathbb{1}_A(x) : \mathsf{shO}(A) \leftarrow \mathsf{shO.Obf}(A)] \geq 1 - negl(\lambda),$$

  where: $\mathbb{1}_A(x) = 1$ if $x \in A$ and 0, otherwise.
- **Quantum-Security:** *Any QPT adversary $\mathcal{A}$ can win the following challenge with probability at most negligibly greater than $\frac{1}{2}$.*
  1. *$\mathcal{A}$ chooses a d_0-dimensional subspace $A \subset \mathbb{F}^\lambda$.*
  2. *Challenger chooses uniformly at random a d_1-dimensional subspace $S \supseteq A$. It samples a random bit b. If $b = 0$, it sends $\widetilde{g_0} \leftarrow \mathsf{shO.Obf}(A)$. Otherwise, it sends $\widetilde{g_1} \leftarrow \mathsf{shO.Obf}(S)$*
  3. *$\mathcal{A}$ receives $\widetilde{g_b}$ and outputs b'. It wins if $b' = b$.*

Instantiation. Zhandry presented a construction of subspace obfuscators from indistinguishability obfuscation [BGI+01, GGH+16] secure against QPT adversaries.

Quantum Virtual Black-Box Obfuscation (QVBB)

One of our results is that quantum virtual black-box obfuscation (QVBB) of classical circuits is impossible for arbitrary classical circuits. QVBB was introduce by Alagic and Fefferman [AF16], as a quantum generalization of VBB. For completeness, we present their definition.

Definition 41 (QVBB [AF16]). *A* **black-box quantum obfuscator** *is a tuple of QPT algorithms, $(\mathcal{O}, \mathcal{J})$ where:*

- *$\mathcal{O}$: takes as input an n-qubits quantum circuit, C, and outputs an $m = \mathrm{poly}(n)$-qubits quantum state.*
- *$\mathcal{J}$: takes as input a state $\mathcal{O}(C)$ and a state ρ, and attempts to outputs $U_C \rho U_C^\dagger$ where U_C is a unitary implementation of C.*

We say that $(\mathcal{O}, \mathcal{J})$ satisfies **functional equivalence** *if for all n-qubit quantum circuits C, and all n-qubit states ρ, the following holds*

$$\left\| \mathcal{J}(\mathcal{O}(C) \otimes \rho) - U_C \rho U_C^\dagger \right\|_{tr} \leq negl(n).$$

We say that $(\mathcal{O}, \mathcal{J})$ *satisfies* **virtual black-box** *security if for every QPT* $\mathcal{A}$ *there exists a QPT simulator* Sim *with quantum black-box access to* U_C *such that*

$$\left|\Pr\left[\mathcal{A}(\mathcal{O}(C)) = 1\right] - \Pr\left[\mathsf{Sim}^{U_C}(|0^{\otimes n}\rangle\langle 0^{\otimes n}|) = 1\right]\right| \leq negl(n)$$

2.4.5 Secure Function Evaluation (SFE)

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 42 (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,

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

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 43 (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_1^1. 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 and Döttling [BD18].

2.4.6 Non-Interactive Zero-Knowledge (NIZK)

One tool that we use in some of our constructions is that of non-interactive zero-knowledge (NIZK) proofs or arguments . A NIZK is defined between a classical PPT prover P and a verifier V. The goal of the prover is to convince the verifier V to accept an instance x using a witness w while at the same time, not revealing any information about w. Moreover, any malicious prover should not be able to falsely convince the verifier to accept a NO instance. Since we allow the malicious parties to be QPT, sometimes we term this NIZK as qNIZK.

We make use of two types of NIZK: (1) statistical ZK quantum argument system, and (2) simulation-extractable NIZKs. They both have the same syntax but different security properties. A NIZK (in common reference string model) for an NP relation is a triplet of PPT algorithms defined as follows:

- $\mathsf{CRSGen}(1^\lambda)$: On input security parameter λ, it outputs the common reference string crs. When crs is generated according to the uniform distribution, we call this the **common random string model**, and use Gen instead of CRSGen.

- $P(\mathsf{crs}, x, w)$: On input common reference string crs, NP instance x, witness w, it outputs the proof π.

- $V(\mathsf{crs}, x, \pi)$: On input common reference string crs, instance x, proof π, it outputs accept or reject. This is a deterministic algorithm.

We define NIZK argument systems in the common random string model below. In the next section, we define simulation-extractable NIZKs in the common reference string model.

Definition 44 (Non-interactive statistical ZK argument system)**.** *A* **NIZK argument system**, Π_{NIZK}, *for an NP relation* $\mathcal{R}(\mathcal{L})$ *is said to satisfy* **completeness** *if the following holds:*

- **Completeness:** *For all* $(x, w) \in \mathcal{R}(\mathcal{L})$,

$$\Pr\left[\begin{smallmatrix}\mathsf{crs}\leftarrow\mathsf{Gen}(1^\lambda)\\ \pi\leftarrow P(\mathsf{crs},x,w)\end{smallmatrix} : V(\mathsf{crs},x,\pi)=1\right]=1$$

Π_{NIZK} *is said to satisfy* **(quantum computational) soundness** *if the following holds:*

- **(Quantum Computational) Soundness:** *For all* $x \notin \mathcal{L}$*, for any QPT* P^* *and auxiliary* $\mathrm{poly}(\lambda)$*-qubits state* ρ*,*

$$\Pr\left[\begin{smallmatrix}\mathsf{crs}\leftarrow\mathsf{Gen}(1^\lambda)\\ \pi\leftarrow P^*(\mathsf{crs},x,\rho)\end{smallmatrix} : V(\mathsf{crs},x,\pi)=1\right]=\mathit{negl}(\lambda)$$

Π_{NIZK} *is said to satisfy* **statistical zero-knowledge** *if the following holds:*

- **Statistical Zero-Knowledge:** *There exists a QPT simulator* Sim *such that for all* $(x, w) \in \mathcal{R}(\mathcal{L})$*, the following two distributions are statistically close:*
  1. *Sample* $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda)$*, sample* $\pi \leftarrow P(\mathsf{crs}, x, w)$*. Output* (crs, π)*.*
  2. *Sample* $(\mathsf{crs}^*, \pi^*) \leftarrow \mathsf{Sim}(1^\lambda, x)$*. Output* (crs^*, π^*)*.*

Instantiation. The work of [PS19] shows how to construct statistical NIZK arguments for NP in the LWE. We note that the same construction and proof can be ported to the quantum setting to demonstrate a construction of statistical NIZK quantum argument system for NP from QLWE. A discussion on the quantum security of [PS19] can be found in [CVZ20].

2.4.7 Simulation-Extractable Non-Interactive Zero-Knowledge (seNIZK)

A simulation-extractable NIZK is a NIZK that satisfies a stronger property called simulation extractability. We call a NIZK satisfying this stronger property to be q-simulation-extractable NIZK ($\mathsf{qseNIZK}$).

Definition 45 (Completeness)**.** *A non-interactive protocol* $\mathsf{qseNIZK}$ *for a NP language* L *is said to be* **complete** *if the following holds: for every* $(x, w) \in \mathcal{R}(L)$*, we have the following:*

$$\Pr\left[V(\mathsf{crs},x,\pi)\ \textit{accepts}\ :\ \begin{smallmatrix}\mathsf{crs}\leftarrow\mathsf{CRSGen}(1^\lambda)\\ \pi\leftarrow P(\mathsf{crs},x,w)\end{smallmatrix}\right]=1$$

q-Simulation-Extractability. We now describe the simulation-extractability property. Suppose there exists an adversary who upon receiving many proofs $\pi_1, \ldots, \pi_q$ on all YES instances $x_1, \ldots, x_q$, can produce a proof π' on instance x' such that: (a) x' is different from all the instances $x_1, \ldots, x_q$ and, (b) π' is accepting with probability ε. Then, this notion guarantees the existence of two efficient algorithms Sim_1 and Sim_2 such that all the proofs $\pi_1, \ldots, \pi_q$, are now simulated by Sim_1, and Sim_2 can extract a valid witness for x' from (x', π') produced by the adversary with probability negligibly close to ε.

Definition 46 (q-Simulation-Extractability)**.** *A non-interactive protocol* qseNIZK *for a language L is said to satisfy* **q-simulation-extractability** *if there exists a non-uniform QPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such that the following holds:*

$$\Pr\left[\begin{array}{c} V(\mathsf{crs},x',\pi') \ \ accepts \\ \wedge \\ (\forall i\in[q],(x_i,w_i)\in\mathcal{R}(\mathcal{L})) \\ \wedge \\ (\forall i\in[q],\ x'\neq x_i) \end{array} : \begin{array}{c} \mathsf{crs}\leftarrow\mathsf{CRSGen}(1^\lambda), \\ \left(\{(x_i,w_i)\}_{i\in[q]},\mathsf{st}_{\mathcal{A}}\right)\leftarrow\mathcal{A}_1(\mathsf{crs}) \\ \forall i\in[q],\ \pi_i\leftarrow P(\mathsf{crs},\mathsf{td},x_i) \\ (x',\pi')\leftarrow\mathcal{A}_2(\mathsf{st}_{\mathcal{A}},\pi_1,\ldots,\pi_q) \end{array}\right] = \varepsilon$$

Then there exists QPT algorithms FkGen *and* $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *such that the following holds:*

$$\left|\Pr\left[\begin{array}{c} V(\mathsf{crs},x',\pi') \ \ accepts \\ \wedge \\ (\forall i\in[q],(x_i,w_i)\in\mathcal{R}(\mathcal{L})) \\ \wedge \\ (x',w')\in\mathcal{R}(L) \\ \wedge \\ (\forall i\in[q],\ x'\neq x_i) \end{array} : \begin{array}{c} (\mathsf{crs},\mathsf{td})\leftarrow\mathsf{FkGen}(1^\lambda), \\ \left(\{(x_i,w_i)\}_{i\in[q]},\mathsf{st}_{\mathcal{A}}\right)\leftarrow\mathcal{A}_1(\mathsf{crs}) \\ (\pi_1,\ldots,\pi_q,\mathsf{st}_{\mathsf{Sim}})\leftarrow\mathsf{Sim}_1\left(\mathsf{crs},\mathsf{td},\{x_i\}_{i\in[q]}\right) \\ (x',\pi')\leftarrow\mathcal{A}_2(\mathsf{st}_{\mathcal{A}},\pi_1,\ldots,\pi_q) \\ w'\leftarrow\mathsf{Sim}_2(\mathsf{st}_{\mathsf{Sim}},x',\pi') \end{array}\right] - \varepsilon\right| \leq negl(\lambda)$$

We call a non-interactive argument system satisfying q-simulation-extractability property to be a qseNIZK system.

If q-simulation-extractability property holds against quantum adversaries running in time $2^{\tilde{O}(T)}$ ($\tilde{O}(\cdot)$ notation suppresses additive factors in $O(\log(\lambda))$) then we say that (CRSGen, P, V) is a T-sub-exponential qseNIZK system.

Remark 47. *The definition as stated above is weaker compared to other definitions of simulation-extractability considered in the literature. For instance, we can consider general adversaries who also can obtain simulated proofs for false statements which is disallowed in the above setting. Nonetheless, the definition considered above is sufficient for our application.*

Instantiation of qseNIZKs. In the classical setting, simulation-extractable NIZKs can be obtained by generically [Sah99, DSDCO+01] combining a traditional NIZK (satisfying completeness, soundness and zero-knowledge) with a public-key encryption scheme satisfying CCA2 security. We observe that the same transformation can be ported to the quantum setting as well, by suitably instantiating the underlying

primitives to be quantum-secure. These primitives in turn can be instantiated from QLWE. Thus, we can obtain a q-simulation-extractable NIZK from QLWE.

For our construction of SSL, it turns out that we need a q-simulation-extractable NIZK that is secure against quantum adversaries running in sub-exponential time. Fortunately, we can still adapt the same transformation but instead instantiating the underlying primitives to be sub-exponentially secure.

In Appendix A we prove the following lemma.

Lemma 48. *Consider a language $\mathcal{L}_\ell \in NP$ such that every $x \in \mathcal{L}_\ell$ is such that $|x| = \ell$.*

Under the ℓ-sub-exponential QLWE assumption, there exists a q-simulation-extractable NIZKs for $\mathcal{L}_\ell$ satisfying perfect completeness.

2.4.8 Witness Indistinguishability (WI)

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

Definition 49 (Quantum WI for an $\mathcal{L} \in$ NP)**.** *An argument or proof system (P, V) for an NP language $\mathcal{L}$ is* **quantum witness indistinguishable** *if the following hold.*

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

$$\mathsf{View}_{V^*}(\langle P(x, \mathbf{w}_1), V^*(x, \cdot)\rangle) \approx_Q \mathsf{View}_{V^*}(\langle P(x, \mathbf{w}_2), V^*(x, \cdot)\rangle).$$

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.

2.4.9 Post-Quantum Statistical Sender-Private OT

The tool we use in this construction is a two-round oblivious transfer protocol that has computational security against senders and statistical security against receivers. We define this tool below. We instantiate this primitive with the QLWE-based construction in [BD18].

Definition 50 (Post-Quantum Statistical Sender-Private OT)**.** *A* **two-round oblivious transfer** *is a tuple of algorithms $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ which specifies the following protocol.*

Round 1. *The receiver R, on input security parameter λ, bit β, computes $(\mathsf{ot}_1, \mathsf{st}_R) \leftarrow \mathsf{OT}_1(1^\lambda, \beta)$ and sends ot_1 to the sender S.*

Round 2. *The sender* S*, on input* ot_1 *and message bits* (m_0, m_1)*, computes* $\mathsf{ot}_2 \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{OT}_1, (m_0, m_1))$*. It sends* ot_2 *to the receiver* R*.*

Reconstruction. *The receiver computes* $m' \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_1, \mathsf{ot}_2, \mathsf{st}_R)$*.*

Correctness. *For any* $\beta \in \{0,1\}, (m_0, m_1) \in \{0,1\}^2$*, we have:*

$$\mathsf{Pr}\left[\begin{smallmatrix}(\mathsf{ot}_1,\mathsf{st}_R)\leftarrow\mathsf{OT}_1(1^\lambda,\beta)\\ \mathsf{ot}_2\leftarrow\mathsf{OT}_2(1^\lambda,\mathsf{ot}_1,(m_0,m_1))\\ m'\leftarrow\mathsf{OT}_3(1^\lambda,\mathsf{ot}_1,\mathsf{ot}_2,\mathsf{st}_R)\end{smallmatrix} \;:\; m' = m_\beta\right] = 1$$

Post-Quantum Receiver-Privacy. *The following holds:*

$$\{\mathsf{OT}_1(1^\lambda, 0)\} \approx_{c,Q} \{\mathsf{OT}_1(1^\lambda, 1)\}$$

Statistical Sender-Privacy. *There exists a computationally unbounded extractor such that for every the first round message* ot_1*, it outputs a bit* $b \in \{0,1\}$ *such that the following holds for every* $(m_0, m_1) \in \{0,1\}$*:*

$$\mathsf{SD}(\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, (m_0, m_1)), \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, (m_b, m_b))) \leq \mathit{negl}(\lambda),$$

where SD *denotes statistical distance and negl is a negligible function.*

Chapter 3

Quantum Extraction Protocols

The main technique used through this thesis is that of extraction. Concretely, we use different techniques to extract information from (possibly obfuscated) QPT circuits. In the QZK case, our goal is to extract from malicious QPT verifiers, V^*. In the copy-protection case, we extract from efficient QPT circuits that compute a boolean circuit C. Extraction by itself might not be very useful (e.g. example in Figure 1-2), but in follow-up chapters we will see how to leverage the ideas introduce in the context of extraction to achieve primitives like QZK and de-quantumizable circuits. For this reason, we believe that extraction techniques are worth studying by themselves especially in the quantum setting where rewinding is much more restricted. To formally study extraction in the quantum setting, we introduce the notion of secure quantum extraction protocols (QEXT).

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 as input the instance x and witness $\mathbf{w}$ while the receiver only gets the instance x as input. There are two properties associated with a secure quantum extraction protocol: (a) *Extractability*: for any efficient quantum polynomial-time (QPT) adversarial sender, there exists a QPT extractor that can extract a witness $\mathbf{w}'$ such that $(x, \mathbf{w}') \in \mathcal{R}$ and, (b) *Zero-Knowledge*: a malicious receiver, interacting with the sender, should not be able to learn any information about $\mathbf{w}$.

We study and construct two flavors of secure quantum extraction protocols.

- **Security against QPT malicious receivers** (qQEXT): We consider the setting when the malicious receiver is a QPT adversary. In this setting, we construct a secure quantum extraction protocol for NP assuming the existence of QFHE satisfying some mild properties (already satisfied by existing constructions [Mah18a, Bra18]) and QLWE. The novelty of our construction is a new non-black-box technique in the quantum setting.

- **Security against classical PPT malicious receivers** (cQEXT): We also consider the setting when the malicious receiver is a classical probabilistic polynomial time (PPT) adversary. In this setting, we construct a secure quantum extraction protocol for NP solely based on QLWE. Furthermore, our construction satisfies *quantum-lasting security*: 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.

Both the above extraction protocols are *constant round* protocols. We will use the cQEXT construction in Chapter 4 to achieve a constant round QZK classical argument systems for NP. The ideas from the qQEXT protocol will be used to prove the impossibility of copy-protection in Chapter 6.

Extraction protocols are intended to be used as ingredients in other (bigger) protocols whenever they can be used for the following two properties.

- *Extractability*: A QPT algorithm (the extractor) can extract a valid witness from an adversarial sender. We model the adversarial sender as a QPT 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[1].

  We also require *indistinguishability of extraction*: that is, the adversarial sender cannot distinguish whether it's interacting with the honest receiver or an extractor. In QZK applications, this property is used to argue that the adversary cannot distinguish whether it's interacting with the honest party or the simulator.

- *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 receivers (qQEXT) and quantum extraction protocols secure against classical adversarial receivers (cQEXT).

There are two reasons why we only formalize and study extraction against semi-malicious adversaries, instead of malicious adversaries (who can arbitrarily deviate from the protocol): first, even extracting from semi-malicious adversaries turns out to be challenging and we view this as a first step towards extraction from malicious adversaries and second, in the classical setting, there are works that show how to leverage extraction from semi-malicious adversaries to achieve zero-knowledge protocols [BCPR16, BKP19] or secure two-party computation protocols [AJ17].

Quantum extraction protocols are interesting even if we only consider classical adversaries, as they present a new method for proving quantum zero-knowledge. For instance, to demonstrate zero-knowledge, we need to demonstrate a simulator that has a computational capability that a malicious prover doesn't have. 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

[1]In the literature, this type of semi-malicious adversaries are also referred to as *explainable* adveraries.

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, 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 available in the future. First introduced by Unruh [Unr13], we study the concept of quantum-lasting security; any QPT adversary given the transcript and the private state of the malicious receiver, should not be able to learn the witness of the sender. Our construction will satisfy this security notion and thus our protocol is resilient against the possibility of quantum computers being accessible in the future.

3.1 QEXT definitions

The definition of QEXT provided below resembles the concept of zero-knowledge argument of knowledge (ZKAoK) systems. There are two important differences:

- Firstly, we do not impose any completeness requirement on our extraction 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 51 (Quantum extraction protocols secure against quantum adversaries)**.** *A* **quantum extraction protocol secure against quantum adversaries***, denoted by* qQEXT *is a classical protocol between two classical PPT algorithms, sender* S *and a receiver* R *and is associated with an* NP *relation* $\mathcal{R}$*. The input to both the parties is an instance* $x \in \mathcal{R}(\mathcal{L})$*. In addition, the sender also gets as input the witness* $\mathbf{w}$ *such that* $(x, \mathbf{w}) \in \mathcal{R}$*. 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* $(x, \mathbf{w}) \in \mathcal{R}$*, 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,*

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

- **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 $(x, \mathbf{w}) \in \mathcal{R}(\mathcal{L})$, for every semi-malicious*[2] *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:*

  - **Indistinguishability of Extraction**:

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

  - *The probability that Ext_2 outputs $\mathbf{w}'$ such that $(x, \mathbf{w}') \in \mathcal{R}$ is negligibly close to 1.*

Definition 52 (Quantum extraction protocols secure against classical adversaries)**.** *A* **quantum extraction protocol secure against classical adversaries** cQEXT *is defined the same way as in Definition 51 except that instead of quantum zero-knowledge,* cQEXT *satisfies classical zero-knowledge property 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 $(x, \mathbf{w}) \in \mathcal{R}$, 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*

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

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 along with its own private state. We call this property *quantum-lasting security*; first introduced by Unruh [Unr13]. We formally define quantum-lasting security below.

Definition 53 (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 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:*

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

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

3.2 cQEXT

3.2.1 Overview

We start with the overview of quantum extraction protocols with security against classical receivers.

Starting Point: Noisy Trapdoor Claw-Free Functions. 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 functions, denoted by $f_{\mathbf{k},0}(x) \in \mathcal{Y}$ and $f_{\mathbf{k},1}(x) \in \mathcal{Y}$ [3], 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 $f_{\mathbf{k},0}(x_0) = f_{\mathbf{k},1}(x_1)$. 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].

Test of Quantumness [BCM+18]. Using NTCFs, [BCM+18] devised the following test[4]:

- The classical client, who wants to test whether the server it's 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.
- The server responds back with $y \in \mathcal{Y}$.
- The classical client then sends a **challenge** bit $\mathbf{a}$ to the server.
- If $\mathbf{a} = 0$, the server sends a pre-image x_b along with bit b such that $f_{\mathbf{k},b}(x_b) = y$. If $\mathbf{a} = 1$, the server sends a vector d along with a bit u satisfying the condition $\langle d, J(x_0) \oplus J(x_1)\rangle = u$, where x_0, x_1 are such that $f_{\mathbf{k},0}(x_0) = f_{\mathbf{k},1}(x_1) = y$.

The client can check if the message sent by the server is either a valid pre-image or a valid d that is correlated with respect to both the pre-images.

Intuitively, since the (classical) 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 client; thus it will always pass the test.

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

[4] As written, this test doesn't have negligible soundness but we can achieve negligible soundness by parallel repetition.

From Test of Quantumness to Extraction. 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. We sketch this attempt below.

- Sender on input instance-witness pair $(x, \mathbf{w})$ and receiver on input instance x run a "test of quantumness" protocol where the receiver (taking the role of the server) needs to convince the sender (taking the role of the classical client) that it is a quantum computer.

- If the receiver succeeds in the "test of quantumness" protocol then the sender sender $\mathbf{w}$, else it aborts.

Note that a quantum extractor can indeed succeed in the test of quantumness protocol and hence, it would receive $\mathbf{w}$ while a malicious classical adversary will not.

However, the above solution is not good enough for us. It does not satisfy indistinguishability of extraction: the sender can detect whether it's interacting with a quantum extractor or an honest receiver.

Achieving Indistinguishability of Extraction. To ensure indistinguishability of extraction, we rely upon a tool called secure function evaluation [GHV10, BCPR16] that satisfies quantum security. A secure function evaluation (SFE) allows for two parties P_1 and P_2 to securely compute a function on their inputs in a such a way that only one of the parties, say P_2, receives the output of the function. In terms of security, we require that: (i) P_2 doesn't get information about P_1's input beyond the output of the function and, (ii) P_1 doesn't get any information about P_2's input (in fact, even the output of the protocol is hidden from P_1).

The hope is that by combining SFE and test of quantumness protocol, we can guarantee that a quantum extractor can still recover the witness by passing the test of quantumness as before but the sender doesn't even know whether the receiver passed or not. To implement this, we assume a structural property from the underlying test of quantumness protocol: until the final message of the protocol, the client cannot distinguish whether it's talking to a quantum server or a classical server. This structural property is satisfied by the test of quantumness protocol [BCM+18] sketched above.

Using this structural property and SFE, here is another attempt to construct a quantum extraction protocol: let the test of quantumness protocol be a k-round protocol.

- Sender on input instance-witness pair $(x, \mathbf{w})$ and receiver on input instance x run the first $(k-1)$ rounds of the test of quantumness protocol where the receiver (taking the role of the server) needs to convince the sender (taking the role of the receiver) that it can perform quantum computations.

- Sender and receiver then run a SFE protocol for the following functionality G: it takes as input $\mathbf{w}$ and the first $(k-1)$ rounds of the test of quantumness

protocol from the sender, the k^{th} round message from the receiver[5] and outputs $\mathbf{w}$ if indeed the test passed, otherwise output $\perp$. Sender will take the role of P_1 and the receiver will take the role of P_2 and thus, only the receiver will receive the output of G.

Note that the security of SFE guarantees that the output of the protocol is hidden from the sender and moreover, the first $(k-1)$ messages of the test of quantumness protocol doesn't reveal the information about whether the receiver is a quantum computer or not. These two properties ensure the sender doesn't know whether the receiver passed the test or not. Furthermore, the quantum extractor still succeeds in extracting the witness $\mathbf{w}$ since it passes the test.

The only remaining property to prove is zero-knowledge.

Challenges in Proving Zero-Knowledge. How do we ensure that a malicious classical receiver was not able to extract the witness? The hope would be to invoke the soundness of the test of quantumness protocol to argue this. However, to do this, we need all the k messages of the test of quantumness protocol.

To understand this better, let us recall how the soundness of the test of quantumness works: the client sends a challenge bit $\mathbf{a} = 0$ to the server who responds back with (b, x_b), then the client rewinds the server and instead sends the challenge bit $\mathbf{a} = 1$ and it receives (d, u): this contradicts the security of NTCFs since a classical PPT adversary cannot simultaneously produce both a valid pre-image (b, x_b) and a valid correlation vector along with the prediction bit (d, u).

Since the last message is fed into the secure function evaluation protocol and inaccessible to the simulator, we cannot use this rewinding strategy to prove the zero-knowledge of the extraction protocol.

Final Template: Zero-Knowledge via Extractable Commitments [PRS02, PW09]. To overcome this barrier, we force the receiver to commit, using an extractable commitment scheme, to the k^{th} round of the test of quantumness protocol before the SFE protocol begins. An extractable commitment scheme is one where there is an extractor who can extract an input x being committed from the party committing to x. Armed with this tool, we give an overview of our construction below.

- Sender on input instance-witness pair $(x, \mathbf{w})$ and receiver on input instance x run the first $(k-1)$ rounds of the test of quantumness protocol where the receiver (taking the role of the server) needs to convince the sender (taking the role of the receiver) that it can perform quantum computations.
- The k^{th} round of the test of quantumness protocol is then committed by the receiver, call it $\mathbf{c}$, using the extractable commitment scheme[6].

[5] It follows without loss of generality that the server (and thus, the receiver of the quantum extraction protocol) computes the final message of the test of quantumness protocol.

[6] In the technical sections, we use a specific construction of extractable commitment scheme by [PRS02, PW09] since we additionally require security against quantum adversaries.

- Finally, the sender and the receiver then run a SFE protocol for the following functionality G: it takes as input $\mathbf{w}$ and the first $(k-1)$ rounds of the test of quantumness protocol from the sender, the decommitment of $\mathbf{c}$ from the receiver and outputs $\mathbf{w}$ if indeed the test passed, otherwise output $\perp$. Sender will take the role of P_1 and the receiver will take the role of P_2 and thus, only the receiver will receive the output of G.

Let us remark about zero-knowledge since we have already touched upon the other properties earlier. To argue zero-knowledge, we construct a simulator that interacts honestly with the malicious receiver until the point the extraction protocol is run. Then, the simulator runs the extractor of the commitment scheme to extract the final message of the test of quantumness protocol. It then rewinds the test of quantumness protocol to the point where the simulator sends a different challenge bit (see the informal description of [BCM+18] given before) and then runs the extractor of the commitment scheme once again to extract the k^{th} round message of the test of quantumness protocol. Recall that having final round messages corresponding to two different challenge bits is sufficient to break the security of NTCFs; the zero-knowledge property then follows.

A couple of remarks about our simulator. Firstly, the reason why our simulator is able to rewind the adversary is because the adversary is a classical PPT algorithm. Secondly, our simulator performs *double rewinding* – not only does the extractor of the commitment scheme perform rewinding but also the test of quantumness protocol is rewound.

3.2.2 Construction of cQEXT

In this section, we show how to construct quantum extraction protocols secure against classical adversaries based solely on QLWE.

Tools.

- Quantum-secure computationally-hiding and perfectly-binding non-interactive commitments, Comm (Section 2.4.2).

  We instantiate the underlying commitment scheme in [PW09] using Comm to obtain a quantum-secure extractable commitment scheme. Instead of presenting a definition of quantum-secure extractable commitment scheme and then instantiating it, we directly incorporate the construction of [PW09] in the construction of the extraction protocol.

- Noisy trapdoor claw-free functions $\{f_{\mathbf{k},b} : \mathcal{X} \to D_{\mathcal{Y}}\}_{\mathbf{k}\in\mathcal{K},b\in\{0,1\}}$ (Section 2.4.1).

- Quantum-secure secure function evaluation protocol $\mathsf{SFE} = (\mathsf{SFE.S}, \mathsf{SFE.R})$ (Section 2.4.5).

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 $(x_{i,0}, x_{i,1}) \leftarrow \mathsf{Inv}(\mathbf{k}_i, \mathsf{td}_i, y_i)$.
  - *Check if the commitments commit to the same message*: Output $\perp$ if the following does not hold: for every $j, j' \in [k]$, we have $\left(sh_{i,w_i}^{(j)}\right)' \oplus sh_{i,w_i}^{(j)} = \left(sh_{i,w_i}^{(j')}\right)' \oplus sh_{i,w_i}^{(j')}$.
  - 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. Since $J(\cdot)$ can be efficiently inverted, recover x'_{i,b_i}. If $x'_{i,b_i} \neq x_{i,b_i}$, output $\perp$.
  - If $v_i = 1$: let $(u_i, d_i) = \left(sh_{i,w_i}^{(j)}\right)' \oplus sh_{i,\overline{w_i}}^{(j)}$. 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 3-1: Description of the function $\mathbf{F}$ associated with the SFE

Construction. We present the construction of the quantum extraction protocol (S, R) in Figure 3-2 for an NP language $\mathcal{L}$.

Lemma 54. *Assuming the quantum security of* $\mathsf{Comm}, \mathsf{SFE}$ *and NTCFs, the protocol* (S, R) *is a quantum extraction protocol secure against classical adversaries for* NP*, and it is also quantum-lasting secure.*

Proof.

Classical Zero-Knowledge. Let R^* be a classical PPT algorithm. We first describe a classical simulator Sim such that R^* cannot distinguish whether it's interact-

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

- $\underline{\mathsf{S} \to \mathsf{R}}$: Compute $\forall i \in [k], (\mathbf{k}_i, \mathsf{td}_i) \leftarrow \mathsf{Gen}(1^\lambda; r_i)$, where $k = \lambda$. Send $\left(\{\mathbf{k}_i\}_{i\in[k]}\right)$.

- $\underline{\mathsf{R} \to \mathsf{S}}$: 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 $\mathcal{Y}$.)

- $\underline{\mathsf{S} \to \mathsf{R}}$: Send bits $(v_1, \ldots, v_k)$, where $v_i \xleftarrow{\$} \{0,1\}$ for $i \in [k]$.

- $\underline{\mathsf{R} \to \mathsf{S}}$: For every $i, j \in [k]$, compute the commitments $\mathbf{c}^{(j)}_{i,0} \leftarrow \mathsf{Comm}(1^\lambda, sh^{(j)}_{i,0}; \mathbf{d}^{(j)}_{i,0})$ and $\mathbf{c}^{(j)}_{i,1} \leftarrow \mathsf{Comm}(1^\lambda, sh^{(j)}_{i,1}; \mathbf{d}^{(j)}_{i,1})$, where $sh^{(j)}_{i,0}, sh^{(j)}_{i,1} \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$ for $i, j \in [k]$. Send $\left(\left\{\mathbf{c}^{(j)}_{i,0}, \mathbf{c}^{(j)}_{i,1}\right\}_{i,j\in[k]}\right)$.

  Note: The reason why we have k^2 commitments above is because we repeat (in parallel) the test of quantumness protocol k times and for each repetition, the response of the receiver is committed using k commitments; the latter is due to [PW09].

- $\underline{\mathsf{S} \to \mathsf{R}}$: For every $i, j \in [k]$, send random bits $w^{(j)}_i \in \{0,1\}$.

- $\underline{\mathsf{R} \to \mathsf{S}}$: Send $\left(\left\{(sh^{(j)}_{i,w_i})', (\mathbf{d}^{(j)}_{i,w_i})'\right\}_{i,j\in[k]}\right)$.

- S and R run SFE, associated with the two-party functionality $\mathbf{F}$ defined in Figure 3-1; S takes the role of $\mathsf{SFE.S}$ and R takes the role of $\mathsf{SFE.R}$. The input to $\mathsf{SFE.S}$ is $\left(\left\{\mathbf{c}^{(j)}_{i,0}, \mathbf{c}^{(j)}_{i,1}, (sh^{(j)}_{i,w_i})', (\mathbf{d}^{(j)}_{i,w_i})', \mathsf{td}_i, \mathbf{k}_i, y_i, v_i, w^{(j)}_i\right\}_{i,j\in[k]}, \mathbf{w}\right)$ and the input to $\mathsf{SFE.R}$ is $\left(\left\{sh^{(j)}_{i,\overline{w_i}}, \mathbf{d}^{(j)}_{i,\overline{w_i}}\right\}_{i,j\in[k]}\right)$.

Figure 3-2: Quantum Extraction Protocol (S, R) secure against classical receivers

ing with S or with Sim.

Description of Sim.

- Until the SFE protocol is executed, it behaves as the honest sender would. That is,

- 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 $\left(\left\{\mathbf{c}^{(j)}_{i,0}, \mathbf{c}^{(j)}_{i,1}\right\}_{i,j\in[k]}\right)$ from R^*.
- For every $i, j \in [k]$, it sends random bits $w_i^{(j)} \in \{0,1\}$.
- It receives $\left(\left\{(sh^{(j)}_{i,w_i})', (\mathbf{d}^{(j)}_{i,w_i})'\right\}_{i,j\in[k]}\right)$ from R^*.

- It then executes SFE with R^*, associated with the two-party functionality $\mathbf{F}$ defined in Figure 3-1; the input of Sim in SFE is $\perp$.

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

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

In other words, that no QPT distinguisher can distinguish between the view of R^* when interacting with S from the output of Sim. This is stronger than what we need to argue classical ZK, as it would be enough to show that R^*, a PPT machine (not QPT), cannot distinguish. However, the stronger indistinguishability result makes it easier to show that the scheme is quantum-lasting secure.

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

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

1. R^*: Sends $\{y_i\}_{i\in[k]}$.
2. $\mathsf{Hybrid}_2.\mathsf{S}$: Sends $(v_1, \ldots, v_k)$ uniformly at random. If R^* aborts in this step, $\mathsf{Hybrid}_2.\mathsf{S}$ aborts.
3. R^*: Sends $\left\{\left(\mathbf{c}^{(j)}_{i,0}, \mathbf{c}^{(j)}_{i,1}\right)\right\}_{i,j\in[k]}$. If R^* aborts in this step, $\mathsf{Hybrid}_2.\mathsf{S}$ aborts.
4. $\mathsf{Hybrid}_2.\mathsf{S}$: Sends $w_i^{(j)} \in \{0,1\}$ uniformly at random for all $i, j \in [k]$.
5. R^*: Opens up the commitments queried, $\left\{\left(sh^{(j)}_{i,w_i}, \mathbf{d}^{(j)}_{i,w_i}\right)\right\}_{i,j\in[k]}$. If R^* aborts in this step, $\mathsf{Hybrid}_2.\mathsf{S}$ aborts. If $\mathbf{c}^{(j)}_{i,w_i} \neq \mathsf{Comm}(1^\lambda, sh^{(j)}_{i,w_i}; \mathbf{d}^{(j)}_{i,w_i})$ for any $i, j \in [k]$, continue the execution of the protocol as in Step 11.
6. $\mathsf{Hybrid}_2.\mathsf{S}$: 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 $\mathbf{c}^{(j)}_{i,w_i} = \mathsf{Comm}(1^\lambda, sh^{(j)}_{i,w_i}; \mathbf{d}^{(j)}_{i,w_i})$. 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 $\mathsf{Hybrid}_2.\mathsf{S}$ 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. $\mathsf{Hybrid}_2.\mathsf{S}$: (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. $\mathsf{Hybrid}_2.\mathsf{S}$: 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 $\mathsf{Hybrid}_2.\mathsf{S}$ 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 $\mathsf{Hybrid}_2.\mathsf{S}$ queries $(w_1, \ldots, w_k)$ and $(v_1, \ldots, v_k)$.

The only difference between Hybrid_1 and Hybrid_2 is that $\mathsf{Hybrid}_2.\mathsf{S}$ aborts on some transcripts; conditioned on $\mathsf{Hybrid}_2.\mathsf{S}$ not aborting, the transcript produced by the receiver when interacting with S is identical to the transcript produced by $\mathsf{Hybrid}_2.\mathsf{S}$. We claim that the probability that $\mathsf{Hybrid}_2.\mathsf{S}$ aborts, conditioned on the event that R^* does not abort, is negligibly small.

Claim 55. $\Pr[\mathsf{Hybrid}_2.\mathsf{S}$ *aborts*$|\mathsf{R}^*$ *does not abort*$] = negl(k)$

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 56. *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 57. *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 $\mathsf{Hybrid}_2.\mathsf{S}$ 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 $\mathsf{Hybrid}_2.\mathsf{S}$ aborts on an NTCF transcript (Step 9) and the probability that it aborts on the transcript of Comm (Step 6).

Claim 58. *The probability that $\mathsf{Hybrid}_2.\mathsf{S}$ aborts in Step 9 is negligible.*

Proof. Note that $\mathsf{Hybrid}_2.\mathsf{S}$ 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:

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

□

Claim 59. *The probability that $\mathsf{Hybrid}_2.\mathsf{S}$ 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 $\mathsf{Hybrid}_2.\mathsf{S}$ aborts in Step 6 is negligible. Since we already argued in Claim 57 that the number of inner rewinding operations is $\mathrm{poly}(k)$, by union bound, it suffices to argue the probability that for any given NTCF transcript, the probability that $\mathsf{Hybrid}_2.\mathsf{S}$ aborts in Step 6 is negligible. This is similar to the argument in Claim 58: the probability that $\mathsf{Hybrid}_2.\mathsf{S}$ 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 $\mathsf{Hybrid}_2.\mathsf{S}$ 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 Hybrid_1 and Hybrid_2 are indistinguishable:

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

$\underline{\mathsf{Hybrid}_3}$: In this hybrid, $\mathsf{Hybrid}_3.\mathsf{S}$ will do as $\mathsf{Hybrid}_2.\mathsf{S}$ 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 Hybrid_2 and Hybrid_3 follows from the security of the SFE against malicious quantum receivers, and we have:

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

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

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

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

We have the following:

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

This is because either $\mathsf{Hybrid}_3.\mathsf{S}$ inputs $\perp$ into the SFE or it can find $T = \{(b_i, x_i, u_i, d_i) : i \in S\}$ (see Hybrid_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.

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

The statistical distance between Hybrid_4 and Hybrid_5 is negligible in k; this follows from Claim 55.

Quantum-Lasting Security. We have shown that for any auxiliary information $\mathsf{aux} \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

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

Let $\mathcal{A}^*$ be any QPT adversary that is given the transcript, $\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{S}(1^\lambda, x, \mathbf{w}), \mathsf{R}^*(1^\lambda, x, \mathsf{aux})\rangle\right)$. Consider the Sim^* that first runs $\mathsf{Sim}(1^\lambda, \mathsf{R}^*, x, \mathsf{aux})$, and then runs $\mathcal{A}^*$, i.e. Sim^* is the QPT that on a polynomial sized quantum states ρ acts as

$$\mathsf{Sim}^*\left(1^\lambda, \mathcal{A}^*, \mathsf{R}^*, x, \mathsf{aux}, \rho\right) = \mathcal{A}^*\left(\mathsf{Sim}(1^\lambda, \mathsf{R}^*, x, \mathsf{aux}), \rho\right).$$

Since $\mathcal{A}^*$ is QPT, it can't distinguish if it is given the actual transcript or the output of Sim. In particular, we have that

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

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 x.

- 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, $|b,x,0\rangle \to |b,x,J(x)\rangle$, and then uncompute the register containing x by performing J^{-1}, i.e. $|b,x,J(x)\rangle \to |b, x \oplus J^{-1}(J(x)), J(x)\rangle$. The resulting transformation is $|b,x,0\rangle \to |b,0,J(x)\rangle$.

– 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 60. *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 61. $\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, x, \mathbf{w}, \cdot), \mathsf{R}(1^\lambda, x)\rangle\right) \approx_Q \mathsf{Ext}_1\left(1^\lambda, \mathsf{S}^*, x, \cdot\right)$

Proof. Consider the following hybrids.

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

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

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

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

$\underline{\mathsf{Hybrid}_3}$: We define a hybrid receiver $\mathsf{Hybrid}_3.\mathsf{R}$ that behaves as $\mathsf{Hybrid}_2.\mathsf{R}$, 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 Hybrid_2 and Hybrid_3. To see this, note that Hybrid_3 samples from the distribution y_i from the distribution: $\frac{1}{2|\mathcal{X}|}\sum_{b\in\{0,1\},x\in X} f'_{\mathbf{k}_i,b}(x)(y)$. 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 $f'_{\mathbf{k}_i,b}(x_{i,b})$.

$\underline{\mathsf{Hybrid}_4}$: We define a hybrid receiver $\mathsf{Hybrid}_4.\mathsf{R}$ 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, $\mathsf{Hybrid}_4.\mathsf{R}$ compute correct answers to the test of quantumness, then it commits to appropriate shares,

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

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

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

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

$\underline{\mathsf{Hybrid}_5}$: We define a hybrid receiver $\mathsf{Hybrid}_5.\mathsf{R}$ who sets the input in SFE to be $\left(\left\{sh^{(j)}_{i,\overline{w_i}}, \mathbf{d}^{(j)}_{i,\overline{w_i}}\right\}_{i\in[k]}\right)$, 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 $\mathsf{Hybrid}_5.\mathsf{R}$ is identical to that of the extractor Ext.

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

$$\begin{aligned}\mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, x, \mathbf{w}, \cdot), \mathsf{Hybrid}_4.\mathsf{R}(1^\lambda, x)\rangle\right) &\approx_Q \mathsf{View}_{\mathsf{S}^*}\left(\langle \mathsf{S}^*(1^\lambda, x, \mathbf{w}, \cdot), \mathsf{Hybrid}_5.\mathsf{R}(1^\lambda, x)\rangle\right) \\ &\equiv \mathsf{Ext}_1\left(1^\lambda, \mathsf{S}^*, x, \cdot\right)\end{aligned}$$

□

□

Indistinguishability of Extraction Against Malicious Senders. We observe that our construction satifies a stronger property than claimed. Our protocol satisfies indistinguishability of extraction against *malicious* senders, and not just semi-malicious senders. However, the extractability is still required against semi-malicious senders.

We formalize this in the claim below.

Claim 62. *The quantum extraction protocol (S, R) described in Figure 3-2 satisfies indistinguishability of extraction (Definition 51) against malicious senders.*

We omit the proof of the above claim since it is identical to the proof of Claim 61. The indistinguishability of the hybrids in the proof of Claim 61 already hold against malicious senders; in the proof, we never used the fact that the sender was semi-malicious.

The only caveat missing in the proof of Claim 61 but comes up in the proof of the above claim is the fact that the malicious sender could abort. If the malicious sender aborts, then so does the extractor; since the extractor is straightline, the view of the sender until that point will still be indistinguishable from the view of the sender when interacting with the honest receiver.

3.3 qQEXT

3.3.1 Overview

We show how to construct extraction protocols where we prove security against quantum receivers. At first sight, it might seem that quantum extraction and quantum zero-knowledge properties are contradictory since the extractor has the same computational resources as the malicious receiver. However, we provide more power to the extractor by giving the extractor non-black-box access to the semi-malicious sender. There is a rich literature on non-black-box techniques in the classical setting starting with the work of [Bar01].

Quantum Extraction via Circular Insecurity of QFHE. The main tool we employ in our protocol is a fully homomorphic encryption QFHE scheme that allows for public homomorphic evaluation of quantum circuits. Typically, we require a fully homomorphic encryption scheme to satisfy semantic security. However, for the current discussion, we require that QFHE to satisfy a stronger security property called 2-circular **in**security:

> Given $\mathsf{QFHE.Enc}_{\mathsf{pk}_1}(SK_2)$ (i.e., encryption of SK_2 under pk_1), $\mathsf{QFHE.Enc}_{PK_2}(\mathsf{sk}_1)$, where $(\mathsf{pk}_1, \mathsf{sk}_1)$ and (PK_2, SK_2) are independently generated public key-secret key pairs, we can efficiently recover sk_1 and SK_2.

Later, we show how to get rid of 2-circular **in**security property by using lockable obfuscation [GKW17, WZ17]. Here is our first attempt to construct the extraction protocol:

- The sender, on input instance x 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 $\mathsf{ct}_3 \leftarrow \mathsf{QFHE.Enc}_{PK_2}(\mathsf{sk}_1)$.
- The receiver sends td'.
- If $\mathsf{td}' = \mathsf{td}$ then the sender sends SK_2.

A quantum extractor with non-black-box access to the private (quantum) state of the semi-malicious sender S does the following:

- It first encrypts the private (quantum) state of S under public key pk_1.
- Here is our main insight: the extractor can homomorphically evaluate the next message function of S on ct_1 and the encrypted state of S. The result is $\mathsf{ct}_1^* = \mathsf{QFHE.Enc}_{\mathsf{pk}_1}(S(\mathsf{td}))$. But note that $S(\mathsf{td})$ is nothing but SK_2; note that S upon receiving $\mathsf{td}' = \mathsf{td}$ outputs SK_2. Thus, we have $\mathsf{ct}_1^* = \mathsf{QFHE.Enc}_{\mathsf{pk}_1}(SK_2)$.
- Now, the extractor has both $\mathsf{ct}_3 = \mathsf{QFHE.Enc}_{PK_2}(\mathsf{sk}_1)$ and $\mathsf{ct}_1^* = \mathsf{QFHE.Enc}_{\mathsf{pk}_1}(SK_2)$. It can then use the circular **in**security of QFHE to recover sk_1, SK_2.
- Finally, it decrypts ct_2 to obtain the witness $\mathbf{w}$!

The correctness of extraction alone is not sufficient; we need to argue that the sender cannot distinguish whether it's interacting with the honest receiver or the extractor. This is not true in our protocol since the extractor will always compute the next message function of S on $\mathsf{td}' = \mathsf{td}$ whereas an honest receiver will send $\mathsf{td}' = \mathsf{td}$ only with negligible probability.

Indistinguishability of Extraction: SFE strikes again. We already encountered a similar issue when we were designing extraction protocols with security against classical receivers and the tool we used to solve that issue was secure function evaluation (SFE); we will use the same tool here as well.

Using SFE, we make another attempt at designing the quantum extraction protocol.

- The sender, on input instance x 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 $\mathsf{ct}_3 \leftarrow \mathsf{QFHE.Enc}_{PK_2}(\mathsf{sk}_1)$.
- 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 $\mathsf{td}' = \mathsf{td}$, else it recovers $\perp$. The sender doesn't receive any output.

The above template guarantees indistinguishability of extraction property[7].

[7]There is a subtle point here that we didn't address: the transcript generated by the extractor is encrypted under QFHE. But after recovering the secret keys, the extractor could decrypt the encrypted transcript.

We next focus on zero-knowledge. To do this, we need to argue that the td' input by the malicious receiver can never be equal to td. One might falsely conclude that the semantic security of QFHE would imply that td is hidden from the sender and hence the argument follows. This is not necessarily true; the malicious receiver might be able to "maul" the ciphertext ct_1 into the messages of the secure function evaluation protocol in such a way that the implicit input committed by the receiver is td'. We need to devise a mechanism to prevent against such mauling attacks.

Preventing Mauling Attacks. We prevent the mauling attacks by forcing the receiver to commit to random strings $(r_1, \ldots, r_\ell)$ in the first round, where $|\mathsf{td}| = \ell$, even before it receives the ciphertexts $(\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3)$ from the sender. Once it receives the ciphertexts, the receiver is supposed to commit to every bit of the trapdoor using the randomness $r_1, \ldots, r_\ell$; that is, the i^{th} bit of td is committed using r_i.

Using this mechanism, we can then provably show that if the receiver was able to successfully maul the QFHE ciphertext then it violates the semantic security of QFHE using a non-uniform adversary.

Replacing Circular Insecurity with Lockable Obfuscation [GKW17, WZ17]. 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[8] is as follows: obfuscate the circuit, with secret key SK_2, ciphertext $\mathsf{QFHE.Enc}_{SK_2}(r)$ hardwired, that takes as input $\mathsf{QFHE.Enc}_{\mathsf{pk}_1}(SK_2)$, decrypts it to obtain SK_2', then decrypts $\mathsf{QFHE.Enc}_{SK_2}(r)$ to obtain r' and outputs sk_1 if $r' = r$. If the adversary does not obtain $\mathsf{QFHE.Enc}_{\mathsf{pk}_1}(SK_2)$ 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 $\mathsf{QFHE.Enc}_{\mathsf{pk}_1}(\bot)$. The idea of using fully homomorphic encryption along with lockable obfuscation to achieve non-black-box extraction was first 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

[8] It shouldn't be too surprising that lockable obfuscation can be used to replace circular insecurity since one of the applications [GKW17, WZ17] of lockable obfuscation was to demonstrate counter-examples for circular security,

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 attempt to rewind but then this would destroy the initial quantum auxiliary state.

3.3.2 Construction of qQEXT

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 (Section 2.4.2).
- Quantum fully homomorphic encryption scheme with some desired properties, (QFHE.Gen, QFHE.Enc, QFHE.Dec, QFHE.Eval) (Section 2.4.3).
  - It admits homomorphic evaluation of arbitrary computations,
  - It admits perfect correctness,
  - The ciphertext of a classical message is also classical.
- Quantum-secure two-party secure computation SFE with the following properties (Section 2.4.5):
  - 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 $\mathbf{LObf} = (\mathsf{LO.Obf}, \mathsf{LO.Eval})$ for $\mathcal{C}$, where every circuit $\mathbf{C}$, parameterized by $(\mathbf{r}, \mathbf{k}, \mathsf{sk}_1, \mathsf{ct}^*)$, in $\mathcal{C}$ is defined in Figure 3-3. Note that $\mathcal{C}$ is a compute-and-compare functionality (Section 2.4.4).

Construction. We construct a protocol (S, R) in Figure 3-5 for a NP language $\mathcal{L}$, and the following lemma shows that (S, R) is a quantum extraction protocol.

Lemma 63. *Assuming the quantum security of* Comm, SFE, QFHE, *and* **LObf**, (S, R) *is a quantum extraction protocol for* $\mathcal{L}$ *secure against quantum adversaries.*

Proof.

Quantum Zero-Knowledge. Let $(x, \mathbf{w}) \in \mathcal{R}$, 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.

C

Input: ct
Hardwired values: $\mathbf{r}$ (lock), $\mathbf{k}, \mathsf{sk}_1, \mathsf{ct}^*$.

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

Figure 3-3: 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 $\left(\mathbf{c} \leftarrow \mathsf{Comm}\left(1^\lambda, (r_1, \ldots, r_\ell); \mathbf{d}\right)\right) \bigwedge \left(\forall i \in [\ell], \mathbf{c}_i^* \leftarrow \mathsf{Comm}\left(1^\lambda, \mathsf{td}_i; r_i\right)\right)$, output sk_2. Here, td_i denotes the i^{th} bit of td.
- Otherwise, output $\perp$.

Figure 3-4: Description of the function f associated with the SFE.

Description of Sim.

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

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

- $\underline{\mathsf{R} \to \mathsf{S}}$: 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.

- $\underline{\mathsf{S} \to \mathsf{R}}$:

  - 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 $\widetilde{\mathbf{C}} \leftarrow \mathsf{LO.Obf}(1^\lambda, \mathbf{C}[\mathbf{r}, \mathbf{k}, \mathsf{sk}_1, \mathsf{ct}^*])$, 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 3-3.
    - $\mathsf{ct}^* \leftarrow \mathsf{QFHE.Enc}_{\mathsf{pk}_2}(\mathbf{r})$

  Send $\mathsf{msg}_1 = \left(\mathsf{ct}_1, \widetilde{\mathbf{C}}, \mathsf{otp} := \mathbf{k} \oplus \mathsf{sk}_1\right)$.

- $\underline{\mathsf{R} \to \mathsf{S}}$: 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 3-4; 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 3-5: Quantum Extraction Protocol (S, R)

- 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:

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

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

Consider the following claim.

Claim 64. $\mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{S}(1^\lambda, x, \mathbf{w}), \mathsf{R}^*(1^\lambda, x, \cdot)\rangle\right) \approx_Q \mathsf{View}_{\mathsf{R}^*}\left(\langle \mathsf{Hybrid}_2.\mathsf{S}(1^\lambda, x, \mathbf{w}), \mathsf{R}^*(1^\lambda, x, \cdot)\rangle\right)$.

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 65. *Assuming the quantum security of* Comm*,* **LObf** *and* QFHE*, the following holds:*

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

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:
>
> - $\mathbf{c}\leftarrow\mathsf{R}^*(1^\lambda,x,\cdot)$,
> - $\mathsf{ct}_1\leftarrow\mathsf{QFHE.Enc}_{\mathsf{pk}_1}(\mathsf{td}||\mathbf{w})$, where $(\mathsf{pk}_i,\mathsf{sk}_i)\leftarrow\mathsf{QFHE.Setup}(1^\lambda),\forall i\in\{1,2\}$ and $\mathsf{td}\xleftarrow{\$}\{0,1\}^\lambda$,
> - $\widetilde{\mathbf{C}}\leftarrow\mathsf{LO.Obf}(1^\lambda,\mathbf{C}[\mathbf{r},\mathbf{k},\mathsf{sk}_1,\mathsf{ct}^*])$, 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,
> - $\mathsf{R}^*(1^\lambda,x,\cdot)$ 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 $\mathsf{p}_1 = \mathsf{Pr}[\mathsf{BAD}_1 = 1]$.

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:
>
> - $\mathbf{c}\leftarrow\mathsf{R}^*(1^\lambda,x,\cdot)$,

- $\mathsf{ct}_1 \leftarrow \mathsf{QFHE.Enc}_{\mathsf{pk}_1}(\mathsf{td}||\mathbf{w})$, where $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{QFHE.Setup}(1^\lambda), \forall i \in \{1,2\}$ and $\mathsf{td} \xleftarrow{\$} \{0,1\}^\lambda$,
- $\widetilde{\mathbf{C}} \leftarrow \mathsf{LO.Obf}(1^\lambda, \mathbf{C}[\mathbf{r}, \mathbf{k}, \mathsf{sk}_1, \mathsf{ct}^*])$, 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,
- $\mathsf{R}^*(1^\lambda, x, \cdot)$ 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 $|\mathsf{p}_1 - \mathsf{p}_{1.1}| \leq \mathrm{negl}(\lambda)$ 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:

- $\mathbf{c} \leftarrow \mathsf{R}^*(1^\lambda, x, \cdot)$,
- $\mathsf{ct}_1 \leftarrow \mathsf{QFHE.Enc}_{\mathsf{pk}_1}(\mathsf{td}||\mathbf{w})$, where $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{QFHE.Setup}(1^\lambda), \forall i \in \{1,2\}$ and $\mathsf{td} \xleftarrow{\$} \{0,1\}^\lambda$,
- $\underline{\widetilde{\mathbf{C}} \leftarrow \mathsf{LObf.Sim}\left(1^\lambda, 1^{|\mathbf{C}|}\right)}$,
- $\mathsf{otp} = \mathbf{k} \oplus \mathsf{sk}_1$ and,
- $\mathsf{R}^*(1^\lambda, x, \cdot)$ 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 $\mathbf{LObf}$, it follows that $|\mathsf{p}_{1.1} - \mathsf{p}_{1.2}| \leq \mathrm{negl}(\lambda)$. Note that we crucially use the fact that the lock $\mathbf{r}$ 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:

- $\mathbf{c} \leftarrow \mathsf{R}^*(1^\lambda, x, \cdot)$,

- $\mathsf{ct}_1 \leftarrow \mathsf{QFHE.Enc}_{\mathsf{pk}_1}(\mathsf{td}||\mathbf{w})$, where $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{QFHE.Setup}(1^\lambda), \forall i \in \{1,2\}$ and $\mathsf{td} \xleftarrow{\$} \{0,1\}^\lambda$,
- $\widetilde{\mathbf{C}} \leftarrow \mathsf{LObf.Sim}\left(1^\lambda, 1^{|\mathbf{C}|}\right)$,
- $\underline{\mathsf{otp} \xleftarrow{\$} \{0,1\}^{|\mathsf{sk}_1|}}$ and,
- $\mathsf{R}^*(1^\lambda, x, \cdot)$ 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:

- $\mathbf{c} \leftarrow \mathsf{R}^*(1^\lambda, x, \cdot)$,
- $\underline{\mathsf{ct}_1 \leftarrow \mathsf{QFHE.Enc}_{\mathsf{pk}_1}(\perp)}$, where $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{QFHE.Setup}(1^\lambda), \forall i \in \{1,2\}$ and $\mathsf{td} \xleftarrow{\$} \{0,1\}^\lambda$,
- $\widetilde{\mathbf{C}} \leftarrow \mathsf{LObf.Sim}\left(1^\lambda, 1^{|\mathbf{C}|}\right)$,
- $\mathsf{otp} \xleftarrow{\$} \{0,1\}^{|\mathsf{sk}_1|}$ and,
- $\mathsf{R}^*(1^\lambda, x, \cdot)$ 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 65 to prove Claim 64. 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 $\mathsf{Hybrid}_2.\mathsf{S}$; this follows from the IND-security of SFE against quantum receivers since $f((\mathsf{td}, \mathbf{c}, \mathbf{c}_1^*, \ldots, \mathbf{c}_\ell^*, \mathsf{sk}_2), (\mathbf{d}, r_1, \ldots, r_\ell)) = f((\perp), (\mathbf{d}, r_1, \ldots, r_\ell))$.

□

$\underline{\mathsf{Hybrid}_3}$: We define a hybrid sender, denoted by $\mathsf{Hybrid}_3.\mathsf{S}$: it behaves exactly like $\mathsf{Hybrid}_2.\mathsf{S}$ 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:

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

$\underline{\mathsf{Hybrid}_4}$: We define a hybrid sender, denoted by $\mathsf{Hybrid}_4.\mathsf{S}$: it behaves exactly like $\mathsf{Hybrid}_3.\mathsf{S}$ 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:

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

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

The following holds unconditionally:

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

$\underline{\mathsf{Hybrid}_6}$: We define a hybrid sender, denoted by $\mathsf{Hybrid}_6.\mathsf{S}$: it behaves exactly like $\mathsf{Hybrid}_5.\mathsf{S}$ except that ct_1 is generated as $\mathsf{ct}_1 \leftarrow \mathsf{QFHE.Enc}_{\mathsf{pk}_1}(\bot)$.

Assuming the quantum security of QFHE, we have:

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

Since $\mathsf{Hybrid}_6.\mathsf{S}$ 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 $\mathsf{S}^*(1^\lambda, x, \mathbf{w}, \cdot)$, and the instance x.

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

- Run R_1 to compute $\mathbf{c}$, $\mathbf{d}$, and $r_1, \ldots, r_\ell$.
- Apply the channel $S_1^*(1^\lambda, x, \mathbf{w}, \mathbf{c}, \cdot)$.
- 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 $\widetilde{\mathsf{C}}$ 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 $\widetilde{\mathsf{C}}\left(\mathsf{QFHE.Enc}_{\mathsf{pk}_1}(\mathsf{SFE.Out})\right)$.

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

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 $\mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*$,

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

The identity is acting on the distinguisher's private state, and the composition

$$\mathcal{E}_{\mathsf{SFE}}(\cdot\,; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*) \circ S_1^*(1^\lambda, x, \mathbf{w}, \mathbf{c}, \cdot)$$

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

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

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 trapdoor, instead, we have

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

Our goal is to show that these two cases, $\mathbf{c}^*$ and $\mathbf{c}_{\mathsf{td}}^*$, 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 $\mathcal{E}_{\mathsf{SFE}}(\cdot\,; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*_{\mathsf{td}})$ 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 trapdoor 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

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

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

$$\left(\left[\mathcal{E}_{\mathsf{SFE}}(\cdot\,; \mathbf{d}, r_1, ..., r_\ell, \mathsf{td}, \mathbf{w}, \mathbf{c}, \mathbf{c}^*_{\mathsf{td}}) \circ S^*_1(1^\lambda, x, \mathbf{w}, \mathbf{c}, \cdot)\right] \otimes \mathsf{Id}\right)(\rho)$$

on the distinguisher's state.

Next, we show that the view of the sender when interacting with the honest receiver is indistinguishable (against QPT algorithms) from the view of the sender when interacting with the extractor.

$\underline{\mathsf{Hybrid}_1}$: The output of this hybrid is the view of the sender when interacting with the honest receiver.

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

The quantum indistinguishability of Hybrid_1 and Hybrid_2 follows from the semantic security of SFE against QPT adversaries.

$\underline{\mathsf{Hybrid}_3}$: We define a hybrid receiver $\mathsf{Hybrid}_3.\mathsf{R}$ that behaves like $\mathsf{Hybrid}_2.\mathsf{R}$ 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 $\mathsf{Hybrid}_3.\mathsf{R}$.

The quantum indistinguishability of Hybrid_2 and Hybrid_3 follows from the quantum computational hiding of Comm.

$\underline{\mathsf{Hybrid}_4}$: We define a hybrid receiver $\mathsf{Hybrid}_4.\mathsf{R}$ that sets $\mathbf{c}^*_i = \mathsf{Comm}(1^\lambda, \mathsf{td}_i; r_i)$, for every $i \in [\ell]$, where td is extracted inefficiently.

To prove that Hybrid_3 and Hybrid_4 are indistinguishable, we first establish some notation. Let p_x be the probability that the sender samples $\mathsf{td} = x$, and let ε_x denote the probability that the sender distinguishes Hybrid_3 and Hybrid_4 when $\mathsf{td} = x$. Let E_x denote the event that sender chooses $\mathsf{td} = x$ and that it distinguishes correctly.

Suppose a QPT distinguisher can distinguish Hybrid_3 and Hybrid_4. Then it follows

that $\Pr[\cup_x E_x]$ is non-negligible. Moreover, we have the following:

$$\begin{aligned}\Pr[\cup_x E_x] &= \sum_x p_x \varepsilon_x \\ &\leq \max_x(\varepsilon_x)\end{aligned}$$

where we used the fact that $\{E_x\}$ are mutually exclusive events. Since $\Pr[\cup_x E_x]$ is non-negligible, this means that there exists an x such that ε_x is non-negligible. This further implies that $\mathsf{Comm}(0)$ and $\mathsf{Comm}(x)$ are distinguishable with non-negligible probability, thus contradicting the quantum computational hiding security of Comm.

Thus, the computational indistinguishability of Hybrid_3 and Hybrid_4 follows from the quantum computational hiding of Comm.

$\underline{\mathsf{Hybrid}_5}$: We define a hybrid receiver $\mathsf{Hybrid}_5.\mathsf{R}$ that behaves as $\mathsf{Hybrid}_4.\mathsf{R}$ 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 Hybrid_4 and Hybrid_5 follows from the quantum computational hiding of Comm.

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

The quantum indistinguishability of Hybrid_5 and Hybrid_6 follows from the semantic security of SFE against polynomial time quantum adversaries.

□

□

Chapter 4

Quantum Zero-Knowledge Protocols

4.1 Introduction

Zero-knowledge [GMR85] is one of the foundational concepts in cryptography. A zero-knowledge system for NP is an interactive protocol between a prover P, who receives as input an instance x and a witness w, and a verifier V who receives as input an instance x. The (classical) zero-knowledge property roughly states that the view of the malicious probabilistic polynomial-time verifier V^* generated after interacting with the prover P can be simulated by a PPT simulator, who doesn't know the witness w. The goal when constructing ZK protocols is to design protocols with additional desirable properties. Two properties that stand out for practical reasons are: (1) low round complexity, and (2) concurrency. In real life, implementing ZK protocols can be quite computationally expensive, so it is important to construct protocols that need the least amount of resources possible. Furthermore, in real life protocols are not executed in isolation. For this reason, concurrent secure protocols have been widely studied in the classical literature [DS98, DCO99, Can01, CLOS02, CF01, RK99, BS05, DNS04, PRS02, Lin03, Pas04, PV08, PTV14, GJO+13, CLP15, FKP19].

Our goal is to study to what extend it is possible to construct low-round complexity or concurrent protocols in the quantum setting. In this chapter we will present two different QZK protocols. The first QZK protocol is an $O(1)$-round classical argument system for NP, while our second protocol is a bounded concurrent QZK proof system. Both of this protocols are classical protocols, i.e. they do not require quantum resources. We then show how to obtain a bounded concurrent QZK proof system for QMA.

Round Complexity in the Quantum Setting. Protocols with neglible soundness and optimal (constant) round complexity have been already achieved in the classical setting [BCPR16, BBK+16, BKP18, BKP19]. In contrast, the first constant round QZK protocol was constructed very recently in concurrent work [BS20]. Their work provides a positive answer to the question we set out to study:

> ***(Q1.1)*** *Are there constant round QZK protocols for* NP *(with negligible soundness)?*

by exhibiting a QZK argument system for NP (and QMA). Their protocol uses non-black-box techniques similar to those from Chapter 3, so it relies on QLWE as well as QFHE. There are still many open questions related to QZK protocols with low round complexity. For example, are there protocols whose security solely relies on QLWE? In this chapter we will present a black-box constant round QZK construction that solely relies on QLWE, although it is a classical argument system[1].

Protocol Composition in the Quantum Setting. Most of the work in quantum zero-knowledge have been done in the standalone setting. These constructions work under the assumption that the designed protocols work in isolation. That is, a standalone protocol is one that only guarantees security if the parties participating in an execution of this protocol do not partake in any other protocol execution. This is an unrealistic assumption.

A natural question to ask is whether there exist *quantum* zero-knowledge protocols (without any setup) that still guarantee security under composition. Barring a few works [Unr10, JKMR06, ABG+20], this direction has largely been unaddressed. The couple of works [JKMR06, ABG+20] that do address composition only focus on parallel composition; in this setting, all the verifiers interacting with the prover should send the i^{th} round messages before the $(i+1)^{th}$ round begins. The setting of parallel composition is quite restrictive; it disallows the adversarial verifiers from arbitrarily interleaving their messages with the prover. A more reasonable scenario, also referred to as *concurrent composition*, would be to allow the adversarial verifiers to choose the scheduling of their messages in any order they desire. So far, there has been no work that addresses concurrent composition in the quantum setting.

In the concurrent setting, quantum zero-knowledge is defined as follows: there is a single prover, who on input instance-witness pair (x, w), can simultaneously interact with multiple verifiers, where all these verifiers are controlled by a single malicious quantum polynomial-time adversary. All the verifiers can potentially share an entangled state. Moreover, they can arbitrarily interleave their messages when they interact with the prover. For example, suppose the prover sends a message to the first verifier, instead of responding, it could let the second verifier send a message, after which the third verifier interacts with the prover and so on.

We say that zero-knowledge in this setting holds if there exists a quantum polynomial-time simulator (with access to the initial quantum state of all the verifiers) that can simultaneously simulate the interaction between the prover and all the verifiers.

Towards answering the following question, in this chapter we present a construction of a bounded concurrent QZK proof system.

(Q1.2) *Are there concurrent QZK proof systems for* NP*?*

[1]Its soundness only holds against classical PPT provers.

4.2 Constant round quantum zero-knowledge classical argument system for NP

4.2.1 Overview

We want to construct a constant round QZK protocol where soundness only holds against malicious PPT receivers. Before formally define this notion in Section 4.2.2, we begin with an overview of the construction. We will show how to turn the construction of cQEXT (Section 3.2) into a QZK protocol.

From Quantum Extraction to Quantum Zero-Knowledge. As a starting point, we consider the quantum analogue of the seminal FLS technique [FLS99] to transform a quantum extraction protocol into a quantum ZK protocol. A first attempt to construct quantum ZK is as follows: let the input to the prover be instance x and witness w while the input to the verifier is x.

- The verifier commits to some trapdoor td. Call the commitment $\mathbf{c}$ and the corresponding decommitment $\mathbf{d}$.
- The prover and verifier then execute a quantum extraction protocol with the verifier playing the role of the sender, on input $(\mathbf{c}, \mathbf{d})$, while the prover plays the role of the receiver on input $\mathbf{c}$.
- The prover and the verifier then run a witness-indistinguishable protocol where the prover convinces the verifier that either x belongs to the language or it knows td.

At first sight, it might seem that the above template should already give us the result we want; unfortunately, the above template is insufficient. The verifier could behave maliciously in the quantum extraction protocol but the quantum extraction protocol only guarantees security against semi-malicious senders. Hence, we need an additional mechanism to protect against malicious receivers. Of course, we require witness-indistinguishability to hold against quantum verifiers and we do know candidates satisfying this assuming quantum hardness of learning with errors [Blu86, LS19].

Handling Malicious Behavior in QEXT. To check that the verifier behaved honestly in the quantum extraction protocol, we ask the verifier to reveal the inputs and random coins used in the quantum extraction protocol. At this point, the prover can check if the verifier behaved honestly or not. Of course, this would then violate soundness: the malicious prover upon receiving the random coins from the verifier can then recover td and then use this to falsely convince the verifier to accept its proof. We overcome this by forcing the prover to commit (we again use the extractable commitment scheme of [PW09]) to some string td' just before the verifier reveals the inputs and random coins used in the quantum extraction protocol. Then we force the prover to use the committed td' in the witness-indistinguishable protocol; the prover

does not gain any advantage upon seeing the coins of the verifier and thus, ensuring soundness.

One aspect we didn't address so far is the aborting issue of the verifier: if the verifier aborts in the quantum extraction protocol, the simulator still needs to produce a transcript indistinguishable from that of the honest prover. Luckily for us, the quantum extraction protocol we constructed before already allows for simulatability of aborting adversaries.

To summarise, our ZK protocol consists of the following steps: (i) first, the prover and the verifier run the quantum extraction protocol, (ii) next the prover commits to a string td' using [PW09], (iii) the verifier then reveals the random coins used in the extraction protocol and, (iv) finally, the prover and the verifier run a quantum WI protocol where the prover convinces the verifier that it either knows a trapdoor td' or that x is a YES instance.

4.2.2 Definition

The following section contains the construction of 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 67 (Classical arguments for NP)**.** *A classical interactive protocol (P, V) is a* **classical argument system** *for an NP language $\mathcal{L}$, associated with an NP relation $\mathcal{R}(\mathcal{L})$, if the following holds:*

- **Completeness***: For any $(x, w) \in \mathcal{R}(\mathcal{L})$, we have that $\Pr[\langle P(1^\lambda, x, w), V(1^\lambda, x)\rangle = 1] \geq 1 - negl(\lambda)$, for some negligible function negl.*
- **Soundness***: For any $x \notin \mathcal{L}$, any PPT classical adversary P^*, and any polynomial-sized auxiliary information aux, we have that $\Pr[\langle P^*(1^\lambda, x, \mathsf{aux}), V(1^\lambda, x)\rangle = 1] \leq negl(\lambda)$, 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, the classical interactive protocol satisfies zero-knowledge against malicious verifiers.

Definition 68 (QZK classical argument system for NP)**.** *A classical interactive protocol (P, V) is a* **quantum zero-knowledge classical argument system** *for a language $\mathcal{L}$, associated with an NP relation $\mathcal{R}(\mathcal{L})$ if both of the following hold.*

- *(P, V) is a classical argument for $\mathcal{L}$ (Definition 67).*
- **Quantum Zero-Knowledge***: For all $(x, w) \in \mathcal{R}(\mathcal{L})$, for any QPT V^* with private register of size $\mathrm{poly}(|x|)$, there exist a QPT Sim such that*

$$\{\mathsf{View}_{V^*}\langle P(x, w), V^*(x, \cdot)\rangle\}_{(x,w)\in\mathcal{R}(\mathcal{L})} \approx_Q \{\mathsf{Sim}(V^*, x, \cdot)\}_{(x,w)\in\mathcal{R}(\mathcal{L})}$$

4.2.3 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 (Section 2.4.2).
- Quantum extraction protocol secure against classical adversaries $\mathsf{cQEXT} = (\mathsf{S}, \mathsf{R})$ (Section 3.2) associated with the relation $\mathcal{R}_{\mathrm{EXT}}$ below. More generally, cQEXT could be any quantum extraction protocol secure against classical adversaries satisfying Claim 62 (indistinguishability of extraction against malicious senders).
$$\mathcal{R}_{\mathrm{EXT}} = \left\{(\mathbf{c},\ (\mathbf{d}, \mathsf{td}))\ :\ \mathbf{c} = \mathsf{Comm}(1^\lambda, \mathsf{td}; \mathbf{d})\right\}$$
- Quantum witness indistinguishable classical argument system $\Pi_{\mathsf{WI}} = (\Pi_{\mathsf{WI}}.P, \Pi_{\mathsf{WI}}.V)$ for the relation $\mathcal{R}_{\mathsf{wi}}$ (Definition 49).

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

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

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

- $(x, w) \in \mathcal{R}$.
- 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 4-1: Relation $\mathcal{R}_{\mathsf{wi}}$ associated with Π_{WI}.

Construction. Let $\mathcal{L}$ be an NP language. We describe a classical interactive protocol (P, V) for $\mathcal{L}$ in Figure 4-2.

Lemma 69. *The classical interactive protocol (P, V) is a quantum zero-knowledge, classical prover, argument system for NP.*

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

- **Trapdoor Committment Phase**: $\underline{V \to P}$: 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 P.

- **Trapdoor Extraction Phase**: P and V run the quantum extraction protocol cQEXT with V taking the role of the sender $\mathsf{cQEXT.S}$ and P taking the role of the receiver $\mathsf{cQEXT.R}$. The input of $\mathsf{cQEXT.S}$ is $(1^\lambda, \mathbf{c}, (\mathbf{d}, \mathsf{td}); \mathbf{r}_{\mathrm{qext}})$ and the input of $\mathsf{cQEXT.R}$ 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 $\mathcal{T}_{V \to P}$.

  Note: The trapdoor extraction phase will be used by the simulator, while proving zero-knowledge, to extract the trapdoor from the malicious verifier.

- $\underline{P \to V}$: Let $k = \lambda$. For every $j \in [k]$, P 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 $sh_0^{(j)}, sh_1^{(j)} \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$.

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

- $\underline{P \to V}$: P sends $(sh_{b^{(j)}}^{(j)}, \mathbf{d}_{b^{(j)}}^{(j)})$ to V.

- $\underline{V \to P}$: V sends $\mathbf{r}_{\mathrm{qext}}, \mathbf{d}, \mathsf{td}$ to P. Then P checks the following:

  - Let $\mathcal{T}_{V \to P}$ 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[2] and t' is the number of rounds of cQEXT. Let the message produced by $\mathsf{cQEXT.S}\left(1^\lambda, \mathbf{c}, (\mathbf{d}, \mathsf{td}); \mathbf{r}_{\mathrm{qext}}\right)$ 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 P aborts If $\mathbf{c} \neq \mathsf{Comm}(1^\lambda, \mathsf{td}; \mathbf{d})$ then abort.

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

- **Decision step**: V computes the decision step of $\Pi_{\mathsf{WI}}.V$.

Figure 4-2: (Classical Prover) Quantum Zero-Knowledge Argument Systems for NP

Soundness. Let P^* be a classical PPT algorithm. We prove that $P^*(1^\lambda, x, \mathsf{aux})$, for $x \notin \mathcal{L}$ and auxiliary information aux, can convince $V(1^\lambda, x)$ with only negligible

probability. Consider the following hybrids.

$\underline{\mathsf{Hybrid}_1}$: The output of this hybrid is the view of the prover $\mathsf{View}_{P^*}(\langle P^*(1^\lambda, x, \mathsf{aux}), V(1^\lambda, x)\rangle)$ along with the decision bit of V.

$\underline{\mathsf{Hybrid}_2}$: We consider the following hybrid verifier $\mathsf{Hybrid}_2.V$ which executes the trapdoor commitment phase and the trapdoor extraction phase with P^* honestly. It then receives $\{((\mathbf{c}_0^{(j)})^*, (\mathbf{c}_1^{(j)})^*)\}_{j\in[k]}$ from the prover. $\mathsf{Hybrid}_2.V$ sends random bits $\{b^{(j)}\}_{j\in[k]}$ to P^* and it then receives $(sh^{(j)}_{b^{(j)}}, \mathbf{d}^{(j)}_{b^{(j)}})$. At this point, $\mathsf{Hybrid}_2.V$ will rewind until it can extract td^* from the commitments; if it extracted multiple values or it didn't extract any value, set $\mathsf{td}^* = \bot$. 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 Hybrid_1.
The following holds:

$$\begin{aligned}
\mathsf{Pr}\left[1 \leftarrow \langle P^*(1^\lambda, x, \mathsf{aux}), \mathsf{Hybrid}_2.V(1^\lambda, x)\rangle\right] &= \mathsf{Pr}\left[\bigwedge_{(\mathsf{td}^*=\mathsf{td} \bigvee \mathsf{td}^*\neq\mathsf{td})}^{1\leftarrow\langle P^*(1^\lambda,x,\mathsf{aux}),\mathsf{Hybrid}_2.V(1^\lambda,x)\rangle} : \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda, \rho_{\mathsf{aux}})\right] \\
&\leq \underbrace{\mathsf{Pr}\left[\bigwedge_{(\mathsf{td}^*=\mathsf{td})}^{1\leftarrow\langle P^*(1^\lambda,x,\mathsf{aux}),\mathsf{Hybrid}_2.V(1^\lambda,x)\rangle} : \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda, \rho_{\mathsf{aux}})\right]}_{\varepsilon_1} \\
&\quad + \underbrace{\mathsf{Pr}\left[\bigwedge_{(\mathsf{td}^*\neq\mathsf{td})}^{1\leftarrow\langle P^*(1^\lambda,x,\mathsf{aux}),\mathsf{Hybrid}_2.V(1^\lambda,x)\rangle} : \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda, \rho_{\mathsf{aux}})\right]}_{\varepsilon_2}
\end{aligned}$$

We prove the following claims.

Claim 70. *$\varepsilon_1 \leq negl(\lambda)$, for some negligible function negl.*

Proof. Consider the following hybrids.

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

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

$$\varepsilon_1 \leq \mathsf{Pr}\left[\bigwedge_{(\mathsf{td}^*=\mathsf{td})}^{1\leftarrow\langle P^*(1^\lambda,x,\mathsf{aux}),\mathsf{Hybrid}_3.V(1^\lambda,x)\rangle} : \mathsf{td}^* \leftarrow \mathsf{Ext}(1^\lambda, \rho_{\mathsf{aux}})\right] + \mathrm{negl}(\lambda) \tag{4.1}$$

$\underline{\mathsf{Hybrid}_4}$: We define the hybrid verifier $\mathsf{Hybrid}_4.V$ that performs the same steps as $\mathsf{Hybrid}_3.V$ 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:

$$\mathsf{Pr}\left[\bigwedge_{(\mathsf{td}^*=\mathsf{td})}^{1\leftarrow\langle P^*(1^\lambda,x,\mathsf{aux}),\mathsf{Hybrid}_3.V(1^\lambda,x)\rangle} \;:\; \mathsf{td}^*\leftarrow\mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right] \tag{4.2}$$

$$\leq \mathsf{Pr}\left[\bigwedge_{(\mathsf{td}^*=\mathsf{td})}^{1\leftarrow\langle P^*(1^\lambda,x,\mathsf{aux}),\mathsf{Hybrid}_4.V(1^\lambda,x)\rangle} \;:\; \mathsf{td}^*\leftarrow\mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right]+\mathrm{negl}(\lambda) \tag{4.3}$$

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

Note that the output distributions of Hybrid_4 and Hybrid_5 are identical. Moreover, the probability that $\mathsf{Hybrid}_5.V$ accepts and $\mathsf{td}^*=\mathsf{td}$ is at most $\frac{1}{2^\lambda}$. Thus we have,

$$\begin{aligned}
&\mathsf{Pr}\left[\bigwedge_{(\mathsf{td}^*=\mathsf{td})}^{1\leftarrow\langle P^*(1^\lambda,x,\mathsf{aux}),\mathsf{Hybrid}_4.V(1^\lambda,x)\rangle} \;:\; \mathsf{td}^*\leftarrow\mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right]\\
=\;&\mathsf{Pr}\left[\bigwedge_{(\mathsf{td}^*=\mathsf{td})}^{1\leftarrow\langle P^*(1^\lambda,x,\mathsf{aux}),\mathsf{Hybrid}_5.V(1^\lambda,x)\rangle} \;:\; \mathsf{td}^*\leftarrow\mathsf{Ext}(1^\lambda,\rho_{\mathsf{aux}})\right]\\
\leq\;&\mathrm{negl}(\lambda)
\end{aligned}$$

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

□

Claim 71. *$\varepsilon_2\leq negl(\lambda)$, for some negligible function negl.*

Proof. Since the trapdoor td^* extracted from P^* 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 unique values (uniqueness follows from perfect binding) committed to in $(\mathbf{c}_0^{(j)})^*$ and $(\mathbf{c}_1^{(j)})^*$ respectively.

From the soundness of Π_{WI}, it then follows that the probability that the verifier accepts is negligible. □

Quantum Zero-Knowledge. Let V^* be the malicious QPT verifier. We describe the simulator Sim as follows.

- It receives $\mathbf{c}$ from V^*.
- Suppose Ext be the extractor of cQEXT associated with $\mathsf{cQEXT.S}^*$, where $\mathsf{cQEXT.S}^*$ is the adversarial sender algorithm computed by V^*. Compute $\mathsf{Ext}(1^\lambda,\mathsf{cQEXT.S}^*,\cdot)$ to obtain td^*. At any time, if V^* aborts, Sim also aborts with the output, the current private state of V^*.
- 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 V^*.
- It receives bits $\{b^{(j)}\}_{j\in[k]}$ from V^*.

- It sends $(sh^{(j)}_{b^{(j)}}, \mathbf{d}^{(j)}_{b^{(j)}})$ from V^*.
- It receives $(\mathbf{r}_{\text{qext}}, \mathbf{d}, \mathsf{td})$ from V^*. It then checks the following:
  - Let $\mathcal{T}_{V\to P}$ be $(m^S_1, m^R_1, \ldots, m^S_{t'}, m^R_{t'})$, where the message m^R_i (resp., m^S_i) is the message sent by the receiver (resp., sender) in the i^{th} round[3] and t' is the number of rounds of cQEXT. Let the message produced by $\mathsf{cQEXT.S}\left(1^\lambda, \mathbf{c}, (\mathbf{d}, \mathsf{td}); \mathbf{r}_{\text{qext}}\right)$ in the i^{th} round be $\widetilde{m}^S_i$.
  - If for any $i \in [t']$, $\widetilde{m}^S_i \neq m^S_i$ then Sim aborts. If $\mathsf{td} \neq \mathsf{td}^*$ then Sim aborts.
- Sim executes Π_{WI} with V^* on input instance $\left(x, \mathsf{td}, \left\{(\mathbf{c}^{(j)}_0)^*, (\mathbf{c}^{(j)}_1)^*\right\}_{j\in[k]}\right)$. The witness Sim uses in Π_{WI} is $\left(\perp, \left\{(sh^{(j)}_0, \mathbf{d}^{(j)}_0, sh^{(j)}_1, \mathbf{d}^{(j)}_1)\right\}_{j\in[k]}\right)$. If V 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.

$\underline{\mathsf{Hybrid}_1}$: The output of this hybrid is the view of V^* when interacting with P. That is, the output of the hybrid is $\mathsf{View}_{V^*}\left(\langle P(1^\lambda, x, w), V^*(1^\lambda, x, \cdot)\rangle\right)$.

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

- Let $\mathcal{T}_{V\to P}$ be $(m^S_1, m^R_1, \ldots, m^S_{t'}, m^R_{t'})$, where the message m^R_i (resp., m^S_i) 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 $\mathsf{cQEXT.S}\left(1^\lambda, \mathbf{c}, (\mathbf{d}, \mathsf{td}); \mathbf{r}_{\text{qext}}\right)$ in the i^{th} round be $\widetilde{m}^S_i$.
- If for any $i \in [t']$, $\widetilde{m}^S_i \neq m^S_i$ then $\mathsf{Hybrid}_2.P$ aborts. If $\mathsf{td} \neq \mathsf{td}^*$ then $\mathsf{Hybrid}_2.P$ aborts.

$\mathsf{Hybrid}_2.P$ finally executes Π_{WI} with V^*; it still uses w in Π_{WI}.

We claim the following holds:

$$\mathsf{View}_{V^*}\left(\langle P(1^\lambda, x, w), V^*(1^\lambda, x, \cdot)\rangle\right) \approx_Q \mathsf{View}_{V^*}\left(\mathsf{Hybrid}_2.P(1^\lambda, x, w), V^*(1^\lambda, x, \cdot)\right)$$

There are two cases:

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

- $\mathsf{cQEXT.S}^*$ does not behave according to the protocol (i.e., not semi-malicious): The view of the verifier when interacting with $\mathsf{Hybrid}_2.P$ is indistinguishable from the view of the verifier when interacting with the honest prover, from the indistinguishability of extraction against malicious senders property (Claim 62).

- $\mathsf{cQEXT.S}^*$ behaves according to the protocol (i.e., it is semi-malicious): In this case, $\mathsf{cQEXT.Ext}$ is able to extract td with probability negligibly close to 1. Moreovoer, as before, the view of the verifier when interacting with the honest prover is indistinguishable from $\mathsf{Hybrid}_2.P$ from Claim 62.

$\underline{\mathsf{Hybrid}_3}$: We define a hybrid prover $\mathsf{Hybrid}_3.P$ as follows: it behaves exactly like $\mathsf{Hybrid}_2.P$ 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]:

$$\mathsf{View}_{V^*}\left(\langle \mathsf{Hybrid}_2.P(1^\lambda, x, w), V^*(1^\lambda, x, \cdot)\rangle\right) \approx_Q \mathsf{View}_{V^*}\left(\mathsf{Hybrid}_3.P(1^\lambda, x, w), V^*(1^\lambda, x, \cdot)\right)$$

$\underline{\mathsf{Hybrid}_4}$: We define a hybrid prover $\mathsf{Hybrid}_4.P$ as follows: it behaves exactly like $\mathsf{Hybrid}_3.P$ 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 $(w, \bot)$. Note that the description of $\mathsf{Hybrid}_4.P$ is identical to the description of Sim.

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

$$\begin{aligned} & \mathsf{View}_{V^*}\left(\langle \mathsf{Hybrid}_3.P(1^\lambda, x, w), V^*(1^\lambda, x, \cdot)\rangle\right) \\ \approx_Q \ & \mathsf{View}_{V^*}\left(\mathsf{Hybrid}_4.P(1^\lambda, x, w), V^*(1^\lambda, x, \cdot)\right) \\ \equiv \ & \mathsf{Sim}(1^\lambda, x, \cdot) \end{aligned}$$

□

On Classical Verifiers

A desirable property from a QZK protocol is if the verifier is classical then so is the simulator. Our protocol as described above doesn't satisfy this property. That is, our simulator is still a QPT algorithm even if the malicious verifier is classical. However, we can do a simple modification to our QZK protocol (Figure 4-2) to satisfy this desired property.

The modification is as follows: in addition to the cQEXT protocol, also sequentially execute a constant round classical extractable commitment scheme satisfying perfectly binding [PW09]. In the classical scheme, the verifer takes the role of the committer committing to $\mathbf{c}$ and $\mathbf{d}$; note that these are the same values it commits to in the cQEXT protocol as well. Note that this wouldn't affect soundness; the classical malicious prover will still be unable to learn $\mathbf{d}$ from the classical extractable commitment scheme, from its hiding property.

To argue zero-knowledge, first consider the following two simulators:

- Sim_c: This simulator runs the extractor in the classical extractable commitment scheme to extract $\mathbf{d}$. It then runs the honest receiver to interact with the verifier in the cQEXT protocol. The rest of the steps is identical to the simulator described in the proof of Lemma 69.
- Sim_q: This simulator runs the honest receiver to interact with the verifier in the classical extractable commitment scheme. It then runs the extractor in the cQEXT protocol to extract $\mathbf{d}$. The rest of the steps is identical to the simulator described in the proof of Lemma 69.

If the malicious verifier is classical PPT then Sim_c can successfully carry out the simulation whereas if the malicious verifier is QPT then Sim_q is successful. While we wouldn't know whether the malicious verifier is classical PPT or not, we know for a fact that one of two simulators will succeed.

4.3 Bounded concurrent quantum zero-knowledge for NP

4.3.1 Overview

Our goal is to construct a bounded concurrent QZK proof system for NP (see Section 4.3.2 for the formal definition). In the bounded setting, the number of verifiers, Q, is known before specifying the protocol.

Black Box QZK via Watrous Rewinding. The traditional rewinding technique that has been used to prove powerful results on classical zero-knowledge cannot be easily ported to the quantum setting. The fundamental reason behind this difficulty is the fact that to carry out rewinding, it is necessary to clone the state of the verifier. While cloning comes for free in the classical setting, the no-cloning theorem of quantum mechanics prevents us from being able to clone arbitrary states. Nonetheless, the seminal work of Watrous [Wat09] demonstrates that there are rewinding techniques that are amenable to the quantum setting. Watrous used this technique to present the first construction of quantum zero-knowledge for NP. This technique is so powerful that all quantum zero-knowledge protocols known so far (including the ones with non-black box simulation [BS20, ABG+20]!) either implicitly or explicitly use this technique.

We can abstractly think of Watrous technique as follows: to prove that a classical protocol is quantum zero-knowledge, first come up with a (classical) PPT simulator that simulates a (classical) malicious PPT verifier. The classical simulator needs to satisfy the following two conditions:

- **Oblivious Rewinding**: There is a distribution induced on the decision bits of the simulator to rewind in any given round i. This distribution could potentially depend on the randomness of the simulator and also the state of the verifier.

The oblivious rewinding condition requires that this distribution should be independent of the state of the verifier. That is, this distribution should remain the same irrespective of the state of the verifier[4].

- **No-recording**: Before rewinding any round, the simulator could record (or remember) the transcript generated so far. This recorded transcript along with the rewound transcript will be used for simulation. For instance, in Goldreich and Kahan [GK96a], the simulator first commits to garbage values and then waits for the verifier to decommit its challenges. The simulator then records the decommitments before rewinding and then changing its own commitments based on the decommitted values.

  The no-recording condition requires the following to hold: in order for the simulator to rewind from point i to point j $(i > j)$, the simulator needs to forget the transcript generated from j^{th} round to the i^{th} round. Note that the simulator of [GK96a] does not satisfy the no-recording condition.

Once such a classical simulator is identified, we can then simulate quantum verifiers as follows: run the classical simulator and the quantum verifier[5] in superposition and then at the end of each round, measure the appropriate register to figure out whether to rewind or not. The fact that the distribution associated with the decision bits are independent of the verifier's state is used to argue that the state, after measuring the decision register, is essentially not disturbed. Using this fact, we can then reverse the computation and go back to an earlier round. Once the computation is reversed (or rewound to an earlier round), the simulator forgets all the messages exchanged from the point – to which its being rewound to – until the current round.

Incompatibility of Existing Concurrent ZK Techniques. To realize our goal of building bounded concurrent QZK, a natural direction to pursue is to look for classical concurrent ZK protocols with the guarantee that the classical simulator satisfies both the oblivious rewinding and no-recording conditions. However, most known classical concurrent ZK techniques are such that they satisfy one of these two conditions but not both. For example, the seminal work of [PRS02] proposes a concurrent ZK protocol and the simulator they describe satisfies the oblivious rewinding condition but not the no-recording condition. More relevant to our work is the work of Pass, Tseng, and Wikström [PTW09], who construct a bounded concurrent ZK protocol whose simulator satisfies the no-recording condition but not the oblivious rewinding condition.

In more detail, at every round, the simulator (as described in [PTW09]) makes a decision to rewind based on which session verifier sends a message in that round. This means that the probability of whether the simulator rewinds any given round depends on the scheduling of the messages of the verifiers. Unfortunately, the scheduling itself

[4] A slightly weaker property where the distribution is "*approximately*" independent of the state of the verifier also suffices.

[5] Without loss of generality, we can consider verifiers whose next message functions are implemented as unitaries and they perform all the measurements in the end.

could be a function of the state of the verifier. The malicious verifier could look at the first bit of its auxiliary state. If it is 0, it will ask the first session verifier to send a message and if it is 1, it will ask the second session verifier to send a message and so on. This means that a simulator's decision to rewind could depend on the state of the verifier.

Bounded Concurrent QZK. We now discuss our construction of bounded concurrent QZK and how we overcome the aforementioned difficulties. Our construction is identical to the bounded concurrent (classical) ZK construction of Pass, Tseng, and Wikström [PTW09], modulo the setting of parameters. We recall their construction below.

The protocol is divided into two phases. In the first phase, a sub-protocol, referred to as *slot*, is executed many times. We will fix the number of executions later when we do the analysis. In the second phase, the prover and the verifier execute a witness-indistinguishable proof system.

In more detail, one execution of a slot is defined as follows:

- Prover sends a commitment of a random bit b to the verifier. This commitment is generated using a statistically binding commitment scheme that guarantees hiding property against quantum polynomial-time adversaries (also referred to as quantum concealing).
- The verifier then sends a uniformly random bit b' to the prover.

We say that a slot is *matched* if $b = b'$.

In the second phase, the prover convinces the verifier that either the instance is in the language or there is a large fraction, denoted by τ, of matched slots. This is done using a proof system satisfying witness-indistinguishability property against efficient quantum verifiers. Of course, τ needs to be carefully set such that the simulator will be able to satisfy this constraint while a malicious prover cannot. Before we discuss the precise parameters, we first outline the simulator's strategy to prove zero-knowledge. As remarked earlier, the classical simulation strategy described in [PTW09] is incompatible with Watrous rewinding. We first discuss a new classical simulation strategy, that we call *block rewinding*, for this protocol and then we discuss how to combine this strategy along with Watrous rewinding to prove quantum zero-knowledge property of the above protocol.

Block Rewinding. Suppose Q be the number of sessions the malicious verifier initiates with the simulator. Since this is a bounded concurrent setting, Q is known even before the protocol is designed. Let ℓ_{prot} be the number of messages in the protocol. Note that the total number of messages exchanged in all the sessions is at most $\ell_{\mathsf{prot}} \cdot Q$. We assume for a moment that the malicious verifier never aborts. Thus, the number of messages exchanged between the prover and the verifier is exactly $\ell_{\mathsf{prot}} \cdot Q$.

The simulator partitions the $\ell_{\mathsf{prot}} \cdot Q$ messages into many blocks with each block being of a fixed size (we discuss the parameters later). The simulator then runs the

verifier till the end of first block. At this point, it checks if this block contains a slot. Note that the verifier can stagger the messages of a particular session across the different blocks such that the first message of a slot is in one block but the second message of this slot could be in a different block. The simulator only considers those slots such that both the messages of these slots are contained inside the first block. Let the set of all the slots in the first block be denoted by $\mu(B_1)$, where B_1 denotes the first block. Now, the simulator picks a random slot from the set $\mu(B_1)$. It then checks if this slot is matched or not. That is, it checks if the bit committed in the slot equals the bit sent by the verifier. If indeed they are equal, it continues to the next block, else it rewinds to the beginning of the first block and then executes the first block again. Before rewinding, it forgets the transcript collected in the first block. It repeats this process until the slot it picked is matched. The simulator then moves on to the second block and repeats the entire process. When the simulator needs to compute a witness-indistinguishable proof for a session, it first checks if the fraction of matched slots for that particular session is at least τ. If so, it uses this information to complete the proof. Otherwise, it aborts.

It is easy to see why the no-recording condition is satisfied: the simulator never stores the messages sent in a block. Let us now analyze why the oblivious rewinding condition is satisfied. Suppose we are guaranteed that in every block there is at least one slot. Then, we claim that the probability that the simulator rewinds is $\frac{1}{2} \pm \text{negl}(\lambda)$, where negl is a negligible function and λ is the security parameter. This is because the simulator rewinds only if the slot is not matched and the probability that a slot is not matched is precisely $\frac{1}{2} \pm \text{negl}(\lambda)$, from the hiding property of the commitment scheme. If we can show that every block contains a slot, then the oblivious rewinding condition would also be satisfied.

Absence of Slots and Aborting Issues: We glossed over a couple of issues in the above description. Firstly, the malicious verifier could abort all the sessions in some block. Moreover, it can also stagger the messages across blocks such that there are blocks that contain no slots. In either of the above two cases, the simulator will not rewind these blocks and this violates the oblivious rewinding condition: the decision to rewind would be based on whether the verifier aborted or whether there were any slots within a block. In turn, these two conditions could depend on the state of the verifier.

To overcome these two issues, we fix the simulator as follows: at the end of every block, it checks if there are any slots inside this block. If there are slots available, then the simulator continues as detailed above. Otherwise, it performs a dummy rewind: it picks a bit uniformly at random and rewinds only if the bit is 0. If the bit is 1, it continues its execution. This ensures that the simulator will rewind with probability $\frac{1}{2} \pm \text{negl}(\lambda)$ irrespective of whether there are any slots inside a block. Thus, with this fix, the oblivious rewinding condition is satisfied as well.

Parameters and Analysis: We now discuss the parameters associated with the system. We set the number of slots in the system to be $120Q^7\lambda$. We set τ to be $\lfloor \frac{60Q^7\lambda + Q^4\lambda}{120Q^7\lambda} \rfloor$. We set the number of blocks to be $24Q^6\lambda$. Thus, the size of each block

is $\lfloor \frac{120Q^7\lambda}{24Q^6\lambda} \rfloor$. Recall that the reason why we need to set these parameters carefully is to ensure that the malicious prover cannot match more than τ slots with better than negligible probability whereas the simulator can beat this threshold with overwhelming probability.

We now argue that the classical simulator can successfully simulate all the Q sessions. To simulate any given session, say the i^{th} session, the number of matched slots needs to be at least $60Q^7\lambda + Q^4\lambda$. Note that the number of blocks is $24Q^6\lambda$; the best case scenario is that each of these blocks contain at least one slot of the i^{th} session and the simulator picks this slot every time. Even in this best case scenario, the simulator can match at most $24Q^6\lambda$ slots and thus, there still would remain $60Q^7\lambda + Q^4\lambda - 24Q^6\lambda$ number of slots to be matched. Moreover, even the likelihood of this best case scenario is quite low.

Instead, we argue the following:

- The simulator only needs to match $3Q^4\lambda$ number of slots for the i^{th} session. We argue that with overwhelming probability, there are $3Q^4\lambda$ blocks such that (i) there is at least one slot from the i^{th} session and, (ii) the simulator happens to choose a slot belonging to this session in each of these blocks.
- Roughly, $\frac{120Q^7\lambda - 3Q^4\lambda}{2} \gg 60Q^7\lambda - 2Q^4\lambda$ number of slots are matched by luck, even without the simulator picking these slots and trying to match. This follows from the fact that with probability $\frac{1}{2}$, a slot is matched and the number of remaining slots that need to be matched are $120Q^7\lambda - 3Q^4\lambda$.

From the above two bullet points, it follows that with overwhelming probability, the total number of slots matched is at least $60Q^7\lambda + Q^4\lambda$.

We note that although the simulation strategy of Pass, Tseng, and Wikström [PTW09] is quite different, their analysis follows the same template as above.

Simulation of Quantum Verifiers: So far we have demonstrated a simulator that can simulate classical verifiers. We describe, at a high level, how to simulate quantum verifiers. The quantum simulator runs the classical simulator in superposition. At the end of every block, it measures a single-qubit register, denoted by **Dec**, which indicates whether the simulator needs to rewind this block or not. If this register has 0, the simulator does not rewind, otherwise it rewinds. We can show that, no matter what the auxiliary state of the malicious verifier is, at the end of a block, the quantum state is of the following form:

$$\sqrt{p}|0\rangle_{\mathbf{Dec}}|\Psi_{\mathsf{Good}}\rangle + \sqrt{1-p}|1\rangle_{\mathbf{Dec}}|\Psi_{\mathsf{Bad}}\rangle,$$

where $|\Psi_{\mathsf{Good}}\rangle$ is a superposition of all the transcripts where the chosen slot is matched and on the other hand, $|\Psi_{\mathsf{Bad}}\rangle$ is a superposition of all the transcripts where the chosen slot is not matched. Moreover, using the hiding property of the commitment scheme, we can argue that $|p - \frac{1}{2}| \leq \mathrm{negl}(\lambda)$. Then we can apply the Watrous rewinding lemma, to obtain a state that is close to $|\Psi_{\mathsf{Good}}\rangle$. This process is repeated for every block. At the end of the protocol, the simulator measures the registers containing the transcript of the protocol and outputs this along with the private state of the verifier.

4.3.2 Definition

In this section we define bounded concurrent QZK protocols for NP. The definition of bounded concurrent QZK for QMA can be found in Section 4.4.2. We start by recalling the definitions of the completeness and soundness properties of a classical interactive proof system.

Definition 72 (Proof System)**.** *Let Π be an interactive protocol between a classical PPT prover P and a classical PPT verifier V. Let $\mathcal{R}(\mathcal{L})$ be the NP relation associated with Π.*

Π is said to satisfy **completeness** *if the following holds:*

- **Completeness**: *For every $(x, w) \in \mathcal{R}(\mathcal{L})$,*

$$\mathsf{Pr}[\mathsf{Accept} \leftarrow \langle P(x,w), V(x)\rangle] \geq 1 - negl(\lambda),$$

for some negligible function negl.

Π is said to satisfy **(unconditional) soundness** *if the following holds:*

- **Soundness**: *For every prover P^* (possibly computationally unbounded), every $x \notin \mathcal{R}(\mathcal{L})$,*

$$\mathsf{Pr}\left[\mathsf{Accept} \leftarrow \langle P^*(x), V(x)\rangle\right] \leq negl(\lambda),$$

for some negligible function negl.

Remark 73. *In Section 5.3, we define a stronger property called proof of knowledge property that subsumes the soundness property.*

To define (bounded) concurrent QZK, we first define Q-session adversarial verifiers. Roughly speaking, a Q-session adversarial verifier is one that invokes Q instantiations of the protocol and in each instantiation, the adversarial verifier interacts with the honest prover. In particular, the adversarial verifier can interleave its messages from different instantiations.

Definition 74 (Q-session Quantum Adversary)**.** *Let $Q \in \mathbb{N}$. Let Π be an interactive protocol between a (classical) PPT prover and a (classical) PPT verifier V for the relation $\mathcal{R}(\mathcal{L})$. Let $(x, w) \in \mathcal{R}(\mathcal{L})$. We say that an adversarial non-uniform QPT verifier V^* is a* **Q-session adversary** *if it invokes Q sessions with the prover $P(x, w)$.*

Moreover, we assume that the interaction of V^* with P is defined as follows: denote by V_i^* to be the verifier algorithm used by V^* in the i^{th} session and denote by P_i to be the i^{th} invocation of $P(x, w)$ interacting with V_i^*. Every message sent by V^* is of the form $\left((1, \mathsf{msg}_1), \ldots, \left(Q, \mathsf{msg}_Q\right)\right)$, where msg_i is defined as:

$$\mathsf{msg}_i = \begin{cases} \mathsf{N/A}, & \text{if } V_i^* \text{ doesn't send a message,} \\ (t, z), & \text{if } V_i^* \text{ sends } z \text{ in the round } t \end{cases}$$

P_i responds to msg_i. If $\mathsf{msg}_i = \mathsf{N/A}$ then it sets $\mathsf{msg}'_i = \mathsf{N/A}$. If V_i^* has sent the messages in the correct order[6], then P_i applies the next message function on

[6]That is, it has sent $(1, z_1)$ first, then $(2, z_2)$ and so on.

its own private state and msg_i *to obtain* z' *and sets* $\mathsf{msg}'_i = (t+1, z')$*. Otherwise, it sets* $\mathsf{msg}'_i = (\bot, \bot)$*. Finally,* V^* *receives* $\big((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q)\big)$*. In total,* V^* *exchanges* $\ell_{\mathsf{prot}} \cdot Q$ *number of messages,* ℓ_{prot} *is the number of the messages in the protocol.*

While the above formulation of the adversary is not typically how concurrent adversaries are defined in the concurrency literature, we note that this formulation is without loss of generality and does capture all concurrent adversaries.

We define quantum ZK for NP in the concurrent setting below.

Definition 75 (Concurrent Quantum ZK for NP)**.** *An interactive protocol* Π *between a (classical) PPT prover* P *and a (classical) PPT verifier* V *for a language* $\mathcal{L} \in \mathrm{NP}$ *is said to be a* **concurrent quantum zero-knowledge (QZK) proof system** *if it satisfies completeness, unconditional soundness and the following property:*

- Concurrent Quantum Zero-Knowledge*: For every sufficiently large* $\lambda \in \mathbb{N}$*, every polynomial* $Q = Q(\lambda)$*, every* Q*-session QPT adversary* V^* *there exists a QPT simulator* Sim *such that for every* $(x, w) \in \mathcal{R}(\mathcal{L})$*,* $\mathrm{poly}(\lambda)$*-qubit bipartite advice state,* ρ_{AB}*, on registers* A *and* B*, the following holds:*

$$\mathsf{View}_{V^*} \langle P(x, w), V^*(x, \rho_{AB}) \rangle \approx_{\mathcal{Q}} \mathsf{Sim}(x, \rho_{AB})$$

  where V^* *and* Sim *only have access to register* A*. In other words, only the identity is performed on register* B*.*

In this work, we consider a weaker setting, called bounded concurrency. The number of sessions, denoted by Q, in which the adversarial verifier interacts with the prover is fixed ahead of time and in particular, the different complexity measures of a protocol can depend on Q.

Definition 76 (Bounded Concurrent Quantum ZK for NP)**.** *Let* $Q \in \mathbb{N}$*. An interactive protocol between a (classical) probabilistic polynomial time (in* Q*) prover* P *and a (classical) probabilistic polynomial time (in* Q*) verifier* V *for a language* $\mathcal{L} \in \mathrm{NP}$ *is said to be a* **bounded concurrent quantum zero-knowledge (QZK) proof system** *if it satisfies completeness, unconditional soundness and the following property:*

- Bounded Concurrent Quantum Zero-Knowledge*: For every sufficiently large* $\lambda \in \mathbb{N}$*, every* Q*-session concurrent QPT adversary* V^**, there exists a QPT simulator* Sim *such that for every* $(x, w) \in \mathcal{R}(\mathcal{L})$*,* $\mathrm{poly}(\lambda)$*-qubit bipartite advice state,* ρ_{AB}*, on registers* A *and* B*, the following holds:*

$$\mathsf{View}_{V^*} \langle P(x, w), V^*(x, \rho_{AB}) \rangle \approx_{\mathcal{Q}} \mathsf{Sim}(x, \rho_{AB})$$

  where V^* *and* Sim *only have access to register* A*. In other words, only the identity is performed on register* B*.*

4.3.3 Construction

We present the construction of quantum zero-knowledge proof system for NP in the bounded concurrent setting in Figure 4-3. As remarked earlier, the construction is the same as the classical bounded concurrent ZK by Pass, Tseng, and Wikström [PTW09], whereas our proof strategy is significantly different from that of [PTW09].

The relation associated with the bounded concurrent system will be denoted by $\mathcal{R}(\mathcal{L})$, with $\mathcal{L}$ being the associated NP language. Let Q be an upper bound on the number of sessions.

Tools.

- Statistically-binding and quantum-concealing commitment protocol (Section 2.4.2), denoted by $(\mathsf{Comm}, \mathsf{R})$.

- Four round quantum witness-indistinguishable proof system Π_{WI} (Definition 49). The relation associated with Π_{WI}, denoted by $\mathcal{R}_{\mathsf{WI}}$, is defined as follows:

$$\mathcal{R}_{\mathsf{WI}} = \Bigg\{ \Big(\big(x, \mathbf{r}_1, \mathbf{c}_1, b'_1, \ldots, \mathbf{r}_{120Q^7\lambda}, \mathbf{c}_{120Q^7\lambda}, b'_{120Q^7\lambda}\big) \ ; \ (w, r_1, \ldots, r_{120Q^7\lambda}) \Big) \ : \ (x,w) \in \mathcal{R}(\mathcal{L}) \bigvee \Bigg(\exists j_1, \ldots, j_{60Q^7\lambda+Q^4\lambda} \in [120Q^7\lambda] \text{ s.t. } \bigwedge_{i=1}^{60Q^7\lambda+Q^4\lambda} \mathsf{Comm}(1^\lambda, \mathbf{r}_{j_i}, b'_{j_i}; r_{j_i}) = \mathbf{c}_{j_i} \Bigg) \Bigg\}$$

Observe that our construction is also a public-coin system. This follows from the fact that the instantiation of the four-round witness-indistinguishable proof system is a public-coin system. We are now ready to prove the following theorem.

Theorem 77. *Assuming the quantum security of* $(\mathsf{Comm}, \mathsf{R})$ *and* Π_{WI}*, the construction in Figure 4-3 is a bounded concurrent QZK proof system.*

Proof. We prove the completeness, soundness and the quantum zero-knowledge properties.

Completeness. This follows from the completeness of Π_{WI}.

Before we prove soundness and quantum zero-knowledge, we first give the following useful definition.

Definition 78 (Matched Slot)**.** *We say that a slot is matched if the bit committed by* P *equals* V*'s response.*

Input of P: Instance $x \in \mathcal{L}$ along with witness w.
Input of V: Instance $x \in \mathcal{L}$.

Stage 1: For $j = 1$ to $120Q^7\lambda$,

- $\underline{P \leftrightarrow V}$: Sample $b_j \xleftarrow{\$} \{0,1\}$ uniformly at random. P commits to b_j using the statistical-binding commitment scheme. Let the verifier's message (verifier plays the role of the receiver) be $\mathbf{r}_j$ and let the prover's message be $\mathbf{c}_j$.

- $\underline{V \to P}$: Sample $b'_j \xleftarrow{\$} \{0,1\}$ uniformly at random. Respond with b'_j.

// We refer to one execution as a slot. So, P and V execute $120Q^7\lambda$ number of slots.

Stage 2: P and V engage in Π_{WI} with the common input being the following:

$$(x, \mathbf{r}_1, \mathbf{c}_1, b'_1, \ldots, \mathbf{r}_{120Q^7\lambda}, \mathbf{c}_{120Q^7\lambda}, b'_{120Q^7\lambda})$$

Additionally, P uses the witness $(w, \perp, \ldots, \perp)$.

Figure 4-3: Construction of bounded concurrent QZK for NP

Soundness. To argue soundness, we need to argue that with probability negligibly close to 1, the number of matched slots in a transcript, associated with an instance not in the language, is less than $60Q^7\lambda + Q^4\lambda$.

Let P^* be the malicious prover and let $x \notin \mathcal{L}$. Denote by $\mathbf{c}_1, \ldots, \mathbf{c}_{120Q^7\lambda}$, the commitments produced by P^* in Stage 1.

We first observe that $(x, \mathbf{r}_1, \mathbf{c}_1, b'_1, \ldots, \mathbf{r}_{120Q^7\lambda}, \mathbf{c}_{120Q^7\lambda}, b'_{120Q^7\lambda}) \notin \mathcal{R}_{\mathsf{WI}}$ with probability negligibly close to 1. By the statistical binding property of the underlying commitment scheme, we have that for every $j \in [60Q^7\lambda + Q^4\lambda]$, there exists a b_j such that $\mathbf{c}_j$ (prover's message in the j^{th} slot) is a commitment of b_j with respect to some randomness. Let X_j be a random variable such that $\mathsf{X}_j = 1$ if $b_j = b'_j$, where b'_j is the

bit sent by V. The following holds (over the randomness of the verifier):

$$\begin{aligned}
& \mathsf{Pr}\left[\exists j_1,\ldots,j_{60Q^7\lambda+Q^4\lambda} \in [120Q^7\lambda] \text{ s.t. } \bigwedge_{i=1}^{60Q^7\lambda+Q^4\lambda}\left(\mathsf{Comm}(1^\lambda,\mathbf{r}_{j_i},b_{j_i};r_{j_i})=\mathbf{c}_{j_i}\bigwedge b_{j_i}=b'_{j_i}\right)\right]\\
= \;& \mathsf{Pr}\left[\sum_{j=1}^{120Q^7\lambda}\mathsf{X}_j \geq 60Q^7\lambda+Q^4\lambda\right]\\
\leq \;& e^{-\frac{(Q^4\lambda)^2}{3(60Q^7\lambda)}} \text{ (By Chernoff Bound)}\\
= \;& e^{-\frac{Q\lambda}{180}}\\
= \;& \mathrm{negl}(\lambda)
\end{aligned}$$

The above observation, combined with the fact that $x \notin \mathcal{L}$, proves the following holds:

$$(x,\mathbf{r}_1,\mathbf{c}_1,b'_1,\ldots,\mathbf{r}_{120Q^7\lambda},\mathbf{c}_{120Q^7\lambda},b'_{120Q^7\lambda}) \notin \mathcal{R}_{\mathsf{WI}}$$

with probability negligibly close to 1.

Quantum Zero-Knowledge

Let the malicious QPT verifier be V^*. We start by describing some notation.

Parameters.

- ℓ_{prot} denotes the number of messages in any given protocol.
- We divide the messages exchanged by the simulator with all the sessions into blocks. Let L denote the number of blocks. We set $L = 24Q^6\lambda$.
- ℓ_{slot} denotes the number of slots in Stage 1 of the protocol. That is, $\ell_{\mathsf{slot}} = 120Q^7\lambda$. Note that every slot contains three messages. We have $\ell_{\mathsf{prot}} = 3\ell_{\mathsf{slot}}+4$.
- ℓ_B denotes the number of messages contained inside one block. Note that $\ell_B = \frac{\ell_{\mathsf{prot}}\cdot Q}{L}$.
- B_i denote the i^{th} block.
- N_i to be number of blocks containing at least one slot of the i^{th} verifier.

Registers used by the simulator: The quantum simulator uses the following registers:

- $\mathbf{R}_t$, for $t \in [\ell_{\mathsf{prot}}\cdot Q]$: it contains the input and randomness used by the simulator to compute the t^{th} message in the transcript; a transcript consists of all the messages in the Q sessions.
- $\mathbf{Sim}_t$, for $t \in [\ell_{\mathsf{prot}}\cdot Q]$: it contains the t^{th} message if it is sent by the simulator.

- $\mathbf{Ver}_t$, for $t \in [\ell_{\mathsf{prot}} \cdot Q]$: it contains the t^{th} message if it is sent by the malicious verifier V^*.
- $\mathbf{M}_i$, for $i \in [L]$: it contains the matched slots of the i^{th} block.
- $\mathbf{B}_i$, for $i \in [Q]$: this is a single-qubit register that contains a bit that indicates whether the simulator needs to use the witness or the matched slots to compute the i^{th} WI proof (where the ordering is determined based on the point of arrival of WI messages).
- $\mathbf{W}$: it contains the NP witness.
- $\mathbf{Aux}$: it contains the private state of the verifier. It is initialized with the auxiliary state of the verifier.
- $\mathbf{Dec}$: it contains the decision register that indicates whether to rewind or not.
- $\mathbf{X}$: this is a poly(λ)-qubit ancillary register.

Description of $\mathsf{Sim}^{V^*}(1^\lambda, x, |\Psi\rangle)$:

1. For any w, let $|\Psi_{0,w}\rangle$ denote the following state:

$$|\Psi_{0,w}\rangle = \left(\bigotimes_{t=1}^{\ell_{\mathsf{prot}} \cdot Q} |0\rangle_{\mathbf{R}_t} |0\rangle_{\mathbf{Sim}_t} |0\rangle_{\mathbf{Ver}_t}\right) \otimes \left(\bigotimes_{j=1}^{L} |0\rangle_{\mathbf{M}_j}\right) \otimes \left(\bigotimes_{i=1}^{Q} |0\rangle_{\mathbf{B}_i}\right) \otimes |w\rangle_{\mathbf{W}} \otimes |\Psi\rangle_{\mathbf{Aux}} \otimes |0\rangle_{\mathbf{Dec}} \otimes |0^{\otimes \mathrm{poly}(\lambda)}\rangle_{\mathbf{X}}$$

   Initialize the state $|\Psi_{0,\perp}\rangle$.

2. For all $j = \{1, 2, \ldots, L\}$, let $U_j^{V^*}$ be the unitary that performs the following operations ((a) and (b)) in superposition.

   (a) For all integers $t \in [(j-1)\ell_B + 1, j\ell_B]$:

   - If the t^{th} message is a Stage 1 message from the prover responding to the first session message of a slot, apply the following operation in superposition over the receiver's message[7]:

$$|\mathbf{r}\rangle_{\mathbf{Ver}_t} |0\rangle_{\mathbf{R}_t} |0\rangle_{\mathbf{Sim}_t} \rightarrow \frac{1}{\sqrt{2^{\lambda+1}}} \sum_{b \in \{0,1\}, r \in \{0,1\}^\lambda} |\mathbf{r}\rangle_{\mathbf{Ver}_t} |b, r\rangle_{\mathbf{R}_t} |\mathsf{Comm}(1^\lambda, \mathbf{r}, b; r)\rangle_{\mathbf{Sim}_t},$$

     while leaving all the other registers intact. Note that we can prepare this state efficiently by first applying $H^{\otimes(\lambda+1)}$ to the $\mathbf{R}_t$ register followed by applying Comm in superposition and storing the output in the $\mathbf{Sim}_t$ register.

[7] We assume without loss of generality that the length of the sender's randomness in the commitment scheme is λ.

- If the t^{th} message is a verifier's message, apply V^* on the registers corresponding to the transcript of the protocol until the t^{th} message (i.e. registers $\{(\mathbf{Sim}_i)\}_{i\leq t}, \{\mathbf{Ver}_i\}_{i<t}$, $\mathbf{Aux}$) and on $\mathbf{Aux}$ register that corresponds to the verifier's private state, and output in the register $\mathbf{Ver}_t$.
- If the t^{th} message is a Stage 2 message from the prover responding to the i^{th} WI initiated by the verifier (this just means that so far, $(i-1)$ WIs from $(i-1)$ sessions have already been initiated in the transcript), let w be the string in the register $\mathbf{W}$. Let c_i be the bit in register $\mathbf{B}_i$. If $c_i = 1$, use w as the witness to the WI proof. If $c_i = 0$, check if at least $\frac{\ell_{\mathsf{slot}}}{2} + Q^4\lambda$ matched slots corresponding to the session whose WI message is being computed. If so, compute the WI of Stage 2 using these matched slots. Otherwise, abort and output $\perp$ on register $\mathbf{Sim}_t$[8].

(b) Let T contain the transcript of messages sent in block B_j along with the input and randomness used by the simulator to create these messages (i.e. the string stored in the registers $\{(\mathbf{R}_t, \mathbf{Sim}_t, \mathbf{Ver}_t)\}_{i\in B_j}$), and let $\mu(T)$ denote the set of all slots that are inside B_j in the transcript T. In superposition, perform the unitary U' defined below. Let I be a register containing a subset of qubits in $\mathbf{X}$. We omit the subscripts of the registers associated with the transcript T.

$$
\begin{aligned}
& U'|T\rangle|0\rangle_{\mathbf{M}_j}|0\rangle_{\mathbf{Dec}}|0^{\otimes|I|}\rangle_I \\
\approx\ & |T\rangle \otimes \left(\frac{1}{\sqrt{|\mu(T)|}} \sum_{(\mathbf{c},b')\in\mu(T)} |\mathbf{c}, b'\rangle_{\mathbf{M}_j} |1 \oplus \mathsf{Match}(T, \mathbf{c}, b'))\rangle_{\mathbf{Dec}} |\phi_{\mathbf{c},b'}\rangle_I \right) \text{ if } \mu(T) \neq \emptyset \\
=\ & |T\rangle|0\rangle_{\mathbf{M}_j}|+\rangle_{\mathsf{Dec}}|0^{\otimes|I|}\rangle_I \text{ if } \mu(T) = \emptyset
\end{aligned}
$$

where $\mathsf{Match}(T, \mathbf{c}, b') = 1$ if $\mathbf{c}$ is a commitment to b' and 0 otherwise. $|\phi_{\mathbf{c},b'}\rangle$ is some auxiliary state. Note that T, in addition to containing the transcript of messages exchanged in B_j, also contains the input and the randomness used by the simulator to create these messages.

By $\approx$, we mean the following: we say $|\phi_0\rangle \approx |\phi_1\rangle$ if both the states $|\phi_0\rangle$ and $|\phi_1\rangle$ are exponentially close (in trace distance) to each other. To see how we can obtain the above state, the unitary U' creates uniform superpositions over $[1], [2], \ldots, [|T|]$. Then, U' determines $\mu(T)$ and uses the uniform superposition over $[|\mu(T)|]$ to create a uniform superposition over $|\mathbf{c}, b'\rangle$.

Let $W_j = \mathsf{Amplifier}\left(U_j^{V^*}\right)$; where $\mathsf{Amplifier}$ is the circuit guaranteed by Lemma 28. Simulator computes $|\Psi_{j,\perp}\rangle = W_j|\Psi_{j-1,\perp}\rangle$.

[8] It may not be clear why we need this register. However, having this register would help us in the presentation of the hybrids.

3. For all $t \in \{1, ..., \ell_{\mathsf{prot}} \cdot Q\}$, measure all the $\mathbf{Sim}_t$ and $\mathbf{Ver}_t$ registers in the computational basis, and output the measurement outcomes along with the resulting state in the **Aux** register. In other words, let Y be the measurement outcome after measuring the registers corresponding to the protocol's transcript. Then, output Y along with

$$\widetilde{\rho} = \frac{\mathrm{Tr}_{\overline{\mathsf{aux}}}\left[\Pi_Y |\Psi_{L,\perp}\rangle\langle\Psi_{L,\perp}| \Pi_Y\right]}{\mathrm{Tr}\left[\Pi_Y |\Psi_{L,\perp}\rangle\langle\Psi_{L,\perp}| \Pi_Y\right]}$$

where Π_Y projects the registers $(\mathbf{Sim}_1, \mathbf{Ver}_1, \ldots, \mathbf{Sim}_{\ell_{\mathsf{prot}} \cdot Q}, \mathbf{Ver}_{\ell_{\mathsf{prot}} \cdot Q})$ onto Y. By $\mathrm{Tr}_{\overline{\mathsf{aux}}}[\cdot]$, we mean the operation of tracing out all the registers except aux.

Remark 79. *Using the description of the unitaries $U_i^{V^*}$ as above, note that for any $(x, w) \in \mathcal{R}(\mathcal{L})$, if the prover and the verifier ran their protocol in superposition (and never measured), their combined output would be $U_L^{V^*} \cdots U_1^{V^*} \left(I \otimes X^{\otimes_{j \in [Q]} \mathbf{B}_j}\right) |\Psi_{0,w}\rangle$, where $X^{\otimes_{j \in [Q]} \mathbf{B}_j}$ is Pauli X's applied to the $\{\mathbf{B}_i\}_{i \in [Q]}$ registers and I is the identity operator applied on the rest of the registers. On the other hand, the state obtained by the simulator just before the final partial measurement is $W_L \cdots W_1 |\Psi_{0,\perp}\rangle$.*

We will show that for any verifier's auxiliary state $|\Psi\rangle$, the output of this simulator is indistinguishable from the output of the verifier when interacting with the honest prover.

Lemma 80. *For any $(x, w) \in \mathcal{R}(\mathcal{L})$, and for any auxiliary $\mathrm{poly}(\lambda)$-qubits state[9] $|\Psi\rangle$, the output of $\mathsf{Sim}^{V^*}(1^\lambda, x, |\Psi\rangle)$ is computationally indistinguishable from $\mathsf{View}_{V^*}\langle P(x, w), V^*(x, |\Psi\rangle)\rangle$.*

Proof. We will proceed with a series of hybrids.

$\underline{\mathsf{Hybrid}_0}$: The output of this hybrid is the output of the verifier when interacting with the honest prover.

$\underline{\mathsf{Hybrid}_1}$: Define a hybrid simulator $\mathsf{Hybrid}_1.\mathsf{Sim}^{V^*}(x, w, |\Psi\rangle)$ that behaves like the honest prover, but performs the execution of the prover and the verifier in all the sessions in superposition. This simulator first prepares the state $U_L^{V^*} \ldots U_1^{V^*} \left(I \otimes X^{\otimes_{j \in [Q]} \mathbf{B}_j}\right) |\Psi_{0,w}\rangle$, then, it measures the registers corresponding to the transcript (that is, $\{(\mathbf{Sim}_t, \mathbf{Ver}_t)\}_{t \in [\ell_{\mathsf{prot}}]]}$) and outputs the measurement outcome along with the resulting verifier's private state.

The distribution of outputs in Hybrid_0 and Hybrid_1 are identical, since measurements can be deferred to the end by the *principle of deferred measurement.*

$\underline{\mathsf{Hybrid}_{2.i}, \text{ for } i = 1 \text{ to } L}$: Consider the following sequence of hybrid simulators, $\mathsf{Hybrid}_{2.i}.\mathsf{Sim}^{V^*}(x, w)$, that behaves like $\mathsf{Hybrid}_1.\mathsf{Sim}^{V^*}(x, w)$, but perform Watrous' rewinding on blocks $B_1, \ldots, B_i$. In other words, instead of performing the unitary $U_i^{V^*}$, it performs

[9] We can assume without of generality, via the process of purification, that the input state of the verifier is a pure state.

$W_i = \mathsf{Amplifier}\left(U_i^{V^*}\right)$. This means that $\mathsf{Hybrid}_{2.i}.\mathsf{Sim}^{V^*}(x, w, |\Psi\rangle)$ computes:

$$U_L^{V^*} \cdots U_{i+1}^{V^*} W_i \cdots W_1 \left(I \otimes X^{\otimes_{j\in[Q]}\mathbf{B}_j}\right)|\Psi_{0,w}\rangle$$

The final partial measurement is performed as in the previous hybrid.

We defer the proof of the following claim.

Claim 81. *Assuming that* Comm *satisfies hiding against quantum polynomial-time adversaries, the output distributions of the verifier in* $\mathsf{Hybrid}_{2.i}$ *is computationally indistinguishable from the output distribution of the verifier in* $\mathsf{Hybrid}_{2.i+1}$.

$\underline{\mathsf{Hybrid}_{3.i} \text{ for } i \in [Q]}$: Define a hybrid simulator $\mathsf{Hybrid}_{3.i}.\mathsf{Sim}^{V^*}$ that behaves like $\mathsf{Hybrid}_{2.L}$ except that it does not applies the initial bit flip X on registers $\mathbf{B}_k$ for all $k \leq i$. Formally, hybrid $\mathsf{Hybrid}_{3.i}$ computes:

$$W_L W_{L-1} \ldots W_1 \left(I \otimes X^{\otimes_{j>i}\mathbf{B}_j}\right)|\Psi_{0,w}\rangle.$$

This change means that in Stage 2 of the protocol, for the sessions that initiate the first i WI protocols, the hybrid simulator $\mathsf{Hybrid}_{3.i}.\mathsf{Sim}$ will use matched slots instead of the actual witness to compute the WI proof. For the rest of the sessions, the hybrid simulator still uses the witness w to produce the WI proof.

We defer the proof of the following claim.

Claim 82. *Assuming the witness-indistinguishability property of* Π_{WI}*, the output distributions of the hybrids* $\mathsf{Hybrid}_{3.i}$ *and* $\mathsf{Hybrid}_{3.i+1}$ *are computationally indistinguishable.*

$\underline{\mathsf{Hybrid}_4}$: The output of this hybrid is the output of the simulator.

The output distributions of $\mathsf{Hybrid}_{3.Q}$ and Hybrid_4 are identical. □

□

Proof of Claim 81

We prove this in the following steps:

1. First, we reduce proving the indistinguishability of $\mathsf{Hybrid}_{2.i}$ and $\mathsf{Hybrid}_{2.i-1}$ to proving the following statement: the following two distributions are computationally indistinguishable.

   - $\mathcal{D}_1$: Measure the $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t\leq i}$ registers at the end of execution of the block B_i in $\mathsf{Hybrid}_{2.i-1}$ and output the measurement outcome along with the residual state in the register $\mathbf{Aux}$.
   - $\mathcal{D}_2$: Measure the $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t\leq i}$ registers at the end of execution of the block B_i in $\mathsf{Hybrid}_{2.i}$ and output the measurement outcome along with the residual state in the register $\mathbf{Aux}$.

2. Next, we show the indistinguishability of $\mathcal{D}_1$ and $\mathcal{D}_2$ by using Watrous rewinding and quantum-concealing property of the commitments.

Bullet 1 follows from the fact that the registers $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t\leq i}$ are never written upon after the execution of Block B_i and hence measurement operators applied on these registers in the end commute with the unitaries applied after the execution of B_i.

For Bullet 2, we first make some observations on the state obtained in $\mathsf{Hybrid}_{2.i}$ after applying Watrous rewinding.

Applying Watrous Rewinding. Let $|\Psi^{i-1}_{0,w}\rangle = W_{i-1}\dots W_1\left(I\otimes X^{\otimes_{j\in[Q]}\mathbf{B}_j}\right)|\Psi_{0,w}\rangle$. Without loss of generality, we can write $U_i^{V^*}|\Psi^{i-1}_{0,w}\rangle$ the following way:

$$U_i^{V^*}|\Psi^{i-1}_{0,w}\rangle \quad = \quad \sqrt{q}|\Phi_{i,\mathsf{noslot}}\rangle|+\rangle_{\mathbf{Dec}} + \sqrt{(1-q)}|\Phi_{i,\mathsf{slot}}\rangle$$

where:

- $|\Phi_{i,\mathsf{noslot}}\rangle$ is a superposition of all the transcripts containing no slot in the i^{th} block B_i. This is defined on all the registers except the **Dec** register.
- $|\Phi_{i,\mathsf{slot}}\rangle$ is a superposition of all the transcripts containing at least one slot in the i^{th} block B_i. This is defined on all the registers.

Furthermore, $|\Phi_{i,\mathsf{slot}}\rangle$ can be written as $\sqrt{p(\Phi_{i,\mathsf{slot}})}|\Phi_{\mathsf{yes}}\rangle|0\rangle_{\mathbf{Dec}}+\sqrt{1-p(\Phi_{i,\mathsf{slot}})}|\Phi_{\mathsf{no}}\rangle|1\rangle_{\mathbf{Dec}}$, for some states $|\Phi_{\mathsf{yes}}\rangle$, $|\Phi_{\mathsf{no}}\rangle$ and some function $p(\cdot)$. We first claim the following.

Claim 83. *Assuming quantum concealing property of* $(\mathsf{Comm},\mathsf{R})$*, the following holds:*

$$\left|p(\Phi_{i,\mathsf{slot}}) - \frac{1}{2}\right| \leq negl(\lambda)$$

Proof. By the quantum-concealing property of Comm, any QPT adversary $\mathcal{A}$, with auxiliary state $|\Phi\rangle$, can win the following game with probability negligibly close to $\frac{1}{2}$: given a commitment $c = \mathsf{Comm}(b;r)$, where $b \xleftarrow{\$} \{0,1\}$ and $r \xleftarrow{\$} \{0,1\}^\lambda$, we say that $\mathcal{A}$ wins if it outputs $b' = b$.

We execute the above experiment in superposition:

- $\mathcal{A}$ sends the first commitment message, $\mathbf{r}$.
- Challenger prepares the following state (omitting the register containing $\mathbf{r}$):

$$\frac{1}{\sqrt{2^{\lambda+1}}}\sum_{b\in\{0,1\},r\in\{0,1\}^\lambda}|b,r\rangle_X|\mathsf{Comm}(1^\lambda,b,\mathbf{r};r)\rangle_Y|0\rangle_Z|\Phi\rangle_{\mathbf{Aux}}|0\rangle_{\mathbf{Dec}}$$

- $\mathcal{A}$ is computed (over the registers $Y, Z, \mathbf{Aux}$) in superposition:

$$\frac{1}{\sqrt{2^{\lambda+1}}} \sum_{b\in\{0,1\},r\in\{0,1\}^\lambda} |b,r\rangle_X |\mathsf{Comm}(1^\lambda,\mathbf{r},b;r)\rangle_Y |\mathcal{A}(\mathsf{Comm}(b;r))\rangle_Z |\Phi'\rangle_{\mathbf{Aux}} |0\rangle_{\mathbf{Dec}}$$

- The challenger computes the following:

$$\frac{1}{\sqrt{2^{\lambda+1}}} \sum_{b\in\{0,1\},r\in\{0,1\}^\lambda} |b,r\rangle_X |\mathsf{Comm}(1^\lambda,\mathbf{r},b;r)\rangle_Y |\mathcal{A}(\mathsf{Comm}(1^\lambda,\mathbf{r},b;r))\rangle_Z |\Phi'\rangle_{\mathbf{Aux}} |b\oplus\mathcal{A}(\mathsf{Comm}(b;r))\rangle_{\mathbf{Dec}}$$

We can rewrite the above state as follows:

$$\sqrt{p'}|\phi_0\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p'}|\phi_1\rangle|1\rangle_{\mathbf{Dec}}$$

From the above game, it follows that p' is negligibly close to $\frac{1}{2}$. Moreover, if we suitably instantiate $\mathcal{A}$ (using the verifier) and $|\Phi\rangle$, it follows that $|\Phi_{\mathsf{yes}}\rangle = |\phi_0\rangle$ and $|\Phi_{\mathsf{no}}\rangle = |\phi_1\rangle$. Thus, we have $p(\Phi_{i,\mathsf{slot}})$ to be negligibly close to $\frac{1}{2}$.

□

Using above, we write $U_i^{V^*}|\Psi_{0,w}^{i-1}\rangle$ as follows:

$$U_i^{V^*}|\Psi_{0,w}^{i-1}\rangle \quad = \quad \sqrt{p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Bad}}\rangle|1\rangle_{\mathbf{Dec}},$$

where $|\Psi_{i,\mathsf{Good}}\rangle$ is a superposition over transcripts such that either one of the following two conditions are satisfied: (i) the slot chosen in the i^{th} block is matched or, (ii) the verifier aborts and the simulator decides to not rewind. Similarly, we can define $|\Psi_{i,\mathsf{Bad}}\rangle$. Define $p_1 = \frac{1}{2}$ and $p_0 = 0.49$. We note that the following holds:

- $|p(\Phi_{i,\mathsf{slot}}) - p_1| \leq \varepsilon$, where $\varepsilon = \nu(\lambda)$, for some negligible function $\nu(\cdot)$ and,
- $p_0(1-p_0) \leq p_1(1-p_1)$ and,
- $p_0 \leq p(\Phi_{i,\mathsf{slot}})$.

Thus, from the Watrous rewinding lemma (Lemma 28), $\mathsf{Amplifier}\left(U_i^{V^*}\right)$ outputs a circuit W_i, of polynomial size, such that W_i on input the state $|\Psi_{0,w}^{i-1}\rangle$, outputs a state $|\Psi_{0,w}^{i}\rangle$ that is exponentially (in λ) close in trace distance to the state $|\Psi_{i,\mathsf{Good}}\rangle$. This means that, in hybrid $\mathsf{Hybrid}_{2.i+1}$, the state obtained after the execution of block B_i is exponentially close in trace distance to the state $|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}}$.

Indistinguishability of $\mathcal{D}_1$ and $\mathcal{D}_2$. We just argued above that the intermediate state obtained in $\mathsf{Hybrid}_{2.i}$ is $|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}}$. On the other hand, the intermediate state obtained in $\mathsf{Hybrid}_{2.i-1}$ is $|\Psi_{0,w}^{i-1}\rangle$ is $U_i^{V^*}|\Psi_{0,w}^{i-1}\rangle = \sqrt{p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Bad}}\rangle|1\rangle_{\mathbf{Dec}}$. We need to argue that the distribution of measurements of the registers $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}$ along with the residual state $\mathbf{Aux}$ register in both the cases are computationally indistinguishable.

Note that for any ρ_0, ρ_1 such that $\rho_0 \approx_c \rho_1$[10], then for any $p \geq 0$ we have that $\rho_0 = p \cdot \rho_0 + (1-p)\rho \approx_c p \cdot \rho_0 + (1-p) \cdot \rho_1$. In our case we have, ρ_0 is the post-measurement state on the registers $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t \leq i}, \mathbf{Aux}$ after measuring the $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t \leq i}$ registers of the state $|\Psi_{\mathsf{Good}}\rangle$. Similarly, we define ρ_1 with respect to $|\Psi_{\mathsf{Bad}}\rangle$. In $\mathsf{Hybrid}_{2.i-1}$, the intermediate state is a mixture of ρ_0 and ρ_1 and in $\mathsf{Hybrid}_{2.i}$, the intermediate state is ρ_0.

Thus, it suffices to show that with probability neglibly close to 1, the post-measurement states ρ_0 and ρ_1 are computationally indistinguishable. This follows from the quantum-concealing property of commitment schemes and is similar to the proof of Claim 83; if the verifier can distinguish a matched slot versus an unmatched slot then this verifier is violating the quantum-concealing property of the commitment scheme.

This proves that hybrids $\mathsf{Hybrid}_{2.i}$ and $\mathsf{Hybrid}_{2.i+1}$ are computationally indistinguishable.

Proof of Claim 82

Before we prove Claim 82, we first give an auxiliary definition and some claims.

Auxiliary Definition and Claims.

Definition 84 (Partitioning)**.** *We define a partitioning of a protocol transcript (consisting of messages from all the sessions) $\mathcal{S}$ to be $\{B_1, \ldots, B_L\}$ associated with parameter ℓ_B as follows: B_1 consists of the first ℓ_B messages of S, B_2 consists of the second ℓ_B messages of S and so on. If $|S| - \ell_B \cdot (L-1) < \ell_B$ then the last block B_L will just contain the remaining $|S| - \ell_B \cdot (L-1)$ messages.*

The following claim lower bounds the number of blocks that will contain a full slot for any given verifier. In particular, with our chosen parameters, we can show that the number of such blocks is at least $6Q^5\lambda$. This will turn out to be enough number of blocks for the simulator to be able to obtain more than $60Q^7\lambda + Q^4\lambda$ matched commitments, with probability neglibly close to 1, for every verifier before starting Stage 2.

Claim 85. *For any transcript $\mathcal{S}$ of Q verifiers $V_1, \ldots, V_Q$ with partitioning $\{B_1, \ldots, B_L\}$, for every verifier V_i, we have $\mathsf{N}_i \geq 6Q^5\lambda$; that is, there are at least $6Q^5\lambda$ number of blocks containing at least one slot of V_i.*

Proof. Fix a verifier V_i. Note that the number of blocks containing at least 4 messages of V_i lower bounds N_i. Denote μ_i be the number of blocks containing at least 4 messages of V_i.

Let $b_1, \ldots, b_{\mu_i}$ be the number of messages of V_i in each of these μ_i blocks. Let the number of messages in the remaining $L - \mu_i$ blocks be denoted by $a_1, \ldots, a_{L-\mu_i}$.

[10]By $\rho_0 \approx_c \rho_1$, we mean that the state sampled according to ρ_0 is computationally indistinguishable from the state sampled according to ρ_1.

The following holds: $\sum_{i=1}^{\mu_i} b_i + \sum_{i=1}^{L-\mu_i} a_i = \frac{2(\ell_{\mathsf{prot}}-1)}{3}$. Since $\sum_{i=1}^{\mu_i} b_i \leq \ell_B \mu_i$, $\sum_{i=1}^{L-\mu_i} a_i \leq 3(L-\mu_i)$ and $\ell_B = \frac{\ell_{\mathsf{prot}} \cdot Q}{L}$, we have:

$$\mu_i \ell_B + 3(L-\mu_i) \geq \frac{2(\ell_{\mathsf{prot}}-1)}{3} \geq \frac{\ell_{\mathsf{prot}}}{2}$$

From this, we can determine μ_i to be at least $\frac{\frac{\ell_{\mathsf{prot}}}{2}-3L}{\ell_B-3}$. We can now lower bound the number of blocks containing at least 4 messages as follows.

$$\begin{aligned}
\mathsf{N}_i \geq \mu_i &\geq \left(\frac{\frac{\ell_{\mathsf{prot}}}{2} - 3L}{\frac{\ell_{\mathsf{prot}} Q}{L} - 3} \right) \\
&\geq \frac{\frac{\ell_{\mathsf{prot}}}{2} - 3L}{\ell_{\mathsf{prot}}} \cdot \frac{L}{Q} \\
&\geq \left[1 - \frac{6L}{\ell_{\mathsf{prot}}} \right] \frac{L}{2Q} \\
&\geq \left[1 - \frac{6L}{3\ell_{\mathsf{slot}}} \right] \frac{L}{2Q} \\
&\geq \left[1 - \frac{2}{5Q} \right] \frac{L}{2Q} \\
&\geq \left(1 - \frac{1}{2} \right) 12\lambda Q^5 \qquad (\because L = 24Q^6\lambda,\ \ell_{\mathsf{slot}} = 120Q^7\lambda) \\
&\geq 6\lambda Q^5
\end{aligned}$$

□

The following claim lower bounds the expected number of slots that will be *rigged* by the simulator (i.e., these are the slots the simulator matches by rewinding) for any given verifier before starting Stage 2. Specifically, it bounds the number of slots that it will be able to match thanks to block rewinding.

Claim 86 (Matching by Rigging)**.** *Let $\mathcal{S}$ be a scheduling of Q verifiers $V_1, \ldots, V_Q$. Let $\{B_1, \ldots, B_L\}$ be the partitioning associated with $\mathcal{S}$.*

Consider the following process: for $i = 1, \ldots, L$,

- *Let T_i be such that all the verifiers $\{V_j\}_{j \in T_i}$ have a slot in B_i.*
- *Pick $j^* \xleftarrow{\$} T$.*
- *Finally, pick a slot of V_{j^*} in block B_i uniformly at random.*

Let $\mathsf{X}_{i,j}$ be a random variable defined to be 1 if in the j^{th} block, a slot of V_i is picked.

Then, for any $i \in [Q]$, $\mathbb{E}[\sum_{j\in[L]} \mathsf{X}_{i,j}] \geq 6\lambda Q^4$. *Furthermore, we have that*

$$\mathsf{Pr}\left[\exists i \in [Q], \sum_{j\in[L]} \mathsf{X}_{i,j} \leq 3\lambda Q^4\right] \leq negl(\lambda)$$

.

Proof. Let $b_{i,j}$ be such that $b_{i,j} = 1$ if the i^{th} verifier has a slot in the j^{th} block, else its set to 0. Then, we have $\mathbb{E}[\sum_{j\in[L]} \mathsf{X}_{i,j}] \geq \sum_{j\in[L]} b_{i,j} \cdot \frac{1}{Q}$. Note that $|\{j : b_{i,j} \neq 0\}| = \mathsf{N}_i$. Thus, we have $\mathbb{E}[\sum_{j\in[L]} \mathsf{X}_{i,j}] \geq \frac{1}{Q} \cdot \mathsf{N}_i$. Further applying Claim 85, we have $\mathbb{E}[\sum_{j\in[L]} \mathsf{X}_{i,j}] \geq 6\lambda Q^4$. To finish the proof of the claim, first notice that by Chernoff bound, we have that for any $i \in [Q]$,

$$\mathsf{Pr}\left[\sum_{j\in[L]} \mathsf{X}_{i,j} \leq 3\lambda Q^4\right] \leq e^{-\frac{3}{4}Q^4\lambda}.$$

By the union bound, we obtain that

$$\mathsf{Pr}\left[\exists i \in [Q], \sum_{j\in[L]} \mathsf{X}_{i,j} \leq 3\lambda Q^4\right] \leq Qe^{-\frac{3}{4}Q^4\lambda}$$

□

While the above claim provides a lower bound on the number of rigged slots, the following claim lower bounds the number of slots matched by luck. Combining the above and the below claim, it follows that with overwhelming probability, the number of matchd slots is at least $60Q^7\lambda + Q^4\lambda$.

Claim 87 (Matching by Luck)**.** *Let* $\mathcal{S}$ *be a transcript of the* Q *verifiers* $V_1, \ldots, V_Q$. *For any* $i \in [Q]$ *let* $Z_{i,1}, \ldots Z_{i,120Q^7\lambda}$ *be binary random variables such that* $Z_{i,j} = 1$ *iff* $\mathsf{Comm}(b'_j; r_j) = \mathbf{c}_j$ *where* b'_j *is the* j^{th} *response of the* i^{th} *verifier to commitment* $\mathbf{c}_j$ *by the prover. Let* $X_{i,j}$ *be as defined in Claim 86. The following holds:*

$$\mathsf{Pr}\left[\exists T_i \subseteq [L],\ (\forall j \in T_i, X_{i,j} = 1) \bigwedge \left(\sum_{j\in[L]\setminus T_i} Z_{i,j} \geq 60Q^7\lambda - 2Q^4\lambda\right)\right] \geq 1 - \nu(\lambda),$$

for some negligible function $\nu(\cdot)$.

Proof. By the previous Claim, we have that with probability neglible close to 1, for all $i \in [Q]$, there exists T_i satisfying the desired properties, what is left is to show that

$$\sum_{j\in[L]\setminus T_i} Z_{i,j} \geq 60Q^7\lambda - 2Q^4\lambda$$

for all $i \in [Q]$.

For any $i \in [Q]$, we have that $\mathbb{E}[\sum_{j\in[L]\setminus T_i} Z_{i,j}] = 60Q^7\lambda - \frac{3}{2}Q^4\lambda$, and by Chernoff bound:

$$\begin{aligned}
\mathsf{Pr}\left[\sum_{j\in[L]\setminus T_i} Z_{i,j} \leq 60Q^7\lambda - 2Q^4\lambda\right] &= \mathsf{Pr}\left[\sum_{j\in[L]\setminus T_i} Z_{i,j} \leq \left(60Q^7\lambda - \frac{3}{2}Q^4\lambda\right) - \frac{1}{2}Q^4\lambda\right] \\
&\leq \exp\left(-\frac{(\frac{1}{2}Q^4\lambda)^2}{2(60Q^7\lambda - \frac{3}{2}Q^4\lambda)}\right) \\
&\leq \exp\left(-\frac{(\frac{1}{2}Q^4\lambda)^2}{2(60Q^7\lambda)}\right) \\
&= e^{-\frac{Q\lambda}{480}}.
\end{aligned}$$

Again, by union bound, we have that

$$\mathsf{Pr}\left[\exists i \in [Q], \sum_{j\in[L]\setminus T_i} Z_{i,j} \leq 60Q^7\lambda - 2Q^4\lambda\right] \leq Qe^{-\frac{Q\lambda}{480}}.$$

□

Combining these last two claims we conclude that the probability that there is a session V_i^* for which the simulator does not have more than $60Q^7\lambda + Q^4\lambda$ matched commitments is negligibly small in λ.

Finishing Proof of Claim 82. We use the auxiliary claims from the previous section to complete the proof.

We prove this via the following hybrid argument.

$\underline{\mathsf{Hybrid}_{3.i}^{(1)}}$: This is identical to the hybrid $\mathsf{Hybrid}_{3.i}$.

$\underline{\mathsf{Hybrid}_{3.i}^{(2)}}$: This is the same as the previous hybrid except that the simulator sets it responses, to the i^{th} session, as $\perp$ if the number of matched slots for the i^{th} session is $< 60Q^7\lambda + Q^4\lambda$.

From Claim 87, we have that the probability that this hybrid aborts is negligible in λ. Conditioned on this hybrid not aborting, the output distributions of $\mathsf{Hybrid}_{3.i}^{(1)}$ and $\mathsf{Hybrid}_{3.i}^{(2)}$ are identical.

$\underline{\mathsf{Hybrid}_{3.i}^{(3)}}$: This is identical to the hybrid $\mathsf{Hybrid}_{3.i+1}$.

To argue that $\mathsf{Hybrid}_{3.i}^{(3)}$ and $\mathsf{Hybrid}_{3.i}^{(2)}$ are computationally indistinguishable we will use the quantum witness indistinguishable property of Π_{WI}. Suppose that there is an adversary $\mathcal{A}$ that distinguishes the output distributions of these two hybrids. We define the following QPT $\mathcal{B}_i$ that breaks the security of Π_{WI}. That is, $\mathcal{B}_i$ is a QPT verifier, in the WI experiment, that can distinguish whether the prover used one witness versus another. $\mathcal{B}_i$ is given as auxiliary advice a transcript (and verifier's

private state) of $\mathsf{Hybrid}_{3.i}^{(2)}$ executed until the verifier's first message of the i^{th} WI execution in the transcript. In particular, conditioned on not aborting, this transcript has enough number of matching slots corresponding to the i^{th} execution (in the order of arrival of messages) of WI. Then, $\mathcal{B}_i$ interacts with the verifier V^* as in the protocol with P from then on, but forwards the verifier's messages corresponding to the i^{th} WI execution to the prover of WI. The output of $\mathcal{B}_i$ is the same as the output of the verifier V^*.

Firstly, from the security of WI, the output distribution of $\mathcal{B}_i$ when the prover uses w is computationally indistinguishable from the output distribution of $\mathcal{B}_i$ when the prover uses the other witness, i.e., decommitments of matched slots.

If the prover used the witness w, then the output distribution of $\mathcal{B}_i$ is computationally indistinguishable from the output of $\mathsf{Hybrid}_{3.i}^{(2)}$. To see why, note that the only difference between $\mathcal{B}_i$ and $\mathsf{Hybrid}_{3.i}^{(2)}$ is that in $\mathcal{B}_i$, all the blocks starting from the i^{th} WI are not rewound. But we already showed, assuming security of commitments, that V^* cannot distinguish the case when the block is being rewound versus the case when it is not.

Furthermore, similarly, when the prover is using the decommitments of matched slots, the output distribution of $\mathcal{B}_i$ is computationally indistinguishable from the output of $\mathsf{Hybrid}_{3.i}^{(3)}$.

Thus, the output distributions of $\mathsf{Hybrid}_{3.i}^{(2)}$ and $\mathsf{Hybrid}_{3.i}^{(3)}$ are computationally indistinguishable.

4.4 Bounded concurrent quantum zero-knowledge proof for QMA

4.4.1 Overview

We show a construction of bounded concurrent QZK for QMA. Our starting point is the QZK protocol for QMA from [BJSW16], which constructs QZK for QMA from QZK for NP, a commitment scheme and a coin-flipping protocol. We first simplify the protocol of [BJSW16] as follows: their protocol requires security of the coin-flipping protocol to hold against malicious adversaries whereas we only require the security to hold against adversaries who don't deviate from the protocol specification. Once we simplify this step, the resulting protocol will satisfy the property that the QZK simulator only rewinds during the execution of the underlying simulator simulating the QZK protocol for NP. This modification makes it easier for us to extend this protocol to the bounded concurrent setting. We simply instantiate the underlying QZK for NP protocol with its bounded concurrent version.

Lets recall the QZK for QMA construction from [BJSW16]. Their protocol is specifically designed for the QMA promise problem called k-local Clifford Hamiltonian, which they showed to be QMA-complete for $k = 5$. We restate it here for completeness.

Definition 88 (k-local Clifford Hamiltonian Problem [BJSW16])**.** *For all $i \in [m]$,*

let $H_i = C_i|0^{\otimes k}\rangle\langle 0^{\otimes k}|C_i^\dagger$ be a Hamiltonian term on k-qubits where C_i is a Clifford circuit.

- *Input: $H_1, H_2, \ldots, H_m$ and strings 1^p, 1^q where p and q are positive integers satisfying $2^p > q$.*
- *Yes instances ($\mathcal{A}_{yes}$): There exists an n-qubit state such that $Tr[\rho \sum_i H_i] \leq 2^{-p}$*
- *No instances ($\mathcal{A}_{no}$): For every n-qubit state ρ, the following holds: $Tr[\rho \sum_i H_i] \geq \frac{1}{q}$*

BJSW Encoding. A key idea behind the construction from [BJSW16] is for the prover to encode its witness, $|\psi\rangle$, using a secret-key quantum authentication code (that also serves as an encryption) that satisfies the following key properties needed in the protocol. For any state $|\psi\rangle$, denote the encoding of $|\psi\rangle$ under the secret-key s by $\mathsf{E}_s(|\psi\rangle)$.

1. *Homomorphic evaluation of Cliffords.* Given $\mathsf{E}_s(|\psi\rangle)$, and given any Clifford circuit C, it is possible to compute $\mathsf{E}_{s'}(C|\psi\rangle)$ efficiently. Moreover, s' can be determined efficiently by knowing C and s.

2. *Homomorphic measurements of arbitrary Clifford basis.* For any Clifford circuit C and any state $|\psi\rangle$, a computational basis measurement on $C|\psi\rangle$ can be recovered from a computational basis measurement on $\mathsf{E}_{s'}(C|\psi\rangle)$ along with C and s. Formally, there is a classically efficiently computable function g such that if y is sampled from the distribution induced by measuring the state $\mathsf{E}_{s'}(C|\psi\rangle)$ in the computational basis, then $g(s, C, y)$ is sampled from the distribution induced by measuring the state $C|\psi\rangle$ in the computational basis.

3. *Authentication of measurement outcomes.* For any s and any clifford C, there is a set $\mathcal{S}_{s,C}$ such that for any state $|\psi\rangle$, and any computational basis measurement outcome y performed on $\mathsf{E}_{s'}(C|\psi\rangle)$, it holds that $y \in \mathcal{S}_{s,C}$. Furthermore, for any y, given s and C, it can be efficiently checked whether $y \in \mathcal{S}_{s,C}$.

4. *Simulatability of authenticated states*: there exists an efficient QPT algorithm B such that for any adversary $\mathcal{A}$, every $x \in \mathcal{A}_{\mathsf{yes}}$ along with witness $|\psi\rangle$, poly(λ)-qubit advice ρ, the following holds: the probability that $\mathcal{P}(s, C_{r^*}^\dagger, \mathcal{A}(\mathsf{E}_s(|\psi\rangle)))$ outputs 1 is negligibly close to the probability that $\mathcal{P}(s, C_{r^*}^\dagger, \mathcal{A}(B(x, s, r^*)))$ outputs 1, where $\mathcal{P}$ is defined below.

$$\mathcal{P}(s, C^\dagger, y) = \begin{cases} 1 \text{ if } g(s, C^\dagger, y) \neq 0 \\ 0 \text{ otherwise} \end{cases}$$

In both the events, s and r^* are chosen uniformly at random.

The QMA verifier of the k-local Clifford Hamiltonian problem measures terms of the form $C|0^{\otimes k}\rangle\langle 0^{\otimes k}|C^\dagger$ where C is a Clifford circuit on a witness $|\psi\rangle$. Specifically, a verifier will first apply $C^\dagger$ and then measure in the computational basis. If the outcome of the measurement is the 0 string, it rejects. Otherwise, it accepts. In the zero-knowledge case, the witness will be encoded, $\mathsf{E}_s(|\psi\rangle)$, but the verifier can still compute $\mathsf{E}_s(C^\dagger|\psi\rangle)$ and measure to obtain some string y. Then, the prover can prove to the verifier (in NP) that y corresponds to a non-zero outcome on a measurement of $C^\dagger|\psi\rangle$ instead using the predicate $\mathcal{P}$.

We follow the approach of BJSW [BJSW16], except that we instantiate the coin-flipping protocol in a specific way in order to get concurrency when instantiating the underlying QZK for NP with our bounded concurrent construction.

4.4.2 Definition

We start by recalling the definitions of completeness and soundness properties of a quantum interactive proof system for promise problems.

Definition 89 (Interactive Quantum Proof System for QMA)**.** *Π is an interactive proof system between a QPT prover P and a QPT verifier V, associated with a promise problem $\mathcal{A} = \mathcal{A}_{yes} \cup \mathcal{A}_{no} \in$ QMA, if the following two conditions are satisfied.*

- **Completeness***: For all $x \in \mathcal{A}_{yes}$, there exists a $\mathrm{poly}(|x|)$-qubit state $|\psi\rangle$ such that the following holds:*

$$\mathsf{Pr}[\mathsf{Accept} \leftarrow \langle P(x, |\Psi\rangle), V(x)\rangle] \geq 1 - negl(|x|),$$

  for some negligible function negl.

Π is said to satisfy **(unconditional) soundness** *if the following holds:*

- **Soundness***: For every prover P^* (possibly computationally unbounded), every $x \in \mathcal{A}_{no}$, the following holds:*

$$\mathsf{Pr}\left[\mathsf{Accept} \leftarrow \langle P^*(x), V(x)\rangle\right] \leq negl(|x|),$$

  for some negligible function negl.

To define bounded concurrent QZK for QMA, we first define the notion of Q-sesssion adversaries.

Definition 90 (Q-session adversary for QMA)**.** *Let $Q \in \mathbb{N}_{\geq 1}$. Let Π be a quantum interactive protocol between a QPT prover and a QPT verifier V for a* QMA *promise problem $\mathcal{A} = \mathcal{A}_{yes} \cup \mathcal{A}_{no}$. We say that an adversarial non-uniform QPT verifier V^* is a* **Q-session adversary** *if it invokes Q sessions with the prover $P(x, |\psi\rangle)$.*

As in the case of concurrent verifiers for NP, we assume that the interaction of V^* with P is defined as follows: denote by V_i^* to be the verifier algorithm used by V^* in the i^{th} session and denote by P_i to be the i^{th} invocation of $P(x, w)$ interacting with

V_i^*. *Every message sent by* V^* *is of the form* $\big((1,\mathsf{msg}_1),\ldots,(Q,\mathsf{msg}_Q)\big)$, *where* msg_i *is defined as:*

$$\mathsf{msg}_i = \begin{cases} \mathsf{N/A}, & \text{if } V_i^* \text{ doesn't send a message,} \\ (t,\rho), & \text{if } V_i^* \text{ sends the state } \rho \text{ in the round } t \end{cases}$$

P_i *responds to* msg_i. *If* $\mathsf{msg}_i = \mathsf{N/A}$ *then it sets* $\mathsf{msg}'_i = \mathsf{N/A}$. *If* V_i^* *has sent the messages in the correct order,* P_i *applies the next message function (modeled as a quantum circuit) on* msg_i *and its private quantum state to obtain* ρ' *and sets* $\mathsf{msg}'_i = (t+1,\rho')$. *Otherwise, it sets* $\mathsf{msg}'_i = (\bot,\bot)$. *Finally,* V^* *receives* $\big((1,\mathsf{msg}'_1),\ldots,(Q,\mathsf{msg}'_Q)\big)$. *In total,* V^* *exchanges* $\ell_{\mathsf{prot}} \cdot Q$ *number of messages, where* ℓ_{prot} *is the number of the messages in the protocol.*

Remark 91. *To invoke* Q *different sessions, we assume that the prover has* Q *copies of the witness state.*

Remark 92. *We assume, without loss of generality, the prover will measure the appropriate registers to figure out the round number for each verifier. This is because the malicious verifier can always send the superposition of the ordering of messages.*

We define quantum ZK for QMA in the bounded concurrent setting below.

Definition 93 (Bounded Concurrent QZK for QMA)**.** *Let* $Q \in \mathbb{N}$. *An interactive protocol* Π *between a QPT prover* P *(running in time polynomial in* Q*) and a QPT verifier* V *(running in time polynomial in* Q*) for a* QMA *promise problem* $\mathcal{A} = \mathcal{A}_{yes} \cup \mathcal{A}_{no}$ *is a* **bounded concurrent QZK proof system** *for* QMA *if it satisfies completeness, unconditional soundness and the following property:*

- **Bounded Concurrent Quantum Zero-Knowledge:** *For every sufficiently large* $\lambda \in \mathbb{N}$, *for every* Q*-session QPT adversary* V^*, *there exists a QPT simulator* Sim *such that for every* $x \in \mathcal{A}_{yes}$ *and any witness* $|\psi\rangle$, $\mathrm{poly}(\lambda)$*-qubit bipartite advice state,* ρ_{AB}, *on registers* A *and* B, *the following holds:*

  $$\mathsf{View}_{V^*}\langle P(x,|\psi\rangle), V^*(x,\rho_{AB})\rangle \approx_{\mathcal{Q}} \mathsf{Sim}(x,\rho_{AB})$$

  where V^* *and* Sim *only have access to register* A. *In other words, only the identity is performed on register* B.

4.4.3 Construction

We use the following ingredients in our construction:

Tools.

- Statistical-binding and quantum-concealing commitment scheme, $(\mathsf{Comm},\mathsf{R})$ (Section 2.4.2).

- Bounded concurrent QZK proof system, denoted by Π_{NP}, for the following language (Section 4.3).

$$\mathcal{L} = \left\{ ((\mathbf{r}, \mathbf{c}, \mathbf{r}', \mathbf{c}', r^*, y, b) \; ; \; (s, \ell, a, \ell')) \; : \; \begin{array}{c} \mathcal{P}(s, C^{\dagger}_{r^*}, y)=1 \\ \bigwedge \\ \mathsf{Comm}(1^\lambda, \mathbf{r}, s; \ell)=\mathbf{c} \\ \bigwedge \\ \mathsf{Comm}(1^\lambda, \mathbf{r}', a; \ell')=\mathbf{c}' \\ \bigwedge \\ a \oplus b = r^* \end{array} \right\}$$

  Let Q be the maximum number of sessions associated with the protocol.

We describe the construction of bounded concurrent QZK for QMA (with bound Q) in Figure 4.4.3. We prove the following.

Theorem 94. *Assuming that Π_{NP} satisfies the definition of bounded concurrent QZK for NP, the protocol given in Figure 4.4.3 is a bounded concurrent QZK protocol for* QMA *with soundness $\frac{1}{\mathrm{poly}}$.*

Remark 95. *The soundness of the above protocol can be amplified by sequential repetition. In this case, the prover needs as many copies of the witness as the number of repetitions.*

Proof Sketch. Completeness follows from [BJSW16].

Soundness. Once we argue that r^* produced in the protocol is uniformly distributed, even when the verifier is interacting with the malicious prover, we can then invoke the soundness of [BJSW16] to prove the soundness of our protocol.

Suppose the verifier accepts the Π_{NP} proof produced during the execution of the above protocol. From the soundness of Π_{NP}, we have that $r^* = a \oplus b$ where a is the string that the prover initially committed to in $\mathbf{c}'$. By the statistical binding security of the commitment, and the fact that b is chosen at random after a has been committed to, we have that r^* is sampled uniformly from $[M]$.

Bounded-Concurrent Quantum Zero-Knowledge. Suppose $x \in A_{\mathsf{yes}}$. Suppose V^* is a non-uniform malicious QPT Q-session verifier. Then we construct a QPT simulator Sim as follows.

Description of Sim: it starts with the registers $\mathbf{X}_{zk}, \mathbf{X}_{anc}, \mathbf{M}, \mathbf{Aux}$. The register $\mathbf{X}_{zk}$ is used by the simulator of the bounded concurrent QZK protocol, $\mathbf{X}_{anc}$ is an ancillary register, $\mathbf{M}$ is used to store the messages exchanged between the simulator and the verifier and finally, the register $\mathbf{Aux}$ is used for storing the private state of the verifier. Initialize the registers $\mathbf{X}_{zk}, \mathbf{M}$ with all zeroes. Initialize the register $\mathbf{X}_{anc}$ with $(\bigotimes_{j=1}^{Q} |s_j\rangle\langle s_j|) \otimes (\bigotimes_{j=1}^{Q} |r_j^*\rangle\langle r_j^*|) \otimes (\bigotimes_{j=1}^{Q} \rho_j) \otimes |0^{\otimes \mathrm{poly}}\rangle\langle 0^{\otimes \mathrm{poly}}|$, where s_i, r_i^* are generated uniformly at random and $\rho_j \leftarrow B(x, s_j, r_j^*)$ is defined in bullet 4 under BJSW encoding.

Sim applies the following unitary for Q times on the above registers. This unitary is defined as follows: it parses the message $((1, \mathsf{msg}_1), \ldots, (Q, \mathsf{msg}_Q))$ in the register

Instance: A k-local Clifford Hamiltonian, $H = \sum_{r=1}^{M} C_r|0^{\otimes k}\rangle\langle 0^{\otimes k}|C_r^\dagger$.

Witness: $|\psi\rangle$

- $\underline{P \leftrightarrow V}$: Prover P samples a secret-key $s \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(k,M)}$, and commits to s using the commitment protocol $(\mathsf{Comm}, \mathsf{R})$. Let $\mathbf{r}$ be the first message of the receiver (sent by V) and $\mathbf{c}$ be the commitment.

  // We call this commitment, the secret-key commitment.

- $\underline{P \to V}$: P sends $\mathsf{E}_s(|\psi\rangle)$.

- $P \leftrightarrow V$: Prover samples a random string $a \xleftarrow{\$} \{0,1\}^{\log(M)}$, and commits to a using the commitment protocol $(\mathsf{Comm}, \mathsf{R})$. Let $\mathbf{r}'$ be the first message of the receiver and $\mathbf{c}'$ be the commitment.

  // We call this commitment, the coin-flipping commitment.

- $\underline{V \to P}$: Verifier samples a random string $b \xleftarrow{\$} \{0,1\}^{\log(M)}$. Verifier sends b to the prover.

- $\underline{P \to V}$: Prover sends $r^* := a \oplus b$ to the verifier.

- $\underline{V \to P}$ Verifier computes $\mathsf{Eval}\left(C_{r^*}^\dagger, \mathsf{E}_s(|\Psi\rangle)\right) \to \mathsf{E}_s(C_{r^*}^\dagger|\psi\rangle)$ and measures in the computational basis. Let y denote the measurement outcome. Verifier sends y to the prover.

- Prover checks that $y \in \mathcal{S}_{s,C_{r^*}^\dagger}$ and that $\mathcal{P}(s, C_{r^*}^\dagger, y) = 1$. If not, it aborts.

- $\underline{P \leftrightarrow V}$: Prover and verifier engage in a QZK protocol for NP, Π_{NP}, for the statement $(\mathbf{r}, \mathbf{c}, \mathbf{r}', \mathbf{c}', r^*, y, b)$ and the witness (s, ℓ, a, ℓ').

Figure 4-4: Bounded-Concurrent QZK for QMA

M. For every round of conversation, it does the following: if it is V^*'s turn to talk, it applies V^* on **Aux** and **M**. Otherwise,

- Let S_1 be the set of indices such that for every $i \in S_1$, msg_i is a message in the protocol Π_{NP}. Finally, let $S_2 = [Q] \backslash S_1$.
- It copies $((1, \mathsf{msg}_1), \ldots, (Q, \mathsf{msg}_Q))$ into $\mathbf{X}_{zk}$ (using many CNOT operations) and for every $i \notin S_1$, replaces msg_i with N/A. We note that msg_i is a quantum state (for instance, it could be a superposition over different messages).

- For every $i \in S_2$, if msg_i is the first prover's message of the i^{th} session, then set msg'_i to be $|\mathbf{c}_i\rangle\langle\mathbf{c}_i| \otimes \rho_i$, where $\mathbf{c}_i$ is the secret-key commitment of 0. If msg_i corresponds to the coin-flipping commitment, then set msg'_i to be $|\mathbf{c}'_i\rangle\langle\mathbf{c}'_i|$ where $\mathbf{c}'_i$ is a commitment to 0.

- It applies the simulator of Π_{NP} on $\mathbf{X}_{zk}$ to obtain $((1, \mathsf{msg}'_{1,zk}), \ldots (Q, \mathsf{msg}'_{Q,zk}))$. The i^{th} session simulator of Π_{NP} takes as input $(\mathbf{r}_i, \mathbf{c}_i, \mathbf{r}'_i, \mathbf{c}'_i, r^*_i, y_i, b_i)$, where r^*_i was generated in the beginning and $\mathbf{r}_i, \mathbf{c}_i, \mathbf{r}'_i, \mathbf{c}'_i, y_i, b_i$ are generated as specified in the protocol.

- Determine $((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q))$ as follows. Set $\mathsf{msg}'_i = \mathsf{msg}_{i,\mathsf{zk}}$, if $i \in S_1$. Output of this round is $((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q))$.

We claim that the output distribution of Sim (ideal world) is computationally indistinguishable from the output distribution of V^* when interacting with the prover (real world).

$\underline{\mathsf{Hybrid}_1}$: This corresponds to the real world.

$\underline{\mathsf{Hybrid}_2}$: This is the same as Hybrid_1 except that the verifier V^* is run in superposition and the transcript is measured at the end.

The output distributions of Hybrid_1 and Hybrid_2 are identical.

$\underline{\mathsf{Hybrid}_3}$: Simulate the zero-knowledge protocol Π_{NP} simultaneously for all the sessions. Other than this, the rest of the hybrid is the same as before.

The output distributions of Hybrid_2 and Hybrid_3 are computationally indistinguishable from the bounded concurrent QZK property of Π_{NP}.

$\underline{\mathsf{Hybrid}_{4.i} \text{ for } i \in [Q]}$: For every $j \leq i$, the coin-flipping commitment in the j^{th} session is a commitment to 0 instead of a_i. For all $j > i$, the commitment is computed as in the previous hybrid.

The output distributions of $\mathsf{Hybrid}_{4.i-1}$ (or Hybrid_3 if $i = 1$) and $\mathsf{Hybrid}_{4.i}$ are computationally indistinguishable from the quantum concealing property of $(\mathsf{Comm}, \mathsf{R})$.

$\underline{\mathsf{Hybrid}_{5.i} \text{ for } i \in [Q]}$: For every $j \leq i$, the secret-key commitment in the j^{th} session is a commitment to 0. For all $j > i$, the commitment is computed as in the previous hybrid.

The output distributions of $\mathsf{Hybrid}_{5.i-1}$ (or $\mathsf{Hybrid}_{4.Q}$ if $i = 1$) and $\mathsf{Hybrid}_{5.i}$ are computationally indistinguishable from the quantum concealing property of $(\mathsf{Comm}, \mathsf{R})$.

$\underline{\mathsf{Hybrid}_{6.i} \text{ for } i \in [Q]}$: For every $j \leq i$, the encoding of the state is computed instead using $B(x, s_i, r^*_i)$, where s_i, r^*_i is generated uniformly at random.

The output distributions of $\mathsf{Hybrid}_{6.i-1}$ and $\mathsf{Hybrid}_{6.i}$ are statistically indistinguishable from simulatability of authenticated states property of BJSW encoding (bullet 4). This follows from the following fact: conditioned on the prover not aborting, the

output distributions of the two worlds are identical. Moreover, the property of simulatability of authenticated states shows that the probability of the prover aborting in the previous hybrid is negligibly close to the probability of the prover aborting in this hybrid.

$\underline{\mathsf{Hybrid}_7}$: This corresponds to the ideal world.

The output distributions of $\mathsf{Hybrid}_{6.Q}$ and Hybrid_7 are identical.

□

Chapter 5

Quantum Proofs of Knowledge

Proof of knowledge (PoK) is a strengthening of the soundness condition in interactive protocols. In a proof of knowledge, the goal is for the prover to convince the verifier that it knows something. If the verifier accepts in a PoK protocol for a language $\mathcal{L} \in$ NP, then it is convinced that the prover knows a witness. Unruh [Unr12] introduced quantum proofs of knowledge (QPoK) where the verifiers are allowed to be quantum. He also constructed a QPoK protocol for NP, but his protocol does not compose well as it does not quite satisfy the same properties found in classical PoK constructions (simulatability and extractability). We want to construct QPoK with these properties; furthermore, we would like to have concurrent QZK protocols with the QPoK property. In this chapter, we show how to achieve both of these goals. The key ingredient in our construction is statistical receiver-private OT which were recently studied in [GJJM20, DGH+20]. For our purposes, we need a post-quantum secure statistical receiver-private OT, which we also construct in this chapter.

5.1 Overview

We start with an overview of our constructions.

Standalone Quantum Proofs of Knowledge

Towards building a bounded-concurrent QZK system satisfying quantum proof of knowledge property, we first focus on the standalone QZK setting. The quantum proof of knowledge property roughly says the following: for every unbounded prover convincing a verifier to accept an instance x with probability p, there exists an extractor that outputs a witness w with probability negligibly close to p and it also outputs a state $|\Phi\rangle$ that is close (in trace distance) to the output state of the real prover.

Our approach is to design a novel extraction mechanism that uses oblivious transfer to extract a bit from a quantum adversary.

Main Tool: Statistical Receiver-Private Oblivious Transfer. Our starting point is an oblivious transfer (OT) protocol [Rab05]. This protocol is defined between

two entities: a sender and a receiver. The sender has two bits (m_0, m_1) and the receiver has a single bit b. At the end of the protocol, the receiver receives the bit m_b.

The security against malicious senders (receiver privacy) states that the sender should not be able to distinguish (with non-negligible probability) whether the receiver's bit is 0 or 1. The security against malicious receivers (also called sender privacy) states that there is a bit b' such that the receiver cannot distinguish (with non-negligible probability) the case when the sender's input is (m_0, m_1) versus the setting when the sender's input is $(m_{b'}, m_{b'})$.

We require receiver privacy to hold against unbounded senders while we require sender privacy to hold against quantum polynomial-time receivers. The reason we require receiver privacy against unbounded senders is because our goal is to design extraction mechanism against computationally unbounded provers.

We postpone discussing the construction of statistical receiver-private oblivious transfer. We will now see how to use this to achieve extraction.

One-bit Extraction with $\left(\frac{1}{2} \pm \mathbf{negl}\right)$-error. We begin with a naive attempt to design the extraction mechanism for extracting a single secret bit, say s[1]. The prover and the verifier execute the OT protocol; prover takes on the role of the OT sender and the verifier takes on the receiver's role. The prover picks bits b and α uniformly at random and then sets the OT sender's input to be (s, α) if $b = 0$, otherwise if $b = 1$, it sets the OT sender's input to be (α, s). The verifier sets the receiver's bit to be 0. After the OT protocol ends, the prover sends the bit b. Note that if the bit b picked by the prover was 0 then the verifier can successfully recover s, else it recovers α.

We first discuss the classical extraction process. The quantum extractor runs the classical extractor in superposition as we did in the case of quantum zero-knowledge. The extraction process proceeds as follows: the extractor picks a bit $\widetilde{b}$ uniformly at random and sets $\widetilde{b}$ to be the receiver's bit in the OT protocol. By the statistical receiver privacy property of OT, it follows that the probability that the extractor succeeds in recovering s is negligibly close to $\frac{1}{2}$. Moreover, the success probability is independent of the initial state of the prover. This means that we can apply the Watrous rewinding lemma and amplify the success probability.

Malicious Provers: However, we missed a subtle issue: the malicious prover could misbehave. For instance, the prover can set the OT sender's input to be (r, r) and thus, not use the secret bit s at all.

We resolve this issue by additionally requiring the prover to prove to the verifier that one of its inputs in the OT protocol is the secret bit[2] s. This is realized by using a quantum zero-knowledge protocol, denoted by Π.

[1]For instance, s could be the firt bit of the witness.

[2]For now, assume that there exists a predicate that can check if s is a valid secret bit.

Error amplification. A malicious verifier can successfully recover the secret s with probability $\frac{1}{2}$. To reduce the verifier's success probability, we execute the above process (i.e., first executing the OT protocol and then executing the ZK protocol) λ number of times, where λ is the security parameter. First, the prover will additively secret share the bit s into secret shares $sh_1, \ldots, sh_\lambda$. It also samples the bits $b_1, \ldots, b_\lambda$ uniformly at random. In the i^{th} execution, it sets the OT sender's input to be (sh_i, α_i) if $b_i = 0$, otherwise it sets the OT sender's input to be (α_i, sh_i), where α_i is sampled uniformly at random. After all the OT protocols are executed, the prover is going to prove using a QZK protocol Π, as considered above, that the messages in the OT protocols were correctly computed.

We first argue that even in this protocol, the extraction still succeeds with overwhelming probability. In each OT execution, the extractor applies Watrous rewinding, as before, to extract all the shares $sh_1, \ldots, sh_\lambda$. From this, it can recover s. All is left is to argue that this template satisfies quantum zero-knowledge property. It turns out that arguing this is challenging[3].

Challenges in Proving QZK and Distinguisher-Dependent Hybrids. We first define the simulator as follows:

- The simulator uses (α_i, α_i) as the sender's input in the i^{th} OT execution, where α_i is sampled uniformly at random.
- It then simulates the protocol Π.

To prove that the output distribution of the simulated world is computationally indistinguishable from the real world, we adopt a hybrid argument. The first hybrid, Hybrid_1, corresponds to the real world. In the second hybrid, Hybrid_2, simulate the protocol Π. The indistinguishability of Hybrid_1 and Hybrid_2 follows from the QZK property of Π. Next, we define the third hybrid, Hybrid_3, that executes the simulator. To prove the indistinguishability of Hybrid_2 and Hybrid_3, we consider a sequence of intermediate hybrids, denoted by $\{\mathsf{Hybrid}_{2.j}\}_{j\in[\lambda]}$. Using this sequence of hybrids, we change the inputs in all the λ OT executions one at a time. Finally, we define the third hybrid, Hybrid_3, that corresponds to the ideal world. Proving the indistinguishability of the consecutive hybrids, $\mathsf{Hybrid}_{2.j}$ and $\mathsf{Hybrid}_{2.j+1}$, in this sequence turns out to be challenging.

The main issue is the following: suppose we are in the j^{th} intermediate hybrid $\mathsf{Hybrid}_{2.j}$, for $j \leq \lambda$. At this point, we have changed the inputs to the first j OT executions and we are about to change the input to the $(j+1)^{th}$ OT. But what exactly are the inputs we are using for the first j OT executions? It is unclear whether we

[3]We would like to point out that we are designing the standalone PoK protocol as a stepping stone towards the bounded concurrent PoK protocol. If one were to be interested in just the standalone setting, then it might be possible to avoid the subtleties described above by making use of a simulation-secure OT rather than an indistinguishable-secure OT. The reason why we use an indistinguishable-secure OT in the concurrent PoK setting instead of a simulation-secure OT is because we want to avoid using more than one simulator in the analysis; otherwise, we would have multiple simulators trying to rewind the verifier, making the analysis significantly complicated.

use the input (sh_i, sh_i) or the input (α_i, α_i), for $i \leq j$, in the i^{th} OT execution. Note that the OT security states that we can either switch the real sender's inputs to either (sh_i, sh_i) or (α_i, α_i), based on the sender's and the distinguisher's randomness. And hence, we define an *inefficient* intermediate hybrid, which is a function (not necessarily computable), that determines for every i, where $i \leq j$, whether to use (sh_i, sh_i) or (α_i, α_i). Moreover, *this hybrid depends on the distinguisher*, that distinguishes the two intermediate hybrids.

The indistinguishability of the consecutive pair of inefficient hybrids, say $\mathsf{Hybrid}_{2.j}$ and $\mathsf{Hybrid}_{2.j+1}$, is proven by a non-uniform reduction that receives as input the advice corresponding to the first j executions of OT, where the sender's inputs are correctly switched to either (sh_i, sh_i) or (α_i, α_i), for $i \leq j$. This in turn depends on the distinguisher distinguishing these two hybrids. Then, the reduction uses the $(j+1)^{th}$ OT execution in the protocol to break the sender privacy property of OT. If the two hybrids can be distinguished with non-negligible probability then the reduction can succeed with the same probability.

In the hybrid $\mathsf{Hybrid}_{2.\lambda-1}$, we additionally include an abort condition: if the inputs in the first $\lambda - 1$ OT executions are all switched to (sh_i, sh_i) then we abort. We show that the probability that $\mathsf{Hybrid}_{2.\lambda-1}$ aborts is negligible. This is necessary to argue that the verifier does not receive all the shares of the secret.

Note that only the intermediate hybrids, namely $\{\mathsf{Hybrid}_{2.j}\}_{j\in[\lambda]}$, are inefficient, and in particular, the final hybrid Hybrid_3 is still efficient.

Extraction of Multiple Bits. To design a quantum proof of knowledge protocol, we need to be able to extract not just one bit, but multiple bits. To achieve this, we design the prover as follows: on input a witness w, it sequentially executes the above extraction template for each bit of the witness. That is, for every $i \in [\ell_w]$, where ℓ_w is the length of w, it additively secret shares w_i into the shares $(sh_{i,1}, \ldots, sh_{i,\lambda})$. It then invokes $\ell_w \cdot \lambda$ number of OT executions, where in the $(i,j)^{th}$ execution, it chooses the input $(sh_{i,j}, \alpha_{i,j})$ if $b_{i,j} = 0$, or the input $(\alpha_{i,j}, sh_{i,j})$ if $b_{i,j} = 1$, where $\alpha_{i,j}, b_{i,j}$ are sampled uniformly at random. Finally, it uses a QZK protocol to prove that it behaved honestly in the earlier OT executions.

The proofs of quantum proof of knowledge and the QZK properties follow along the same lines as the single-bit extraction case.

Statistical Receiver-Private OT with Post-Quantum Security

All that is left is to construct an oblivious transfer protocol that guarantees statistical indistinguishability property against malicious senders and indistinguishability property against QPT malicious receivers. We denote the protocol that we intend to construct to be Π_{SROT}.

The starting point of our construction is another oblivious transfer protocol, denoted by Π_{SSOT}, that has its properties flipped. That is, Π_{SSOT} satisfies statistical indistinguishability property against malicious *receivers* and indistinguishability property against QPT malicious *senders*. The reason we start with this protocol is that we

do know how to achieve this; Brakerski-Döttling [BD18] constructed such a protocol from QLWE.

Our approach is inspired from previous works [KKS18, GJJM20, DGH+20] that show how to construct statistical receiver-private OT from statistical sender-private OT.

Our first attempt to construct Π_{SROT} is the following:

- The sender of Π_{SSOT} samples a random bit $r \xleftarrow{\$} \{0,1\}$. It takes the role of the receiver in the underlying Π_{SROT}. It then sends the first message of Π_{SROT} with the receiver's message set to be r.

- The receiver of Π_{SSOT}, on input choice bit β, samples another random bit r'. It takes the role of the sender in the underlying protocol Π_{SSOT}. It then sends the sender's message in Π_{SSOT}, where the sender's input in Π_{SSOT} is set to be $(r', r' \oplus \beta)$.

- After the end of the execution of Π_{SSOT}, the sender on input (m_0, m_1), does the following: it recovers the message $\widetilde{r}$ from the underlying OT. It then sends $(\widetilde{r} \oplus m_0, \widetilde{r} \oplus r \oplus m_1)$ to the receiver.

If $\beta = 0$ then $\widetilde{r} = r'$ and so, the receiver can recover m_0. If $\beta = 1$ then $\widetilde{r} = r' \oplus r$ and so, the receiver can recover m_1.

The receiver privacy against computationally unbounded senders follows from the statistical sender privacy of the underlying two-round oblivious transfer protocol.

To prove sender privacy against QPT receivers, first let us make the previously described security notion more precise. The malicious receiver R^*, on input state $|\Psi\rangle$, interacts with the sender and produces an auxiliary state $|\widetilde{\Psi}\rangle$. During this interaction, the sender does not use (m_0, m_1). The sender uses (m_0, m_1) to compute the final round message. We define two games: in the first game, the adversary tries to distinguish (m_0, m_1) versus (m_0, m_0) and in the second game, the adversary tries to distinguish (m_0, m_1) versus (m_1, m_1). We say that oblivious transfer satisfies post-quantum computational sender privacy property if the malicious receiver cannot succeed in both the games with non-negligible advantage.

A natural approach to prove that the malicious receiver cannot win both the games is to extract the bit β from the malicious receiver; if $\beta = 0$ then the receiver will not be able to succeed in the second game if $\beta = 1$ then the receiver will not succeed in the first game. To ensure that we can extract the bit β from the receiver, we additionally introduce an extraction phase to the protocol.

Extraction Phase: To design the extraction phase, we use the same technique we introduced earlier. The main difference is that instead of using statistical receiver-private OT, we instead use a statistical sender-private OT for extraction.

In the extraction phase of Π_{SROT}, the sender and the receiver do the following:

- As before, the sender of Π_{SROT}, plays the role of the receiver of Π_{SSOT} and the receiver of Π_{SROT} plays the role of the sender of Π_{SSOT}.

- The sender does the following: it samples a bit b uniformly at random. It sets the Π_{SSOT}'s receiver's bit to be b.

- The receiver, on the other hand, samples $\alpha \xleftarrow{\$} \{0,1\}$ and sets the Π_{SSOT}'s sender's input to be (β, α) with probability $\frac{1}{2}$ and (α, β) with probability $\frac{1}{2}$.

- At the end of the execution of Π_{SSOT}, the receiver reveals the location of β – i.e., it sends 0 if (β, α) was used in Π_{SSOT} or it sends 1 if (α, β) was used.

Note that if the location matched with b then the sender can recover β, otherwise it cannot. With probability at most $\frac{1}{2}$, the sender can recover β. We can use the same error amplification technique (via secret sharing) introduced earlier to reduce the probability of success of the malicious sender to be negligible. On the other hand, we can design an extractor that uses Watrous rewinding, as mentioned earlier to recover the bit β with probability close to 1.

Template. Using the above ingredients, we now summarise the template to construct a statistical receiver-private oblivious transfer.

The sender, on input (m_0, m_1), and the receiver of Π_{SROT} on input β, do the following:

- The sender and the receiver execute the extraction phase described above. The receiver uses its bit β in the extraction phase.

- The sender and the receiver then execute Π_{SSOT}, where each party play the opposite role. The sender sets the input of the receiver in Π_{SSOT} to be r, where $r \xleftarrow{\$} \{0,1\}$ and the receiver sets the input of the sender in Π_{SSOT} to be $(r', r' \oplus \beta)$, where $r' \xleftarrow{\$} \{0,1\}$. After the end of the execution of Π_{SSOT}, the sender recovers $\widetilde{r}$.

- Of course, the receiver could have cheated and used a different β in both the extraction phase and in the execution of Π_{SSOT}. To ensure that the receiver does not cheat, we force the receiver to prove that it used β consistently. We use a computational argument system satisfying statistical zero-knowledge property for this step.

- Once the sender gets convinced that the receiver did not cheat, it sends $(\widetilde{r} \oplus m_0, \widetilde{r} \oplus r \oplus m_1)$ to the receiver.

Finally, we show how to implement computational argument system satisfying statistical zero-knowledge property from QLWE. The idea is to start with a statistical NIZK computational argument system in the CRS model and then generate the CRS using a coin flipping protocol.

Quantum PoK in the Bounded Concurrent Setting

Our construction of bounded concurrent quantum proof of knowledge is the same as the one described in Section 5.1, except that we instantiate Π using the bounded concurrent QZK protocol that we constructed in Section 4.3.3[4].

However, proving the bounded concurrent QZK protocol turns out to be even more challenging than the standalone setting. To grasp the underlying difficulties, let us revisit the proof of QZK in Section 5.1. To prove the indistinguishability of the real and the ideal world, we first simulated the protocol Π. Since we are in the bounded concurrent setting, the simulator of Π is now simultaneously simulating multiple sessions of the verifier. Then using a sequence of intermediate hybrids, we changed the inputs used in the OT executions of all the sessions one at a time. However, in the bounded concurrent setting, the OT messages can be interleaved with QZK messages. This means that the simulator of QZK could be rewinding the OT messages along with the QZK messages. This makes it difficult to invoke the security of OT.

To reduce the indistinguishability of hybrids to breaking OT, we will carefully design the security reduction such that it does not rewind the blocks (the definition of a block is the same as the one described in Section 4.3) containing the messages of the OT protocol. This ensures that we can embed the messages exchanged with the external challenger (in the OT game) without the fear of being rewound. Of course, we need to be cautious: the decision to not rewind a specific block could leak information about the private state of the verifier and this could affect the zero-knowledge propety of the underlying QZK protocol. To overcome this issue, for a block containing the OT messages, we perform a dummy rewind where the transcript of conversation in this block does not change. Thus, we can still interact with the external challenger using the messages in this block. Another issue that arises is that we might end up not rewinding as many blocks as the round complexity of the underlying OT protocol, which is polynomially many rounds. We show that the simulator of the bounded concurrent QZK we constructed in Section 4.3 can be modified in such a way that it can successfully simulate all the sessions even if polynomially many blocks are ignored.

5.2 Receiver statistical oblivious transfer

We begin by presenting the definition of statistical receiver oblivious transfer with post-quantum security. We consider a natural adaption of the definition of [GJJM20] (see also [DGH+20]), who originally defined in the classical setting.

The definition we provide is written this way to make it compatible with the 3-round OT definition from [GJJM20]. The main difference is that we allow for an interactive phase instead of the sender's first message in [GJJM20].

[4]We emphasize that we use the specific bounded concurrent QZK protocol that we constructed earlier and we do not know how to provide a generic transformation.

5.2.1 Definition

Definition 96 (Post-Quantum Statistical Receiver-Private Oblivious Transfer)**.** *An* **oblivious transfer protocol**, Π_{OT}, *is an interactive protocol between a PPT sender and a PPT receiver* (S, R)*, and a triplet of algorithms* $(\mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$ *such that*

Interactive Phase. S *and* P *interact for* $\mathrm{poly}(\lambda)$ *rounds. The receiver's input is* λ *and a bit* $\beta \in \{0,1\}$*. The sender's input is* λ*. Let* ot_1 *be the transcript generated in this round, and let* st_S *and* st_R *be the private state of the sender and receiver (respectively) at the end of the round.*

Receiver's Final Message. *The receiver* R *computes* $(\mathsf{ot}_2, \mathsf{st}'_R) \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, \beta, \mathsf{st}_R)$

Sender's Final Message. S *with input* $(m_0, m_1) \in \{0,1\}^2$ *computes* $(\mathsf{ot}_3, \mathsf{st}'_S) \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, \mathsf{st}_S, m_0, m_1)$*, and it sends* ot_3 *to* R*.*

Reconstruction. *The receiver computes* $m' \leftarrow \mathsf{OT}_4(1^\lambda, \mathsf{ot}_3, \mathsf{st}'_R)$*. Output* m'*.*

Correctness. *For any* $\beta \in \{0,1\}, (m_0, m_1) \in \{0,1\}^2$*, we have:*

$$\mathsf{Pr}\left[\begin{array}{c}(\mathsf{ot}_1,\mathsf{st}_S,\mathsf{st}_R)\leftarrow\langle\mathsf{S}(1^\lambda),\mathsf{R}(1^\lambda,\beta)\rangle \\ (\mathsf{ot}_2,\mathsf{st}'_R)\leftarrow\mathsf{OT}_2(1^\lambda,\mathsf{ot}_1,\beta,\mathsf{st}_R) \\ (\mathsf{ot}_3,\mathsf{st}'_S)\leftarrow\mathsf{OT}_3(1^\lambda,\mathsf{ot}_2,\mathsf{st}_S,m_0,m_1) \\ m'\leftarrow\mathsf{OT}_4(1^\lambda,\mathsf{ot}_3,\mathsf{st}'_R)\end{array} \;:\; m' = m_\beta\right] = 1$$

Statistical Receiver-Privacy. *For any sender* S^**, denote* $(\mathsf{ot}_1^{(0)}, \mathsf{st}_R^{(0)}) \leftarrow \langle S^*(1^\lambda), R(1^\lambda, 0)\rangle$ *and* $(\mathsf{ot}_1^{(1)}, \mathsf{st}_R^{(1)}) \leftarrow \langle S^*(1^\lambda), R(1^\lambda, 1)\rangle$*. Furthermore, let* $(\mathsf{ot}_2^{(0)}, (\mathsf{st}_R^{(0)})') \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1^{(0)}, 0, \mathsf{st}_R^{(0)})$ *and let* $(\mathsf{ot}_2^{(1)}, (\mathsf{st}_R^{(1)})') \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1^{(1)}, 1, \mathsf{st}_R^{(1)})$*.*

Then the statistical distance between the marginal distributons $\{(\mathsf{ot}_1^{(0)}, \mathsf{ot}_2^{(0)})\}$ *and* $\{(\mathsf{ot}_1^{(1)}, \mathsf{ot}_2^{(1)})\}$ *is a negligible function in* λ*.*

Post-Quantum Sender-Privacy. *For any non-uniform QPT distinguisher* $\mathcal{A}$ *and any malicious receiver* R^**, which receives as input state that is possibly entangled with the input state of* $\mathcal{A}$*, we define the following games.*

Interact with R^*. *The challenger plays the role of an honest sender in the interactive phase with* R^**. Then the receiver* R^* *outputs a state in a register* $\mathbf{B}$ *and a message* z*. The message* z *is sent to the challenger. The register* $\mathbf{B}$ *is given to* $\mathcal{A}$*.*

Game $G_0(m_0, m_1)$*: The challenger samples* $b_0 \leftarrow \{0,1\}$ *at random and computes* $ot_3 \leftarrow \mathsf{OT}_3(1^\lambda, z, \mathsf{st}_S, m_{b_0}, m_1)$*. Then,* ot_3 *is sent to* $\mathcal{A}$*. Finally,* $\mathcal{A}$ *outputs two bits* b'_0 *and* b'_1*. If* $b_0 = b'_0$ *then we say that* $\mathcal{A}$ *wins the game* G_0*.*

Game $G_1(m_0, m_1)$*: The challenger samples* $b_1 \xleftarrow{\$} \{0,1\}$ *at random, and then computes* $ot_3 \leftarrow \mathsf{OT}_3(1^\lambda, z, \mathsf{st}_S, m_0, m_{b_1})$*. Then,* ot_3 *is sent to* $\mathcal{A}$*. Finally,* $\mathcal{A}$ *outputs two*

bits b'_0 and b'_1. If $b_1 = b'_1$ then we say that $\mathcal{A}$ wins the game G_1.

We define the advantage as follows:

$$\mathsf{Adv}(\mathcal{A}, \mathsf{R}^*, m_0, m_1) = \mathbb{E}_{\mathsf{View}_{R^*}} \left[\min \{p_0, p_1\}\right],$$

where:

- $p_0 = \left|\mathsf{Pr}[\mathcal{A} \text{ wins } \mathsf{G}_0(m_0, m_1)] - \frac{1}{2}\right|$
- $p_1 = \left|\mathsf{Pr}[\mathcal{A} \text{ wins } \mathsf{G}_1(m_0, m_1)] - \frac{1}{2}\right|$

We say that the oblivious transfer scheme is **(quantum) computational sender-secure** *if for every $m_0, m_1 \in \{0,1\}$, we have $\mathsf{Adv}(\mathcal{A}, \mathsf{R}^*, m_0, m_1)$ to be negligible in λ.*

5.2.2 Tool: Statistical ZK quantum argument system

To construct a statistical receiver-private oblivious transfer, we use two tools: (i) a statistical zero-knowledge argument system and, (ii) statistical sender-private oblivious transfer (Section 2.4.9).

In this section, we show how to obtain a statistical ZK quantum argument starting from a statistical NIZK (Section 2.4.6). We start by defininining a statistical ZK quantum argument system.

Definition 97 (Statistical ZK Quantum Argument System)**.** *Let Π be an interactive protocol between a classical PPT prover P and a classical PPT verifier V. Let $\mathcal{R}(\mathcal{L})$ be the NP relation associated with Π.*

Π is said to satisfy **completeness** *if the following holds:*

- **Completeness***: For every $(x, w) \in \mathcal{R}(\mathcal{L})$,*

$$\mathsf{Pr}[\mathsf{Accept} \leftarrow \langle P(x, w), V(x)\rangle] \geq 1 - negl(\lambda),$$

  for some negligible function negl.

Π is said to satisfy **(quantum computational) soundness** *if the following holds:*

- **(Quantum Computational) Soundness***: For every QPT adversary P^*, every $x \notin \mathcal{R}(\mathcal{L})$, every $\mathrm{poly}(\lambda)$-qubit advice ρ,*

$$\mathsf{Pr}\left[\mathsf{Accept} \leftarrow \langle P^*(x, \rho), V(x)\rangle\right] \leq negl(\lambda),$$

  for some negligible function negl.

Π is said to satisfy **statistical zero-knowledge** *if the following holds:*

- **Statistical Zero-Knowledge**:*For every sufficiently large $\lambda \in \mathbb{N}$, every computationally unbounded adversary V^*, there exists a QPT simulator* Sim *such that for every $(x, w) \in \mathcal{R}(\mathcal{L})$, the state output by V^* is close in trace distance to the state output by* Sim.

To construct a statistical ZK quantum argument system, we will use a non-interactive (statistical) ZK protocol for NP (NIZK).

Construction

In order to construct a statistical ZK quantum argument system for an NP relation $\mathcal{R}(\mathcal{L})$, we use the following ingredients.

Tools.

- A quantum zero-knowledge protocol Π_{zk} for the NP relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$. We described the relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$, parametrized by security parameter λ, described below:
$$\mathcal{R}(\mathcal{L}_{\mathsf{zk}}) = \{((crs, \mathbf{c}, b); (a, \mathbf{r})) : \overset{crs=a\oplus b}{\underset{\mathbf{c}=\mathsf{Comm}(1^\lambda,a;\mathbf{r})}{\bigwedge}}\}$$
- A perfectly binding and quantum computationally hiding commitment scheme, Comm, where the length of randomness is λ (Section 2.4.2).
- A non-interactive statistical zero-knowledge argument system Π_{NIZK} for $\mathcal{R}(\mathcal{L})$, where the length of the CRS is $q(\lambda)$ (Section 2.4.6).

We present a construction in Figure 5-1.

Completeness. The completeness follows from the completeness of Π_{zk} and Π_{NIZK}.

Quantum Computational Soundness. Let $x \notin \mathcal{L}$. Suppose P^* be a QPT prover that on input (x, ρ), for some poly(λ)-qubit advice ρ, convinces the verifier V^* to accept x with probability ε. We prove that ε is negligible using a hybrid argument.

$\underline{\mathsf{Hybrid}_1}$: This corresponds to the execution of P^* and V. The probability that V accepts is ε.

$\underline{\mathsf{Hybrid}_{2.i} \text{ for } i \in [q(\lambda)]}$: We consider a hybrid verifier $\mathsf{Hybrid}_{2.i}.V$ that executes the simulator Sim in the i^{th} execution of Π_{zk}, instead of running the real prover. Except this change, the hybrid verifier $\mathsf{Hybrid}_{2.i}.V$ behaves the same as $\mathsf{Hybrid}_{2.i-1}.V$ if $i > 1$ or as V if $i = 1$.

From the (computational) quantum zero-knowledge property of Π_{zk}, the probability that $\mathsf{Hybrid}_{2.i}.V$ accepts is negligibly close to ε.

$\underline{\mathsf{Hybrid}_{3.i}, \text{ for } i \in [q(\lambda)]}$: We consider a hybrid verifier $\mathsf{Hybrid}_{3.i}.V$ that computes the i^{th} commitment $\mathbf{c}_i$ as follows: $\mathbf{c}_i \leftarrow \mathsf{Comm}(1^\lambda, 0)$. Except this change, the hybrid

Instance: x.
Witness: w.

- V samples $\mathbf{a} \xleftarrow{\$} \{0,1\}^\lambda$.
- For every $i \in [q(\lambda)]$, P and V perform the following operations:
  - $V \to P$: V computes $\mathbf{c}_i \leftarrow \mathsf{Comm}(1^\lambda, a_i; \mathbf{r}_i)$, where $\mathbf{r}_i \xleftarrow{\$} \{0,1\}^\lambda$ and a_i is the i^{th} bit of $\mathbf{a}$. V sends $\mathbf{c}_i$ to P.
  - $P \to V$: P sends b_i to V, where $b_i \xleftarrow{\$} \{0,1\}$.
  - $V \to P$: V sends $crs_i = a_i \oplus b_i$ to P.
  - $V \leftrightarrow P$: P and V will execute Π_{zk}, with P playing the role of verifier in Π_{zk} and V plays the role of the prover. The instance is $(crs_i, \mathbf{c}_i, b_i)$ and the witness is $(a_i, \mathbf{r}_i)$.
- Set $\mathsf{crs} = \big(crs_1, \ldots, crs_{q(\lambda)}\big)$.
- $P \to V$: P computes a proof π on input common random string crs, instance x and witness w using Π_{NIZK}. It sends π to V.
- V computes the verifier of Π_{NIZK} on input (crs, x, π). It outputs the decision bit of the verifier of Π_{NIZK}.

Figure 5-1: Statistical ZK Quantum Argument System

verifier $\mathsf{Hybrid}_{3.i}.V$ behaves the same as $\mathsf{Hybrid}_{3.i-1}.V$ if $i > 1$ or as $\mathsf{Hybrid}_{2.q(\lambda)}.V$ if $i = 1$.

From the quantum-concealing property of Comm, the probability that $\mathsf{Hybrid}_{3.i}.V$ accepts is negligibly close to ε.

$\underline{\mathsf{Hybrid}_4}$: We consider a hybrid verifier $\mathsf{Hybrid}_4.V$, which is essentially the same as $\mathsf{Hybrid}_{3.q(\lambda)}.V$, except that it generates $\mathsf{crs} \xleftarrow{\$} \{0,1\}^{q(\lambda)}$ and sends crs to P.

Since the hybrids $\mathsf{Hybrid}_{3.q(\lambda)}.V$ and $\mathsf{Hybrid}_4.V$ are identical, the probability that $\mathsf{Hybrid}_4.V$ accepts is negligibly close to ε.

From the computational soundness of Π_{NIZK}, the probability that $\mathsf{Hybrid}_4.V$ accepts is negligible. Thus, ε is negligible.

Statistical Zero-Knowledge. Let V^* be a computationally unbounded verifier and let $|\Psi\rangle$ be the initial state of V^*. Before we describe the simulator we first define

the following registers. For $i = 1, \ldots, q(\lambda)$:

- $\mathbf{B}_i$: it contains the bit sent by the simulator in the i^{th} iteration.
- $\mathbf{R}_i$: it contains the receiver's commitment and the i^{th} bit of crs sent during the i^{th} iteration.
- $\mathbf{IZ}_i$: it contains the messages of zero-knowledge exchanged during the i^{th} iteration.
- **Dec**: it contains the decision bit.
- **Aux**: it contains the auxiliary state of the verifier.
- **NZ**: it contains the final NIZK proof sent by the simulator.
- **C**: it contains the common reference string.
- **X**: this is a poly(λ)-qubit ancillary register.

Description of Simulator:

- The simulator prepares the following state:

$$|\Psi_1\rangle = \left(\bigotimes_{i=1}^{q(\lambda)} |0\rangle_{\mathbf{R}_i} |0\rangle_{\mathbf{B}_i} |0\rangle_{\mathbf{IZ}_i}\right) \otimes |0\rangle_{\mathbf{NZ}} |0\rangle_{\mathbf{X}} |0\rangle_{\mathbf{C}} |\Psi\rangle_{\mathbf{Aux}} |0\rangle_{\mathsf{Dec}}$$

- It runs the NIZK simulator, $(\widehat{\mathsf{crs}}, \widehat{\pi}) \leftarrow \Pi_{\mathsf{NIZK}}.\mathsf{Sim}(1^\lambda, x)$, to compute $\widehat{\mathsf{crs}}$. It stores $\widehat{\mathsf{crs}}$ in the register $\mathbf{C}$, and it stores $\widehat{\pi}$ in $\mathbf{NZ}$.
- For all $i = 1, \ldots, q(\lambda)$, let U_i be the unitary that performs the following operations in superposition.
  - It first applies V^* on the registers $\{\mathbf{B}_j, \mathbf{R}_j, \mathbf{IZ}_j, \mathbf{Aux}\}_{j<i}, \mathbf{R}_i$.
  - It then maps $|0\rangle_{\mathbf{B}_i}$ to $|+\rangle_{\mathbf{B}_i}$.
  - It then applies V^* on the registers $\{\mathbf{B}_j, \mathbf{R}_j, \mathbf{IZ}_j, \mathbf{Aux}\}_{j<i}, \mathbf{R}_i, \mathbf{B}_i$.
  - It then performs the i^{th} iteration of Π_{zk} with V^* in superposition. The transcript is stored in $\mathbf{IZ}_i$.
  - It then applies the following unitary $\widehat{U}$:

$$\widehat{U}|b_i\rangle_{\mathbf{B}_i}|\mathbf{c}_i\ crs_i\rangle_{\mathbf{R}_i}|\tau_i\rangle_{\mathbf{IZ}_i}|\widehat{\mathsf{crs}}\rangle_{\mathbf{C}}|0\rangle_{\mathbf{Dec}} = \begin{cases} |b_i\rangle_{\mathbf{B}_i}|\mathbf{c}_i\ crs_i\rangle_{\mathbf{R}_i}|\tau_i\rangle_{\mathbf{IZ}_i}|\widehat{\mathsf{crs}}\rangle_{\mathbf{C}}|1 \oplus \theta_i\rangle_{\mathbf{Dec}}, & \text{if } \tau_i \text{ is valid,} \\ |b_i\rangle_{\mathbf{B}_i}|\mathbf{c}_i\ crs_i\rangle_{\mathbf{R}_i}|\tau_i\rangle_{\mathbf{IZ}_i}|\widehat{\mathsf{crs}}\rangle_{\mathbf{C}}|+\rangle_{\mathbf{Dec}}, & \text{othewise} \end{cases}$$

  We define $\theta_i = 1$ if the i^{th} bit of $\widehat{\mathsf{crs}}$ is the same as crs_i, where crs_i is the i^{th} bit of crs computed by V^* in the register $\mathbf{R}_i$. Furthermore, we define τ_i to be valid if the verifier in the i^{th} execution of Π_{zk} accepts.

- Let $W_i = \mathsf{Amplifier}\,(U_i)$; where $\mathsf{Amplifier}$ is the circuit guaranteed by Lemma 28. Simulator computes $|\Psi_i\rangle = W_i|\Psi_{i-1}\rangle$.
- Finally, it uses $\widehat{\pi}$ stored in $\mathbf{NZ}$ as the proof for the NIZK step.
- Measure all the registers except for $\mathbf{Aux}$.

We now prove the statistical indistinguishability of the real and the ideal world using a hybrid argument. Consider the following hybrids.

$\underline{\mathsf{Hybrid}_1}$: This corresponds to the real execution between P and V^*.

$\underline{\mathsf{Hybrid}_{2.i} \text{ for } i \in [q(\lambda)]}$: We define a hybrid prover as follows: sample $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda)$. Note that crs is generated according to the uniform distribution. Prepare the state $|\Psi_1\rangle$ as given in the description of the simulator. Apply $W_i \cdots W_1|\Psi_1\rangle$ to obtain $|\Psi_i\rangle$. That is, perform Watrous rewinding for the first i iterations of the OT protocol, similarly to the way the simulator does, but using crs, instead of $\widehat{\mathsf{crs}}$. Then, the hybrid prover uses the real prover to interact with V^*, that receives as input $|\Psi_i\rangle$, to perform the operations for the rest of the protocol.

We now show that the output distributions of the hybrids $\mathsf{Hybrid}_{2.i-1}$ and $\mathsf{Hybrid}_{2.i}$ are computationally indistinguishable. In order to show this, we use a similar argument that was used in the proof of Claim 81. It suffices to argue that the following distributions are statistically close:

- $\mathcal{D}_1$: Measure the registers $\{\mathbf{R}_i, \mathbf{IZ}_i\}_{i\in[q(\lambda)]}, \mathbf{NZ}$ after the i^{th} iteration in $\mathsf{Hybrid}_{2.i-1}$ and output the measurement outcome along with the residual state in $\mathbf{Aux}$.
- $\mathcal{D}_2$: Measure the registers $\{\mathbf{R}_i, \mathbf{IZ}_i\}_{i\in[q(\lambda)]}, \mathbf{NZ}$ after the i^{th} iteration in $\mathsf{Hybrid}_{2.i}$ and output the measurement outcome along with the residual state in $\mathbf{Aux}$.

We prove this in two steps: first, we apply Watrous rewinding and analyze the state obtained after the i^{th} iteration in $\mathsf{Hybrid}_{2.i}$. In the next step, we use this to argue the indistinguishability of $\mathcal{D}_1$ and $\mathcal{D}_2$.

Applying Watrous Rewinding. Suppose $|\Psi_{i-1}\rangle = W_{i-1}\cdots W_1|\Psi_1\rangle$. Consider the following:

$$\begin{aligned} U_i|\Psi_{i-1}\rangle &= \sqrt{q}\left(\sqrt{p}|\phi_{\mathsf{good}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p}|\phi_{\mathsf{bad}}\rangle|1\rangle_{\mathbf{Dec}}\right) + \sqrt{1-q}|\phi_{\mathsf{invalid}}\rangle|+\rangle_{\mathbf{Dec}}, \\ &= \sqrt{p}\left(\sqrt{q}|\phi_{\mathsf{good}}\rangle + \sqrt{1-q}|\phi_{\mathsf{invalid}}\rangle\right)|0\rangle_{\mathbf{Dec}} + \sqrt{1-p}\left(\sqrt{q}|\phi_{\mathsf{bad}}\rangle + \sqrt{1-q}|\phi_{\mathsf{invalid}}\rangle\right)|1\rangle_{\mathbf{Dec}}, \end{aligned}$$

where:

- q is the probability with which V^* convinces P in the i^{th} execution of Π_{zk},
- $\left|p - \frac{1}{2}\right| \leq \mathrm{negl}(\lambda)$: this follows from a similar argument as in the proof of Claim 83,

- $|\phi_{\mathsf{invalid}}\rangle$ (defined on all the registers except the **Dec** register) is a superposition over the messages containing the i^{th} iteration Π_{zk} transcripts that are not accepted by the verifier of Π_{zk},

- $|\phi_{\mathsf{good}}\rangle$ (defined on all the registers except the **Dec** register) is a superposition over the messages of the i^{th} iteration containing $crs_i = \mathsf{crs}_i$ and,

- $|\phi_{\mathsf{bad}}\rangle$ (defined on all the registers except the **Dec** register) is a superposition over the messages of the i^{th} iteration containing $crs_i \neq \mathsf{crs}_i$.

Once we apply Lemma 28, the resulting state will be $W_i|\Psi_{i-1}\rangle = \left(\sqrt{q}|\phi_{\mathsf{good}}\rangle + \sqrt{1-q}|\phi_{\mathsf{invalid}}\rangle\right)|0\rangle_{\mathbf{Dec}}$ with probability negligibly close to 1.

Indistinguishability of $\mathcal{D}_1$ and $\mathcal{D}_2$. As in the proof of Claim 81, it suffices to argue that the distribution of measurements of $\{\mathbf{R}_i, \mathbf{IZ}_i\}_{i\in[q(\lambda)]}, \mathbf{NZ}$ in $|\phi_{\mathsf{good}}\rangle$ along with the residual state in **Aux** is computationally indistinguishable from the distribution of measurements of $\{\mathbf{R}_i, \mathbf{IZ}_i\}_{i\in[q(\lambda)]}, \mathbf{NZ}$ in $|\phi_{\mathsf{bad}}\rangle$ along with the residual state in **Aux**. This follows from the perfect binding property of Comm and the statistical soundness property of Π_{zk} using a similar argument used in Claim 83: if the verifier is not computed in superposition then the verifier cannot distinguish whether $crs_i = \mathsf{crs}_i$ or whether $crs_i \neq \mathsf{crs}_i$. Moreover, this is true even if the verifier is computed in superposition and measuring the transcript registers in the end.

$\underline{\mathsf{Hybrid}_3}$: Execute the simulator on input the state $|\Psi\rangle$.

From the statistical zero-knowledge property of Π_{NIZK}, it follows that the state output by V^* in the hybrid $\mathsf{Hybrid}_{2.q(\lambda)}$ is close in trace distance to the state output by V^* in the hybrid Hybrid_3.

5.2.3 Post-quantum statistical receiver OT: Construction

The ingredients needed for our construction are the following.

Tools.

- A 2-round post-quantum statistical sender-private OT, $\Pi_{OT} = (OT_1, OT_2, OT_3)$ (Section 2.4.9). Without loss of generality, we assume that the length of the randomness is λ.

  We say that a transcript τ, consisting of messages $(\mathsf{msg}_1, \mathsf{msg}_2)$, is valid with respect to sender's randomness r and its input bits (m_0, m_1) if the following holds: $(\mathsf{msg}_2, \mathsf{st}) \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{msg}_1, m_0, m_1; r)$.

- A statistical zero-knowledge quantum argument system, Π_{zk}, for the NP relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$ (Section 5.2.2). We described the relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$, parametrized by security parameter λ, described below:

$$\mathcal{R}\left(\mathcal{L}_{\mathsf{zk}}\right) = \Bigg\{ \left(\left(\tau^*_{\mathsf{OT}}, \{\tau^{(i,j)}_{\mathsf{OT}}, b_{i,j}\}_{i\in[\lambda+2], j\in[\lambda]} \right); \left(r', \beta, r^*_{\mathsf{OT}}, \left\{ r^{(i,j)}_{\mathsf{OT}}, sh_{i,j}, \alpha_{i,j} \right\}_{i\in[\lambda+2], j\in[\lambda]} \right) \right) :$$

$$\left(\begin{array}{c} \forall i\in[\lambda+2], j\in[\lambda], \\ \tau^{(i,j)}_{\mathsf{OT}} \text{ is valid w.r.t} \\ r^{(i,j)}_{\mathsf{OT}} \text{ and } (((1-b_{i,j})sh_{i,j}+b_{i,j}\cdot\alpha_{i,j}),\ (b_{i,j}sh_{i,j}+(1-b_{i,j})\cdot\alpha_{i,j})) \\ \bigwedge \\ \tau^*_{\mathsf{OT}} \text{ is valid w.r.t } r^*_{\mathsf{OT}} \text{ and } (r', r'\oplus\beta) \end{array} \right) \bigwedge \left(\begin{array}{c} \forall i\in[\lambda+2], \\ \oplus^{\lambda}_{j=1} sh_{i,j} = w_i \end{array} \right) \bigwedge w = (r', \beta, r^*_{\mathsf{OT}}) \Bigg\}$$

We show that the construction in Figure 5-2 is a post-quantum statistical receiver oblivious transfer protocol.

Correctness. The correctness of our protocol follows from the correctness of Π_{OT} and the completeness of Π_{zk}.

Statistical Receiver Privacy. Let S^* be a computationally unbounded sender. Instead of proving that the sender cannot distinguish receiver's bit to be 0 versus receiver's bit to be 1 with non-negligible probability, we instead prove the following: suppose receiver chooses its bit uniformly at random then the probability that the malicious sender can output β with probability negligibly close to $\frac{1}{2}$. We prove this via a hybrid argument. In the first hybrid, the receiver behaves honestly and uses the receiver's bit to be β, where β is chosen uniformly at random. We define a sequence of hybrids and show computational indistinguishability of every pair of consecutive hybrids. In the final hybrid, the receiver's bit will be information-theoretically hidden in the messages exchanged with S^*, which will prove the statistical receiver privacy property.

$\underline{\mathsf{Hybrid}_1}$: In this hybrid, the receiver uses the bit β.

Let ε be the probability that S^* outputs β.

$\underline{\mathsf{Hybrid}_2}$: Let $\mathsf{Sim}_{\mathsf{zk}}$ be the simulator associated with Π_{zk}. Instead of R playing the role of the prover in Π_{zk}, it executes $\mathsf{Sim}_{\mathsf{zk}}$.

From the statistical zero-knowledge property of Π_{zk}, the output distributions of S^* in the hybrids Hybrid_1 and Hybrid_2 are statistically close. The probability that S^* outputs β is negligibly close to ε in this hybrid.

$\underline{\mathsf{Hybrid}_{3.(i,j)}}$, for $i \in [\lambda+2]$, $j \in [\lambda]$: In the $(i,j)^{th}$ execution of Π_{OT}, perform inefficient extraction to extract $b'_{i,j}$ from R'_{OT}. Recall that S^* plays the role of R'_{OT}. If $b'_{i,j} = b_{i,j}$ then set the input of the sender S'_{OT} to be $(sh_{i,j}, sh_{i,j})$ and if $b'_{i,j} \neq b_{i,j}$ then set the input of the sender S'_{OT} to be $(\alpha_{i,j}, \alpha_{i,j})$.

The statistical indistinguishability of this hybrid and the previous hybrid follows from the statistical sender-privacy property of Π_{OT}. The probability that S^* outputs β is negligibly close to ε in this hybrid.

Input of sender S: (m_0, m_1).
Input of receiver R: β.

- R generates $r' \xleftarrow{\$} \{0,1\}$ uniformly at random. R samples $r^*_{OT} \xleftarrow{\$} \{0,1\}^\lambda$.
- Let $w = (r', \beta, r^*_{OT})$. For every $i \in [\lambda + 2]$, R generates shares $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random conditioned on $\oplus_{j=1}^{\lambda} sh_{i,j} = w_i$.
- For $i \in [\lambda + 2]$, R also generates bits $\alpha_{i,1}, \ldots, \alpha_{i,\lambda}$ uniformly at random.
- For $i \in [\lambda + 2], j \in [\lambda]$, do the following:
  - $S \leftrightarrow R$: S and R execute Π_{OT} with S playing the role of the receiver, denoted by R'_{OT}, in Π_{OT} and R playing the role of the sender, denoted by S'_{OT}, in Π_{OT}. The input of the receiver R'_{OT} in this protocol is 0, while the input of the sender S'_{OT} is set to be $(sh_{i,j}, \alpha_{i,j})$ if $b_{i,j} = 0$, otherwise it is set to be $(\alpha_{i,j}, sh_{i,j})$ if $b_{i,j} = 1$, where the bit $b_{i,j}$ is sampled uniformly at random by S'_{OT}. We call this execution as $(i,j)^{th}$ execution of Π_{OT}.

    Call the resulting transcript of the protocol to be $\tau^{(i,j)}_{OT}$ and let $r^{(i,j)}_{OT}$ be the randomness used by the sender S'_{OT} in Π_{OT}.
  - $R \to S$: R sends $b_{i,j}$ to S.
- S samples $r \xleftarrow{\$} \{0,1\}$.
- $S \leftrightarrow R$: S and R execute Π_{OT} with S playing the role of the receiver, denoted by R'_{OT}, in Π_{OT} and R playing the role of the sender, denoted by S'_{OT}, in Π_{OT}. The input of the receiver R'_{OT} is r and the input of the sender is $(r', r' \oplus \beta)$. Let $\widetilde{r}$ be the bit recovered by R'_{OT} at the end of the protocol.

  We call this execution the main execution of Π_{OT}. Call the resulting transcript of the protocol to be τ^*_{OT} and let r^*_{OT} be the randomness used by the sender S'_{OT} in Π_{OT}.
- $S \leftrightarrow R$: S and R execute Π_{zk} with R playing the role of the prover P of Π_{zk} and S playing the role of the verifier V of Π_{zk}. The instance is $\left(\tau^*_{OT}, \left\{\tau^{(i,j)}_{OT}, b_{i,j}\right\}_{i\in[\ell_w], j\in[\lambda]}\right)$ and the witness is $\left(r', \beta, r^*_{OT}, \left\{r^{(i,j)}_{OT}, sh_{i,j}, \alpha_{i,j}\right\}_{i\in[\ell_w], j\in[\lambda]}\right)$. If the verifier in Π_{zk} rejects, then S aborts.
- S sends $(\widetilde{r} \oplus m_0, \widetilde{r} \oplus r \oplus m_1)$.

Figure 5-2: Post-quantum statistical receiver oblivious transfer protocol

$\underline{\mathsf{Hybrid}_4}$: If there exists $i \in [\lambda+2]$, such that for every $j \in [\lambda]$, $b_{i,j} = b'_{i,j}$ then abort.

The probability that this hybrid aborts is at most $\frac{\lambda+2}{2^\lambda}$. Conditioned on this hybrid not aborting, this hybrid is identical to the previous one. The probability that S^* outputs β is negligibly close to ε in this hybrid.

$\underline{\mathsf{Hybrid}_5}$: In the main execution of Π_{OT}, perform inefficient extraction to extract r from Π_{OT}. If $r = 0$, then set the input of the sender S'_{OT} to be (r', r') and if $r = 1$, then set the input of the sender to be $(r' \oplus \beta, r' \oplus \beta)$.

The statistical indistinguishability of Hybrid_4 and Hybrid_5 follows from the statistical sender-privacy property of Π_{OT}. The probability that S^* outputs β is negligibly close to ε in this hybrid.

Note that in Hybrid_5, the receiver's bit β is information-theoretically hidden in the messages exchanged with S^*. Thus, the probability that S^* guesses β in Hybrid_5 is $\frac{1}{2}$. This proves that the probability that the receiver outputs β in Hybrid_1 is negligibly close to $\frac{1}{2}$.

Post-Quantum Sender Privacy. Let R^* be a QPT receiver and $\mathcal{A}$ be a QPT adversary such that the following holds for some $m_0 \in \{0,1\}, m_1 \in \{0,1\}$,

$$\mathbb{E}_{\mathsf{View}_{R^*}} \left[\min \{p_0, p_1\}\right] \geq \nu(\lambda),$$

where:

- View_{R^*} is the view of R^*.
- $p_0 = \left|\Pr[\mathcal{A} \text{ wins } \mathsf{G}_0(m_0, m_1)] - \frac{1}{2}\right|$
- $p_1 = \left|\Pr[\mathcal{A} \text{ wins } \mathsf{G}_1(m_0, m_1)] - \frac{1}{2}\right|$

for some non-negligible fucntion $\nu(\lambda)$, where G_0, G_1 are defined with respect to R^* and $\mathcal{A}$ as in Section 5.2.1. We define $p_0^{(\mathbf{i})}$ to be the absolute difference of the probability that $\mathcal{A}$ wins in the game G_0 and $\frac{1}{2}$ in the hybrid $\mathsf{Hybrid}_\mathbf{i}$. Similarly, we define $p_1^{(\mathbf{i})}$.

Consider the following hybrids.

$\underline{\mathsf{Hybrid}_1}$: This hybrid corresponds to the real execution of the protocol.

By our initial assumption, we have $\mathbb{E}_{\mathsf{View}_{R^*}} \left[\min \{p_0, p_1\}\right] \geq \nu(\lambda)$.

$\underline{\mathsf{Hybrid}_2}$: In this hybrid, defer the measurements of the receiver until the end.

The output distributions of Hybrid_1 and Hybrid_2 are identical. Thus, $\mathbb{E}_{\mathsf{View}_{R^*}} \left[\min \left\{p_0^{(2)}, p_1^{(2)}\right\}\right] \geq \nu(\lambda)$.

$\underline{\mathsf{Hybrid}_{3.(i,j)} \text{ for every } i \in [\lambda+2], j \in [\lambda]}$: Instead of computing S, perform the following hybrid extractor as follows.

We first give a description of the registers used in the system.

- $\mathbf{R}_{i,j}$ for $i \in [\lambda+2], j \in [\lambda]$: this consists of the sender S – recall that S is taking the role of the receiver R' of the $(i,j)^{th}$ execution of the OT – randomness used by the extractor in the $(i,j)^{th}$ executions of Π_{OT}.
- $\mathbf{B}_{i,j}$, for $i \in [\lambda+2], j \in [\lambda]$: this is a single-qubit register that contains a bit that is used by the extractor in the $(i,j)^{th}$ execution of the OT protocol.
- **Dec**: it contains the decision register that indicates whether to rewind or not.
- **Aux**: this is initialized with the auxiliary state of the receiver.
- $\mathbf{T}_{i,j}$, for $i \in [\lambda+2], j \in [\lambda]$: it contains the transcripts of the $(i,j)^{th}$ executions of the OT protocol.
- $\mathbf{T}^*$: it contains the transcript of the protocol Π_{zk}.
- $\mathbf{X}$: this is a poly(λ)-qubit ancillary register.

Description of $\mathsf{Hybrid}_{3.(i,j)}.\mathsf{Ext}(x, |\Psi\rangle)$: The state of the extractor is initialized as follows:

$$\left(\bigotimes_{i\in[\ell_w],j\in[\lambda]} |0\rangle_{\mathbf{B}_{i,j}}|0\rangle_{\mathbf{R}_{i,j}}|0\rangle_{\mathbf{T}_{i,j}}\right) \otimes |0\rangle_{\mathbf{T}^*} \otimes |\Psi\rangle_{\mathbf{Aux}} \otimes |0\rangle_{\mathbf{Dec}} \otimes |0^{\otimes \mathrm{poly}(\lambda)}\rangle_{\mathbf{X}}$$

- For $i' \in [\lambda+2], j' \in [\lambda]$ such that $(i',j') \geq (i,j)$, perform the following operations in superposition:
  - Let $|\widetilde{\Psi}\rangle$ be the state of the system at the beginning of the $(i,j)^{th}$ execution.
  - Prepare the following state[5]:

$$|0\rangle_{\mathbf{B}_{i,j}}|0\rangle_{\mathbf{R}_{i,j}} \rightarrow \frac{1}{\sqrt{2^{\lambda+1}}} \sum_{\beta_{i,j}\in\{0,1\}, s_{\mathsf{OT}}^{(i,j)}\in\{0,1\}^\lambda} |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}$$

  - It then performs the $(i,j)^{th}$ execution of Π_{OT} along with the R^*'s message immediately after the $(i,j)^{th}$ execution of Π_{OT} in superposition. The resulting transcript is stored in the register $\mathbf{T}_{i,j}$. We denote the unitary that performs this step to be $U_{i,j}^{(1)}$.
  - After R^* sends the message immediately after the $(i,j)^{th}$ execution of Π_{OT}, apply the unitary $U_{i,j}^{(2)}$ defined as follows:

$$\begin{aligned} & U_{i,j}^{(2)}|\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|0\rangle_{\mathbf{Dec}} \\ = & \begin{cases} |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|\mathsf{Match}_{i,j}\rangle_{\mathbf{Dec}} \text{ if } \mathsf{acc}_{i,j}=1, \\ |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|+\rangle_{\mathbf{Dec}}, \quad \text{otherwise} \end{cases} \end{aligned}$$

[5]We will assume, without loss of generality, that the length of the random strings used in the OT protocol be λ.

Here, $\mathsf{Match}_{i,j} = 0$ if and only if $\beta_{i,j} = b_{i,j}$, where $b_{i,j}$ is the bit sent by R^* after the $(i,j)^{th}$ execution of the OT protocol. Moreover, $\mathsf{acc}_{i,j} = 1$ only if R^* has not aborted in $(i,j)^{th}$ OT execution.

Let $W_{i,j} = \mathsf{Amplifier}\left(U^{(2)}_{i,j} U^{(1)}_{i,j}\right)$, where $\mathsf{Amplifier}$ is defined in Lemma 28. Perform $W_{i,j}|\widetilde{\Psi}\rangle$ to obtain $|\Psi_{i,j}\rangle$.

- For $i' \in [\lambda+2], j' \in [\lambda]$, such that $(i',j') < (i,j)$ perform the $(i',j')^{th}$ execution of Π_{OT} as in the previous hybrid.
- Perform the main execution of Π_{OT} in superposition.
- Perform the execution of Π_{zk} in superposition.
- Measure all the registers except **Aux**. Perform the OT reconstruction on input the measured transcript $\tau^{i,j}_{\mathsf{OT}}$, for $i \in [\lambda+2], j \in [\lambda]$, measured randomness $s^{i,j}_{\mathsf{OT}}$ and receiver's bit $b_{i,j}$. Call the resulting reconstruction output to be $\widetilde{sh_{i,j}}$. Let $\widetilde{u}_i = \oplus_{j=1}^{\ell_w} \widetilde{sh_{i,j}}$. Let (r', β, r^*_{OT}) be the concatenation of all the $\widetilde{u}_i$ bits. If either the S'_{OT}'s inputs to the main execution of Π_{OT} is not $(r', r' \oplus \beta)$ or if S'_{OT}'s randomness is not r^*_{OT} then abort. Otherwise output the state in **Aux** along with w.

From the post-quantum computational receiver privacy of Π_{OT}, it holds that $\mathsf{Hybrid}_{3.(i,j)}$ and the previous hybrid are computationally indistinguishable. Thus, the following holds:

$\mathbb{E}_{\mathsf{View}_{R^*}}\left[\min\left\{p_0^{(3.(i,j))}, p_1^{(3.(i,j))}\right\}\right] \geq \nu_{3.(i,j)}(\lambda)$, where $\nu_{3.(i,j)}$ is a non-negligible function.

$\underline{\mathsf{Hybrid}_4}$: In the main execution of Π_{OT}, the input of R'_{OT} is set to be 0. Recall that in the previous hybrids, the input of R'_{OT} was r.

From the post-quantum computational receiver privacy of Π_{OT}, it holds that the hybrids Hybrid_4 and $\mathsf{Hybrid}_{3.(\lambda+2,\lambda)}$ are computationally indistinguishable. Thus, the following holds:

$\mathbb{E}_{\mathsf{View}_{R^*}}\left[\min\left\{p_0^{(4)}, p_1^{(4)}\right\}\right] \geq \nu_4(\lambda)$, where ν_4 is a non-negligible function.

$\underline{\mathsf{Hybrid}_5}$: Sample $u \xleftarrow{\$} \{0,1\}$. If $\beta = 1$, set the last message to be $(u, r' \oplus m_1)$. Else if $\beta = 0$, set the last message to be $(r' \oplus m_0, u)$.

From the computational soundness of Π_{zk}, it follows that β extracted from all the $(i,j)^{th}$, for $i \in [\lambda+2], j \in [\lambda]$, executions of Π_{OT} is the same as the β used by the receiver in the main execution of Π_{OT} with probability negligibly close to 1. This further implies that the bit reconstructed by R'_{OT} in the main execution is $\widetilde{r} = r' \oplus (r \cdot \beta)$. Thus, the last message sent by S can be rewritten as follows: $(\widetilde{r} \oplus m_0, \widetilde{r} \oplus r \oplus m_1) = (r' \oplus (r \cdot \beta) \oplus m_0, r' \oplus (r \cdot \beta) \oplus r \oplus m_1)$. If $\beta = 1$, we have the message sent by S to be $(r' \oplus r \oplus m_0, r' \oplus m_1)$. If $\beta = 0$, we have the message

sent by S to be $(r' \oplus m_0, r' \oplus r \oplus m_1)$. We now use the fact that r is information-theoretically hidden from the receiver R^* to show that the hybrids Hybrid_4 and Hybrid_5 are computationally indistinguishable. Thus, the following holds:

$\mathbb{E}_{\mathsf{View}_{R^*}}\left[\min\left\{p_0^{(5)}, p_1^{(5)}\right\}\right] \geq \nu_5(\lambda)$, where ν_5 is a non-negligible function.

But one of the two sender's inputs are information-theoretically hidden from the malicious receiver; in one of the two games G_0 or G_1, the adversary can win only with negligible probability. This contradicts the fact that $\mathbb{E}_{\mathsf{View}_{R^*}}\left[\min\left\{p_0^{(5)}, p_1^{(5)}\right\}\right]$ is non-negligible. Thus, our construction satisfies post-quantum sender privacy.

5.3 Quantum proofs of knowledge

5.3.1 Definition

We present the definition of quantum proof of knowledge; this is the traditional notion of proof of knowledge, except that the unbounded prover could be a quantum algorithm and specifically, its intermediate states could be quantum states.

Definition 98 (Quantum Proof of Knowledge)**.** *We say that an interactive proof system (P, V) for a NP relation $\mathcal{R}$ satisfies (ε, δ)-proof of knowledge property if the following holds: suppose there exists a malicious (possibly computationally unbounded prover) P^* such that for every x, and quantum state ρ it holds that:*

$$\mathsf{Pr}\left[(\widetilde{\rho}, \mathsf{decision}) \leftarrow \langle P^*(x,\rho), V(x)\rangle \bigwedge \mathsf{decision} = \mathsf{accept}\right] = \varepsilon$$

Then there exists a quantum polynomial-time extractor Ext*, such that:*

$$\mathsf{Pr}\left[(\widetilde{\rho}', \mathsf{decision}, w) \leftarrow \mathsf{Ext}\,(x,\rho) \bigwedge \mathsf{decision} = \mathsf{accept}\right] = \delta$$

Moreover, we require $T(\widetilde{\rho}, \widetilde{\rho}') = negl(|x|)$, where $T(\cdot,\cdot)$ denotes the trace distance and negl is a negligible function.

We drop (ε, δ) from the notation if $|\delta - \varepsilon| \leq negl(|x|)$, for a negligible function negl.

Remark 99 (Comparison with Unruh's Proof of Knowledge [Unr12])**.** *Our definition is a special case of Unruh's quantum proof of knowledge definition. Any proof system satisfying our definition is a quantum proof of knowledge system (according to Unruh's definition) with knowledge error κ, for any κ. Moreover, in Unruh's definition, the extraction probability is allowed to be polynomially related to the acceptance probability whereas in our case, the extraction probability needs to be negligibly close to the acceptance probability.*

Definition 100 (Concurrent Quantum ZK PoK)**.** *We say that a concurrent (resp., bounded) quantum ZK is a concurrent (resp., bounded) QZKPoK if it satisfies proof of knowledge property.*

5.3.2 Construction of (Standalone) QZKPoK

In the next section, we construct a bounded concurrent QZK satisfying quantum proof of knowledge property assuming post-quantum statistical receiver-private oblivious transfer. We first start with a simpler case: we present a construction of quantum proof of knowledge for a standalone quantum ZK proof system for NP. We then show how to extend this construction to the bounded concurrent QZK setting.

We construct a (standalone) QZKPoK (P, V) for an NP relation $\mathcal{R}(\mathcal{L})$. The following tools are used in our construction.

Tools.

- A post-quantum statistical receiver-private oblivious transfer protocol, $\Pi_{\mathsf{OT}} = (\mathsf{S}, \mathsf{R})$ (Section 5.2) satisfying perfect correctness property.

  We say that a transcript τ is valid with respect to sender's randomness r and its input bits (m_0, m_1) if τ can be generated with a sender that uses r as randomness for the protocol and uses (m_0, m_1) as inputs.

- A (standalone) QZK proof system Π_{zk} for $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$. We describe the relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$, parameterized by security parameter λ, below.

$$\mathcal{R}\left(\mathcal{L}_{\mathsf{zk}}\right) = \Bigg\{ \left(\left(x, \{\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\}_{i\in[\ell_w], j\in[\lambda]} \right) ; \left(w, \left\{ r_{\mathsf{OT}}^{(i,j)}, sh_{i,j}, \alpha_{i,j} \right\}_{i\in[\ell_w], j\in[\lambda]} \right) \right) :$$

$$\left(\begin{array}{c} \forall i\in[\ell_w], j\in[\lambda], \\ \tau_{\mathsf{OT}}^{(i,j)} \text{ is valid w.r.t} \\ r_{\mathsf{OT}}^{(i,j)} \text{ and } (((1-b_{i,j})sh_{i,j}+b_{i,j}\cdot\alpha_{i,j}),\ (b_{i,j}sh_{i,j}+(1-b_{i,j})\cdot\alpha_{i,j})) \end{array} \right) \bigwedge \left(\begin{array}{c} \forall i\in[\ell_w], \\ \oplus_{j=1}^{\lambda} sh_{i,j}=w_i \end{array} \right) \bigwedge (x,w) \in \mathcal{R}(\mathcal{L}) \Bigg\}$$

  In other words, the relation checks if the shares $\{sh_{i,j}\}$ used in all the OT executions so far are defined to be such that the XOR of the shares $sh_{i,1}, \ldots, sh_{i,\lambda}$ yields the bit w_i. Moreover, the relation also checks if $w_1 \cdots w_{\ell_w}$ is the witness to the instance x.

We describe the construction in Figure 5-3.

Completeness. The completeness follows by the completeness of Π_{zk}.

Quantum Proof of Knowledge. Let P^* be a malicious prover, that on input (x, ρ), can convince V to accept x with non-negligible probability ε. Before we construct a QPT extractor Ext, we first give a description of the registers used in the system.

- $\mathbf{R}_{i,j}$ for $i \in [\ell_w], j \in [\lambda]$: this consists of the receiver randomness used by the extractor in the $(i,j)^{th}$ executions of Π_{OT}.

Input of P: Instance $x \in \mathcal{L}$ along with witness w. The length of w is denoted to be ℓ_w.
Input of V: Instance $x \in \mathcal{L}$.

- For every $i \in [\ell_w]$, P samples the shares $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random conditioned on $\oplus_{j=1}^{\lambda} sh_{i,j} = w_i$, where w_i is the i^{th} bit of w.
- For every $i \in [\ell_w]$, P samples the bits $\alpha_{i,1}, \ldots, \alpha_{i,\lambda}$ uniformly at random.
- For $i \in [\ell_w], j \in [\lambda]$, do the following:
  - $P \leftrightarrow V$: P and V execute Π_{OT} with V playing the role of the receiver in Π_{OT} and P playing the role of the sender in Π_{OT}. The input of the receiver in this protocol is 0, while the input of the sender is set to be $(sh_{i,j}, \alpha_{i,j})$ if $b_{i,j} = 0$, otherwise it is set to be $(\alpha_{i,j}, sh_{i,j})$ if $b_{i,j} = 1$, where the bit $b_{i,j}$ is sampled uniformly at random.
    Call the resulting transcript of the protocol to be $\tau_{\mathsf{OT}}^{(i,j)}$ and let $r_{\mathsf{OT}}^{(i,j)}$ be the randomness used by the sender in OT.
  - $P \rightarrow V$: P sends $b_{i,j}$ to V.
- $P \leftrightarrow V$: P and V execute Π_{zk} with P playing the role of the prover of Π_{zk} and V playing the role of the verifier of Π_{zk}. The instance is $\left(x, \left\{\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\right\}_{i\in[\ell_w], j\in[\lambda]}\right)$ and the witness is $\left(w, \left\{r_{\mathsf{OT}}^{(i,j)}, sh_{i,j}, \alpha_{i,j}\right\}_{i\in[\ell_w], j\in[\lambda]}\right)$. If the verifier in Π_{zk} rejects, then V rejects.

Figure 5-3: Construction of (standalone) QZKPoK for NP.

- $\mathbf{B}_{i,j}$, for $i \in [\ell_w], j \in [\lambda]$: this is a single-qubit register that contains a bit that is used by the extractor in the $(i,j)^{th}$ execution of the OT protocol.
- **Dec**: it contains the decision register that indicates whether to rewind or not.
- **Aux**: this is initialized with the auxiliary state of the prover.
- $\mathbf{T}_{i,j}$, for $i \in [\ell_w], j \in [\lambda]$: it contains the transcripts of the $(i,j)^{th}$ executions of the OT protocol.
- $\mathbf{T}^*$: it contains the transcript of the protocol Π_{zk}.
- $\mathbf{X}$: this is a poly(λ)-qubit ancillary register.

Description of $\mathsf{Ext}(x, |\Psi\rangle)$: The state of the extractor is initialized as follows:

$$\left(\bigotimes_{i\in[\ell_w],j\in[\lambda]} |0\rangle_{\mathbf{B}_{i,j}}|0\rangle_{\mathbf{R}_{i,j}}|0\rangle_{\mathbf{T}_{i,j}}\right) \otimes |0\rangle_{\mathbf{T}^*} \otimes |\Psi\rangle_{\mathbf{Aux}} \otimes |0\rangle_{\mathbf{Dec}} \otimes |0^{\otimes \mathrm{poly}(\lambda)}\rangle_{\mathbf{X}}$$

- For all $i \in [\ell_w], j \in [\lambda]$, perform the following operations in superposition:
  - Let $|\widetilde{\Psi}\rangle$ be the state of the system at the beginning of the $(i,j)^{th}$ execution.
  - Prepare the following state[6]:

$$|0\rangle_{\mathbf{B}_{i,j}}|0\rangle_{\mathbf{R}_{i,j}} \to \frac{1}{\sqrt{2^{\lambda+1}}} \sum_{\beta_{i,j}\in\{0,1\}, s^{\mathsf{OT}}_{i,j}\in\{0,1\}^\lambda} |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s^{\mathsf{OT}}_{i,j}\rangle_{\mathbf{R}_{i,j}}$$

  - It then performs the $(i,j)^{th}$ execution of Π_{OT} along with the P^*'s message immediately after the $(i,j)^{th}$ execution of Π_{OT} in superposition. The resulting transcript is stored in the register $\mathbf{T}_{i,j}$. We denote the unitary that performs this step to be $U^{(1)}_{i,j}$.
  - After P^* sends the message immediately after the $(i,j)^{th}$ execution of Π_{OT}, apply the unitary $U^{(2)}_{i,j}$ defined as follows:

$$\begin{aligned} & U^{(2)}_{i,j}|\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s^{(i,j)}_{\mathsf{OT}}\rangle_{\mathbf{R}_{i,j}}|\tau^{(i,j)}_{\mathsf{OT}}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|0\rangle_{\mathbf{Dec}} \\ = & \begin{cases} |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s^{(i,j)}_{\mathsf{OT}}\rangle_{\mathbf{R}_{i,j}}|\tau^{(i,j)}_{\mathsf{OT}}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|\mathsf{Match}_{i,j}\rangle_{\mathbf{Dec}} \text{ if } \mathsf{acc}_{i,j}=1, \\ |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s^{(i,j)}_{\mathsf{OT}}\rangle_{\mathbf{R}_{i,j}}|\tau^{(i,j)}_{\mathsf{OT}}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|+\rangle_{\mathbf{Dec}}, \quad \text{otherwise} \end{cases} \end{aligned}$$

    Here, $\mathsf{Match}_{i,j} = 0$ if and only if $\beta_{i,j} = b_{i,j}$, where $b_{i,j}$ is the bit sent by P^* after the $(i,j)^{th}$ execution of the OT protocol. Moreover, $\mathsf{acc}_{i,j} = 1$ only if P^* has not aborted in $(i,j)^{th}$ OT execution.

    Let $W_{i,j} = \mathsf{Amplifier}\left(U^{(2)}_{i,j}U^{(1)}_{i,j}\right)$, where $\mathsf{Amplifier}$ is defined in Lemma 28. Perform $W_{i,j}|\widetilde{\Psi}\rangle$ to obtain $|\Psi_{i,j}\rangle$.

- Perform the execution of Π_{zk} in superposition.

- Measure all the registers except **Aux**. Perform the OT reconstruction on input the measured transcript $\tau^{\mathsf{OT}}_{i,j}$, measured randomness $s^{\mathsf{OT}}_{i,j}$ and receiver's bit $b_{i,j}$. Call the resulting reconstruction output to be $\widetilde{sh_{i,j}}$. Let $\widetilde{w}_i = \oplus_{j=1}^{\ell_w} \widetilde{sh_{i,j}}$. Let w be the concatenation of the bits $\widetilde{w}_1, \ldots, \widetilde{w}_{\ell_w}$. If w is not a witness for x, abort. Otherwise output the state in **Aux** along with w.

[6] We will assume, without loss of generality, that the length of the random strings used in the OT protocol be λ.

We now argue that our protocol satisfies the proof of knowledge property. We assume that there is some total ordering defined on (i, j), for $i \in [\ell_w]$ and $j \in [\lambda]$. Without loss of generality, we assume that $(1, 1)$ is the least element in this total ordering.

$\underline{\mathsf{Hybrid}_1}$: In this hybrid, P^* interacts with the honest verifier V. The verifier V accepts the proof with probability ε.

$\underline{\mathsf{Hybrid}_{2.(i,j)}}$, for $i \in [\ell_w], j \in [\lambda]$: We define a hybrid verifier $\mathsf{Hybrid}.V_{i,j}$ as follows. Let $|\Phi\rangle$ be the initial state of the system. Compute $|\Psi_{i,j}\rangle = \prod_{(i',j')\leq(i,j)} W_{i',j'}|\Phi\rangle$. From here on, the rest of the iterations of Π_{OT} are computed by interacting with P^* interacting honestly as specified in the real execution. The protocol Π_{zk} is computed by interacting with P^* honestly as in the real execution. Finally, $\mathsf{Hybrid}.V_{i,j}$ outputs its decision.

The probability that $\mathsf{Hybrid}.V_{i,j}$ accepts is negligibly close to ε. Moreover, from the statistical security against senders, it follows that the state output by P^* in this hybrid is close in trace distance to the state output by P^* in the previous hybrid. We omit the proof since it essentially follows the same line of argument used in Section 4.3.3.

$\underline{\mathsf{Hybrid}_3}$: We define a hybrid verifier $\mathsf{Hybrid}.V_3$ as follows. Let $|\Phi\rangle$ be the initial state of the system. Compute $|\Psi_{i,j}\rangle = \prod_{(i\in[\ell_w],j\in[\lambda])} W_{i,j}|\Phi\rangle$. The protocol Π_{zk} is computed by interacting with P^* honestly as in the real execution. Finally, $\mathsf{Hybrid}.V_3$ outputs its decision.

The probability that $\mathsf{Hybrid}.V_3$ accepts is negligibly close to ε. This follows from the fact that $\mathsf{Hybrid}.V_3$ is identical to $\mathsf{Hybrid}.V_{i^*,j^*}$, where (i^*, j^*) is the highest element in the total ordering.

Moreover, the state output by P^* in this hybrid is the same as the state output by P^* in the previous hybrid.

$\underline{\mathsf{Hybrid}_4}$: Define a hybrid verifier $\mathsf{Hybrid}.V_4$ as follows: it executes the hybrid verifier $\mathsf{Hybrid}.V_3$ until the step just before it outputs its decision. Let $\widetilde{sh_{i,j}}$ be the share output by the reconstruction algorithm of the receiver of Π_{OT}. Let $\widetilde{w}_i = \oplus_{j=1}^{\ell_w} \widetilde{sh_{i,j}}$. Let w be the concatenation of the bits $\widetilde{w}_1, \ldots, \widetilde{w}_{\ell_w}$. If w is not a witness for x, abort. Otherwise, output the decision of $\mathsf{Hybrid}.V_3$.

The probability that $\mathsf{Hybrid}.V_4$ accepts is negligibly close to ε. To see this, note that it is sufficient to argue that $|p_3 - p_4| \leq \mathrm{negl}(\lambda)$, where p_3 is the probability with which $\mathsf{Hybrid}.V_3$ aborts and p_4 is the probability with which $\mathsf{Hybrid}.V_4$ aborts. This fact follows from the soundness of Π_{zk}. Moreover, the output state of the prover in Hybrid_3 is the same as the output state of the prover in Hybrid_4.

Note that the probability that the extractor Ext outputs a valid witness w is the same as the probability that the hybrid verifier $\mathsf{Hybrid}.V_4$ accepts. Moreover, the state output by P^* when interacting with Ext is exactly the same as the state output

by P^* in Hybrid_4.

(Standalone) Quantum Zero-Knowledge

Suppose $(x, w) \in \mathcal{R}(\mathcal{L})$. Suppose V^* is a QPT verifier, that on input $(x, |\Psi\rangle)$, interacts with the honest prover $P(x, w)$. We construct a simulator Sim that takes as input $(x, |\Psi\rangle)$ such that the output distribution of the simulator is computationally indistinguishable from the output distribution of V^*.

Description of $\mathsf{Sim}(x, |\Psi\rangle)$:

- For every $i \in [\ell_w]$, Sim samples $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random.
- For $i \in [\ell_w], j \in [\lambda]$, do the following:
  - Sim and V^* execute Π_{OT}. The verifier V^* takes the role of the receiver of Π_{OT} and Sim takes the role of the sender. The input of the sender is $(sh_{i,j}, sh_{i,j})$.
  - Sim samples a random bit $b_{i,j}$ and sends to V^*.
- Let the state of the verifier, at this point of this protocol, be $\widetilde{|\Psi\rangle}$. Let $\mathsf{Sim}_{\mathsf{zk}}$ be the Π_{zk} simulator associated with the Π_{zk} verifier $\widetilde{V^*}$, where $\widetilde{V^*}$ is the code used by V^* in the execution of the protocol Π_{zk}. Compute $\mathsf{Sim}_{\mathsf{zk}}$ on input the state $\widetilde{|\Psi\rangle}$ and the instance $\left(x, \left\{\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\right\}\right)$.
- Output the transcript of the protocol along with the private state of the verifier V^*.

We now prove that the state output by V^* when interacting with the honest prover $P(x, w)$ is computationally indistinguishable from the state output by $\mathsf{Sim}(x, |\Psi\rangle)$. Consider the following hybrids. As before we consider a total ordering on (i, j), for $i \in [\ell_w], j \in [\lambda]$.

Hybrid_1: In this hybrid, P and V^* interact with each other. The output of this hybrid is the output of V^*.

Hybrid_2: We define another hybrid prover $\mathsf{Hybrid}_2.P$ that behaves as follows: it simulates the protocol Π_{zk} using the simulator $\mathsf{Sim}_{\mathsf{zk}}$ as given in the description of Sim. The rest of the protocol is the same as in the hybrid Hybrid_1.

The computational indistinguishability of Hybrid_1 and Hybrid_2 follows from the quantum zero-knowledge property of Π_{zk}.

Hybrid_3: The output of this hybrid is the output of $\mathsf{Sim}(x, |\Psi\rangle)$.

Claim 101. *Assuming the post-quantum sender-privacy of Π_{OT}, the output of Hybrid_2 is computationally indistinguishable from the output of Hybrid_3.*

Proof. Let $\mathcal{A}$ be the distinguisher distinguishing Hybrid_2 and Hybrid_3. We are going to prove that $\mathcal{A}$ can only distinguish with negligible probability. Consider the intermediate hybrids.

$\underline{\mathsf{Hybrid}_{2.1}}$: This is identical to Hybrid_2.

We now consider a series of hybrids that are defined with respect to $\mathcal{A}$.

$\underline{\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(i^*,j^*)} \text{ for } i^* \in [\ell_w], j^* \in [\lambda-1]}$: We say that a hybrid prover $\mathsf{Hybrid}_{2.2.(i^*,j^*)}.P$, uses $\left(\left\{\widehat{b}_{i,j}\right\}_{(i,j)\leq(i^*,j^*)}\right)$, if the following holds: it executes the prover as in $\mathsf{Hybrid}_{2.1}$, except, for $(i,j) \leq (i^*,j^*)$, it uses the input $(sh_{i,j}, sh_{i,j})$ if $\widehat{b}_{i,j} \neq b_{i,j}$ or uses the input $(\alpha_{i,j}, \alpha_{i,j})$ if $\widehat{b}_{i,j} = b_{i,j}$.

Now, execute the above hybrid prover $\mathsf{Hybrid}_{2.2.(i^*,j^*)}.P$ by adaptively choosing $\left(\left\{\widehat{b}_{i,j}\right\}_{(i,j)\leq(i^*,j^*)}\right)$ (as a function of the current state of the verifier and $\mathcal{A}$) such that the output distributions of $\mathsf{Hybrid}_{2.2.(i^*,j^*)}.P$ and $\mathsf{Hybrid}_{2.2.(i^*,j^*)-1}.P$ cannot be distinguished by $\mathcal{A}$. If such a set of bits cannot be adaptively chosen then abort. Otherwise, this hybrid prover interacts with the verifier and the output of this hybrid is set to be the output of the verifier.

Claim 102. *The hybrid* $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(i^*,j^*)}$ *aborts with negligible probability.*

Proof. We prove this by induction.

Base Case: $(i^*,j^*) = (1,1)$**.** We prove that $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(1.1)}$ aborts with negligible probability. From the post-quantum sender privacy property of Π_{OT} (Definition 96), it follows that upon fixing the view of the verifier until the last message of execution of $(1,1)^{th}$ OT protocol, there exists a bit $\widehat{b}$, with probability negligibly close to 1, such that the adversary cannot win the Game $G_{\widehat{b}}\left(m_0^{(1,1)}, m_1^{(1,1)}\right)$ (specified in Definition 96) where, $\left(m_0^{(1,1)}, m_1^{(1,1)}\right) = (sh_{1,1}, \alpha_{1,1})$ is $b_{1,1} = 0$ and $\left(m_0^{(1,1)}, m_1^{(1,1)}\right) = (\alpha_{1,1}, sh_{1,1})$ is $b_{1,1} = 1$.

Induction Hypothesis. Suppose this statement is true for all $(i,j) < (i^*,j^*)$. We prove this statement to be true even for (i^*,j^*) using proof by contradiction.

Suppose $\mathsf{Hybrid}_{2.2.(i^*,j^*)}$ aborts with non-negligible probability then we design a QPT adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$, that receives as input non-uniform quantum advice, and breaks the post-quantum sender privacy property of Π_{OT}.

We first define the non-uniform advice as follows: it computes the interaction between the hybrid prover $\mathsf{Hybrid}_{2.2.(i^*,j^*)-1}.P$ and the verifier V^*, until the $((i^*,j^*)-1)^{th}$ execution of OT. It outputs the private state of $\mathsf{Hybrid}_{2.2.(i^*,j^*)-1}.P$ and the private state of V^*. Call this state $|\Psi_{adv}\rangle$.

$\mathcal{B}_1$, upon receiving $|\Psi_{adv}\rangle$, takes the role of the receiver and interacts with the external challenger until the receiver's last message of the $(i^*, j^*)^{th}$ execution of Π_{OT}. $\mathcal{B}_1$ uses the code of V^* to interact with the external challenger. The external challenger on the other receives as input $m_0^{(i^*,j^*)} = sh_{i^*,j^*}$ and $m_1^{(i^*,j^*)} = \alpha_{i^*,j^*}$ if $b_{i^*,j^*} = 0$ or $m_1^{(i^*,j^*)} = sh_{i^*,j^*}$ and $m_0^{(i^*,j^*)} = \alpha_{i^*,j^*}$ if $b_{i^*,j^*} = 1$, from $\mathcal{B}_1$, where $sh_{i^*,j^*}, \alpha_{i^*,j^*}, b_{i^*,j^*}$ are generated as in $\mathsf{Hybrid}_{2.1}$. It then outputs the state $|\Psi_1\rangle$ of V^* obtained after the receiver sends the last message in the $(i^*, j^*)^{th}$ execution of Π_{OT}.

$\mathcal{B}_2$, upon receiving $|\Psi_1\rangle$, computes the rest of the executions of Π_{OT} and Π_{zk} by emulating the interaction between the hybrid prover $\mathsf{Hybrid}_{2.2.(i^*,j^*)}.P$ and the verifier V^*. It then inputs the final state of V^* to $\mathcal{A}$. The output of $\mathcal{B}_2$ is set to be the output of $\mathcal{A}$.

Our initial assumption was that the $\mathsf{Hybrid}_{2.2.(i^*,j^*)}$ aborts with non-negligible probability. This means that the adversary $\mathcal{A}$ can distinguish with non-negligible probability (over the view of the verifier until the $(i^*, j^*)^{th}$ OT execution) both the games – that is, distinguishing $(m_0^{(i^*,j^*)}, m_1^{(i^*,j^*)})$ from $(m_1^{(i^*,j^*)}, m_1^{(i^*,j^*)})$ (Game 0) as well as distinguishing $(m_0^{(i^*,j^*)}, m_1^{(i^*,j^*)})$ from $(m_0^{(i^*,j^*)}, m_0^{(i^*,j^*)})$ (Game 1) – with probability significantly greater than $\frac{1}{2}$. This in turn means that $\mathcal{B}$ can break the post-quantum sender privacy property of Π_{OT} with non-negligible probability. Thus, we arrived at a contradiction. □

$\underline{\mathsf{Hybrid}^{\mathcal{A}}_{2.3}}$: This hybrid is the same as $\mathsf{Hybrid}_{2.2.(\ell_w,\lambda-1)}$, except that the hybrid prover will abort if for there is $i \in [\ell_w]$ such that for all $j \in [\lambda - 1]$, it holds that $b_{i,j} \neq \widehat{b_{i,j}}$.

Claim 103. *The hybrids* $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ *and* $\mathsf{Hybrid}^{\mathcal{A}}_{2.3}$ *can be distinguished by* $\mathcal{A}$ *with only negligible probability.*

Proof. To prove this, we consider an alternate hybrid prover in $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ which samples, for any i, $b_{i,j} \xleftarrow{\$} \{0,1\}$ at the end of first $(\lambda - 1)$ iterations of Π_{OT}. It then sets the input to the λ^{th} iteration of Π_{OT} to be $\left(\oplus_{j=1}^{\lambda-1} m_{b_{i,j}}^{(i,j)}, u\right)$ with probability $\frac{1}{2}$ or $\left(u, \oplus_{j=1}^{\lambda-1} m_{b_{i,j}}^{(i,j)}\right)$ with probability $\frac{1}{2}$, where $u \xleftarrow{\$} \{0,1\}$ and $\{m_0^{(i,j)}, m_1^{(i,j)}\}_{j\in[\lambda-1]}$ are the inputs used in the first $\lambda - 1$ iterations of Π_{OT}. Note that the output distribution of $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ remains the same even with this change.

Since the $b_{i,j}$'s, for $j \leq \lambda - 1$, are sampled after the $\widehat{b}_{i,j}$'s are decided, the probability that $\widehat{b}_{i,j} \neq b_{i,j}$ is $\frac{1}{2}$ for any $i \in [\ell_w], j \in [\lambda - 1]$. Thus, the probability that $\left(\exists i \in [\ell_w], j \in [\lambda - 1], b_{i,j} \neq \widehat{b}_{i,j}\right)$ is $\leq \frac{\ell_w}{2^{\lambda-1}}$. Conditioned on this bad event, the output distributions of $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ and $\mathsf{Hybrid}_{2.3}$ are identical. Thus, $\mathcal{A}$ cannot distinguish the hybrids $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ and $\mathsf{Hybrid}^{\mathcal{A}}_{2.3}$.

□

$\underline{\mathsf{Hybrid}^{\mathcal{A}}_{2.4.i^*} \text{ for all } i \in [\ell_w]}$: This hybrid is the same as $\mathsf{Hybrid}^{\mathcal{A}}_{2.3}$ except that the hy-

brid prover $\mathsf{Hybrid}_{2.4.i^*}.P$ is additionally parameterized by $\left(\left\{\widehat{b}_{i,\lambda}\right\}_{i\leq i^*}\right)$. The only change from the previous hybrid is that the hybrid prover, for $i \leq i^*$, use the input $(sh_{i,\lambda}, sh_{i,\lambda})$ if $\widehat{b}_{i,\lambda} \neq b_{i,\lambda}$ or use $(\alpha_{i,\lambda}, \alpha_{i,\lambda})$ if $\widehat{b}_{i,\lambda} = b_{i,\lambda}$.

Now, consider a hybrid prover $\mathsf{Hybrid}_{2.4.i^*}.P$, parameterized by $\left(\left\{\widehat{b}_{i,j}\right\}_{(i\leq i^*)\vee(j\leq\lambda-1)}\right)$, where $\left(\left\{\widehat{b}_{i,j}\right\}_{(i,j)\leq(i^*,j^*)}\right)$, is defined to be such that the output distributions of $\mathsf{Hybrid}_{2.i^*}.P$ and $\mathsf{Hybrid}_{2.4.i^*-1}.P$ cannot be distinguished by $\mathcal{A}$. If such a hybrid prover does not exist, then abort. Otherwise, this hybrid prover interacts with the verifier and the output of this hybrid is set to be the output of the verifier.

Claim 104. *The hybrid* $\mathsf{Hybrid}_{2.4.i^*}$ *aborts with negligible probability.*

We omit the proof of the above claim since it uses the same inductive argument as the proof of Claim 102.

$\underline{\mathsf{Hybrid}_{2.5}}$: This hybrid is the same as Hybrid_3, i.e. the output of the simulator.

Conditioned on $\mathsf{Hybrid}_{2.4.\ell_w}$ not aborting, the output distributions of $\mathsf{Hybrid}_{2.4.\ell_w}$ and $\mathsf{Hybrid}_{2.5}$ are the same. This follows from the fact that if $\mathsf{Hybrid}_{2.4.\ell_w}$ does not abort then the distribution of the inputs used in all the OT executions in the hybrids $\mathsf{Hybrid}_{2.4.\ell_w}$ and $\mathsf{Hybrid}_{2.5}$ are the same. Thus, $\mathcal{A}$ can distinguish $\mathsf{Hybrid}_{2.4.\ell_w}$ and $\mathsf{Hybrid}_{2.5}$ only with negligible probability.

From the above hybrids, it follows that $\mathcal{A}$ can distinguish the hybrids $\mathsf{Hybrid}_{2.1}$ and $\mathsf{Hybrid}_{2.5}$ with only negligible probability.

□

5.3.3 Extending to Bounded Concurrent QZK Setting

We show how to adopt the construction in Section 5.3.2 to the bounded concurrent setting.

Construction

The construction of bounded concurrent quantum proof of knowledge system is the same as Figure 5-3, except that we instantiate Π_{zk} with a modified version of the bounded concurrent QZK for NP construction in Section 4.3.

Modified Bounded Concurrent QZK for NP Construction. We modify the construction in Section 4.3 as follows: Let M be the round complexity of the statistical receiver private OT protocol and let $M = \lambda^c$ for some constant c, where λ is the security parameter used in the OT protocol. Let λ' denote a different security parameter used in Π_{zk} such that $\lambda' - M \geq \lambda$. We set the threshold of matched slots needed in the WI protocol from Section 4.3, to instead be, $60Q^7\lambda' + Q^4\lambda' - M$, provided we set $\lambda' \gg M$.

The completeness and soundness proofs of this modified construction are the same as the ones in Section 4.3. Even the quantum zero-knowledge property is the same as before. However, we will need the simulator to satisfy a stronger property defined next.

Strong QZK Simulator. We explain the differences between the strong QZK simulator and the simulator Sim defined in Section 4.3. The strong simulator proceeds as follows:

1. It simulates block-by-block similarly to Sim, but instead of using $|+\rangle_{\mathbf{Dec}}$ only in the decision bit of the registers that aborted, it can choose to use $|+\rangle_{\mathbf{Dec}}$ on other transcripts as well. This decision is based on an efficiently computable function f. For example, on a transcript t, it can set $\mathbf{Dec}$ to $|+\rangle_{\mathbf{Dec}}$ conditioned on $f(t) = 1$.

2. After rewinding a block, it measures the transcript of that block instead of waiting until the end to measure. Furthermore, it keeps tracks of the total number of blocks on which the measurement outcomes correspond to a transcript in which it used $|+\rangle_{\mathbf{Dec}}$.

3. If at any point, the number of block measurement outcomes that correspond to $|+\rangle_{\mathbf{Dec}}$ transcripts is greater than M, it aborts.

Conditioned on the strong simulator not aborting in Step 3, its output is computationally indistinguishable from the output of Sim.

Arguing Bounded Concurrent Quantum Zero-Knowledge for Figure 5-3. In the proof of bounded concurrent quantum zero-knowledge, we now need to handle Q-session verifiers, where Q is the number of sessions associated with the protocol.

The description of the simulator is the same as in the description of the simulator in Section 5.3.2 except that we now execute the bounded concurrent strong simulator described above for Π_{zk} instead of the standalone ZK simulator.

We describe the hybrids below. Our description of hybrids and the proofs of indistinguishability between the hybrids closely follows the structure of the proof in Section 5.3.2 and hence we only highlight the main differences.

$\underline{\mathsf{Hybrid}_1}$: Same as Hybrid_1 in Section 5.3.2.

$\underline{\mathsf{Hybrid}_2}$: We define another hybrid prover $\mathsf{Hybrid}_2.P$ that behaves as follows: it simulates the protocol Π_{zk} using the bounded concurrent simulator $\mathsf{Sim}_{\mathsf{zk}}$ as given in the description of Sim. The rest of the protocol is the same as in the hybrid Hybrid_1.

The computational indistinguishability of Hybrid_1 and Hybrid_2 follows from the bounded concurrent quantum zero-knowledge property of Π_{zk}.

$\underline{\mathsf{Hybrid}_3}$: The output of this hybrid is the output of the simulator.

Claim 105. *Assuming the post-quantum sender-privacy of* Π_{OT}*, the output of* Hybrid_2 *is computationally indistinguishable from the output of* Hybrid_3*.*

Proof. Let $\mathcal{A}$ be the distinguisher distinguishing Hybrid_2 and Hybrid_3. We are going to prove that $\mathcal{A}$ can only distinguish with negligible probability. Consider the intermediate hybrids.

$\underline{\mathsf{Hybrid}_{2.1}}$: This is identical to Hybrid_2.

We now consider a sequence of hybrids. Each hybrid in this sequence is parameterized by the number of OT executions and the number of sessions. We first replace the inputs of all the OTs corresponding to one session before we move on to the next session. That is, each hybrid is of the form $\mathsf{Hybrid}_{2.2.i.j.k}$. We first start with $i = 1, j = 1, k = 1$. We then iterate over $j = 1, \ldots, \lambda - 1$, and then we increment i. We keep doing this, until we reach the hybrid $\mathsf{Hybrid}_{2.2.\ell_w.\lambda-1.k}$. The next hybrid is $\mathsf{Hybrid}_{2.3.k}$. After this, we have the hybrid, $\mathsf{Hybrid}_{2.4.i.k}$, where $i = 1$ and $k = 1$. We then iterate over $i = 1, \ldots, \ell_w$. Immediately after the hybrid $\mathsf{Hybrid}_{2.4.\ell.k}$, we have the hybrid $\mathsf{Hybrid}_{2.5.k}$. At this point, all the OTs corresponding to the first session have been replaced. Immediately after this hybrid, we then move on to the hybrid $\mathsf{Hybrid}_{2.2.i.j.k}$, where $i = 1, j = 1, k = 2$. We then continue as above, until we reach the hybrid $\mathsf{Hybrid}_{2.5.Q}$. The hybrid that follows $\mathsf{Hybrid}_{2.5.Q}$ is the hybrid $\mathsf{Hybrid}_{2.6}$.

$\underline{\mathsf{Hybrid}^{\mathcal{A}}_{2.2.i.j.k} \text{ for } i \in [\ell_w], j \in [\lambda - 1], k \in [Q]}$: We say that a prover, uses $\left(\left\{\widehat{b}_j\right\}_{j \leq i}\right)$ in a particular transcript, if the following holds: in superposition, execute the prover as in $\mathsf{Hybrid}_{2.1}$, except that in the first $j \leq i$ OT executions to end in the transcript being generated, it uses the input (sh_j, sh_j) if $\widehat{b}_j \neq b_j$ or uses the input (α_j, α_j) if $\widehat{b}_j = b_j$.

We define a hybrid prover $\mathsf{Hybrid}_{2.2.i.j.k}.P$ as follows:

- For $k' < k$, it chooses the input to the $(i,j)^{th}$ execution of the k'-session to be $(\alpha_{i,j}, \alpha_{i,j})$, where $\alpha_{i,j}$ is sampled uniformly at random.
- For $k' > k$, it chooses the inputs to the OT executions as done by the prover in $\mathsf{Hybrid}_{2.1}$.
- For $k' = k$, the hybrid prover, *in superposition*, adaptively uses $\left(\left\{\widehat{b}_{(i',j')}\right\}_{(i',j') \leq (i,j)}\right)$ such that the output distributions of $\mathsf{Hybrid}_{2.2.(i,j).k}.P$ and $\mathsf{Hybrid}_{2.2.(i,j)-1.k}.P$ (if $(i,j) = (1,1)$ then the hybrid $\mathsf{Hybrid}_{2.2.(i,j).k}.P$ is defined to be $\mathsf{Hybrid}_{2.4.k-1}.P$) cannot be distinguished by $\mathcal{A}$. That is, since the whole protocol is being executed in superposition, as a function of each term in the superposition, the bits $\left(\left\{\widehat{b}_{(i',j')}\right\}_{(i',j') \leq (i,j)}\right)$ are adaptively determined and stored in a separate register to be used by the hybrid prover. If the entire sequence of bits cannot be determined, then store $\perp$ in the same register. At the end of the protocol,

we measure this register. If the outcome is $\perp$ then abort, otherwise, measure the registers storing the transcript, trace out all the registers except the register containing the auxiliary state of the verifier and output the measured transcript along with the residual auxiliary state of the verifier.

Claim 106. *The hybrid* $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.i.j.k}$ *aborts with negligible probability.*

Proof. We prove this by induction.

Base Case: $(i,j) = (1,1)$**.** We prove that $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.1.1.k}$ aborts with negligible probability. Suppose not. We demonstrate a reduction that breaks the sender privacy property of OT with non-negligible probability. The goal of the reduction is to win in both the games with non-negligible probability: in the first game, it needs to distinguish the case when the challenger uses the input (m_0, m_1), where $(m_0, m_1) = (sh_{1,1}, \alpha_{1,1})$ with probability $\frac{1}{2}$ and $(m_0, m_1) = (\alpha_{1,1}, sh_{1,1})$ or when it uses the input $(sh_{1,1}, sh_{1,1})$. In the second game, it needs to distinguish the case when the challenger uses the input (m_0, m_1), where (m_0, m_1) is defined as above, versus the case when it uses the input $(\alpha_{1,1}, \alpha_{1,1})$.

We describe a reduction that does the following: just like the simulator of the bounded concurrent QZK, it divides the entire protocol transcript into blocks $B_1, \ldots, B_L$, where L is as defined in Π_{zk}. For every block B_i, it does the following: it executes the simulator in superposition. If it encounters a message of $(1,1)^{th}$ OT, it stops simulating the rest of the block. It then puts $|+\rangle$ state in the decision register. Otherwise, it continues the simulation until the end of the block. It then performs Watrous rewinding. At the end, it measures the trascript. There are two cases:

- If the block B_i has completed it's execution and if no message in the $(1,1)^{th}$ OT execution has been encountered so far, then continue to the block B_{i+1}.
- If a message in the $(1,1)^{th}$ OT execution has been encountered then forward to the challenger of the OT protocol. Use the response by the challenger to continue the execution of B_i, albeit by interacting V^*, rather running V^* in superposition. Once this is completed, move on to the block B_{i+1}.

Finally, after all the blocks are executed, the transcript along with the final private state of the verifier is input to $\mathcal{A}$. Note that there will be at most M (the round complexity of the statistical receiver private OT protocol) blocks where the simulator's decision to rewind gets changed to $|+\rangle$ instead of using the rewinding decision that it was using previously. This is why we change the parameters of the zero-knowledge simulator to make sure that there are still enough blocks being appropriately rewound as needed for the simulation to execute correctly.

If the challenger uses the input (m_0, m_1) then it corresponds to the hybrid $\mathsf{Hybrid}_{2.4.k-1}$ (if $k = 1$, then $\mathsf{Hybrid}_{2.4.k-1}$ is the hybrid $\mathsf{Hybrid}_{2.1}$) and if the challenger uses the input $(sh_{1,1}, sh_{1,1})$ or the input $(\alpha_{1,1}, \alpha_{1,1})$ then it corresponds to the hybrid $\mathsf{Hybrid}_{2.2.1.1.k}$.

If $\mathcal{A}$ can distinguish the hybrids $\mathsf{Hybrid}_{2.2.1.1.k}$ and $\mathsf{Hybrid}_{2.4.k-1}$ with non-negligible probability then the reduction can break the sender privacy property with non-negligible probability.

Induction Hypothesis. Suppose this statement is true for all $(i', j') < (i, j)$. We then show this to be true even for (i, j).

Suppose this is not true. We then design a reduction that violates the sender privacy of OT with non-negligible probability. The reduction essentially is defined along the same lines as the base case, except that the first $((i, j) - 1)$ OT executions of the k^{th} verifier are generated as non-uniform advice. That is, the advice generation algorithm executes the protocol in superposition and in each term of the superposition, it halts after the final $((i, j) - 1)^{th}$ execution of the k^{th}. Until this point, the inputs to the $(i', j')^{th}$ OT execution, for $(i', j') < (i, j)$, is set to be either $(sh_{(i',j')}, sh_{(i',j')})$ or $(\alpha_{(i',j')}, \alpha_{(i',j')})$, depending on the distinguishing probability of $\mathcal{A}$. Finally, the advice generation measures the transcript and outputs the transcript along with the residual state.

The reduction then performs block-by-block execution of the protocol and interacts with the challenger as in the base case. The final state of the verifier along with the transcript of the entire protocol is input to $\mathcal{A}$.

As in the base case, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible probability then the reduction can also violate the sender privacy property with non-negligible probability.

□

$\underline{\mathsf{Hybrid}^{\mathcal{A}}_{2.3.k} \text{ for } k \in [Q]}$: This hybrid is the same as $\mathsf{Hybrid}_{2.2.\ell_w,\lambda-1.k}$, except that the hybrid prover will abort if there is $i \in [\ell_w]$ such that for all $j \in [\lambda - 1]$, it holds that $b_{i,j} \neq \widehat{b_{i,j}}$.

Claim 107. *The hybrids* $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.\ell_w,\lambda-1.k}$ *and* $\mathsf{Hybrid}^{\mathcal{A}}_{2.3.k}$ *can be distinguished by* $\mathcal{A}$ *with only negligible probability.*

Proof. To prove this, we consider an alternate hybrid prover in $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.\ell_w,\lambda-1.k}$ which samples, for any i, $b_{i,j} \xleftarrow{\$} \{0, 1\}$ at the end of first $(\lambda - 1)$ iterations of Π_{OT}. It then sets the input to the λ^{th} iteration of Π_{OT} to be $\left(\oplus_{j=1}^{\lambda-1} m^{(i,j)}_{b_{i,j}}, u\right)$ with probability $\frac{1}{2}$ or $\left(u, \oplus_{j=1}^{\lambda-1} m^{(i,j)}_{b_{i,j}}\right)$ with probability $\frac{1}{2}$, where $u \xleftarrow{\$} \{0, 1\}$ and $\{m^{(i,j)}_0, m^{(i,j)}_1\}_{j\in[\lambda-1]}$ are the inputs used in the first $\lambda - 1$ iterations of Π_{OT}. Note that the output distribution of $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ remains the same even with this change.

Since the $b_{i,j}$'s, for $j \leq \lambda - 1$, are sampled after the $\widehat{b}_{i,j}$'s are decided, the probability that $\widehat{b}_{i,j} \neq b_{i,j}$ is $\frac{1}{2}$ for any $i \in [\ell_w], j \in [\lambda - 1]$. Thus, the probability that $\left(\exists i \in [\ell_w], j \in [\lambda - 1], b_{i,j} \neq \widehat{b}_{i,j}\right)$ is $\leq \frac{\ell_w}{2^{\lambda-1}}$. Conditioned on this bad event, the output distributions of $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.\ell_w.\lambda-1.k}$ and $\mathsf{Hybrid}^{\mathcal{A}}_{2.3.k}$ are identical. Thus, $\mathcal{A}$ cannot distinguish the hybrids $\mathsf{Hybrid}^{\mathcal{A}}_{2.2.\ell_w.\lambda-1.k}$ and $\mathsf{Hybrid}^{\mathcal{A}}_{2.3.k}$.

□

$\underline{\mathsf{Hybrid}^{\mathcal{A}}_{2.4.i^*.k} \text{ for all } i^* \in [\ell_w], k \in [Q]}$: This hybrid is the same as $\mathsf{Hybrid}^{\mathcal{A}}_{2.3.k}$ except that the hybrid prover $\mathsf{Hybrid}_{2.4.i^*.k}.P$ is additionally parameterized by $\left(\left\{\widehat{b}_{i,\lambda}\right\}_{i\leq i^*}\right)$. The

only change from the previous hybrid is that the hybrid prover, for $i \leq i^*$, use the input $(sh_{i,\lambda}, sh_{i,\lambda})$ if $\widehat{b}_{i,\lambda} \neq b_{i,\lambda}$ or use $(\alpha_{i,\lambda}, \alpha_{i,\lambda})$ if $\widehat{b}_{i,\lambda} = b_{i,\lambda}$.

Now, consider a hybrid prover $\mathsf{Hybrid}_{2.4.i^*.k}.P$, parameterized by $\left(\left\{\widehat{b}_{i,j}\right\}_{(i\leq i^*)\vee(j\leq\lambda-1)}\right)$, where $\left(\left\{\widehat{b}_{i,j}\right\}_{(i,j)\leq(i^*,j^*)}\right)$, is defined to be such that the output distributions of $\mathsf{Hybrid}_{2.4.i^*.k}.P$ and $\mathsf{Hybrid}_{2.4.i^*-1.k}.P$ cannot be distinguished by $\mathcal{A}$. If such a hybrid prover does not exist, then abort. Otherwise, this hybrid prover interacts with the verifier and the output of this hybrid is set to be the output of the verifier.

Claim 108. *The hybrid* $\mathsf{Hybrid}_{2.4.i^*.k}$ *aborts with negligible probability.*

We omit the proof of the above claim since it uses the same inductive argument as the proof of Claim 106.

$\underline{\mathsf{Hybrid}_{2.5.k}}$ for $k \in [Q]$: We define a hybrid prover that does the following:

- For $k' \leq k$, it chooses the input to the $(i,j)^{th}$ execution to be $(\alpha_{i,j}, \alpha_{i,j})$, where $\alpha_{i,j}$ is sampled uniformly at random.
- For $k' > k$, it chooses the inputs to the OT executions as done by the prover in $\mathsf{Hybrid}_{2.1}$.

Conditioned on $\mathsf{Hybrid}_{2.4.\ell_w.k}$ not aborting, the output distributions of $\mathsf{Hybrid}_{2.4.\ell_w.k}$ and $\mathsf{Hybrid}_{2.5.k}$ are the same. This follows from the fact that if $\mathsf{Hybrid}_{2.4.\ell_w}$ does not abort then the distribution of the inputs used in all the OT executions in the hybrids $\mathsf{Hybrid}_{2.4.\ell_w.k}$ and $\mathsf{Hybrid}_{2.5.k}$ are the same. Thus, $\mathcal{A}$ can disitnguish $\mathsf{Hybrid}_{2.4.\ell_w.k}$ and $\mathsf{Hybrid}_{2.5.k}$ only with negligible probability.

$\underline{\mathsf{Hybrid}_{2.6}}$: This hybrid is the same as Hybrid_3, i.e. the output of the simulator.

The output distributions of $\mathsf{Hybrid}_{2.5.Q}$ and $\mathsf{Hybrid}_{2.6}$ are identical.

From the above hybrids, it follows that $\mathcal{A}$ can distinguish the hybrids $\mathsf{Hybrid}_{2.1}$ and $\mathsf{Hybrid}_{2.6}$ with only negligible probability.

□

5.4 On proofs of quantum knowledge

We can define an anologous notion of proof of knowledge in the context of interactive protocols for QMA. This notion is called proof of *quantum* knowledge. See [CVZ20] for a definition of this notion. Coladangelo, Vidick and Zhang [CVZ20] show how to achieve quantum proof of quantum knowledge generically using quantum proof of classical knowledge. Their protocol builds upon [BJSW16] to achieve their goal. We can adopt their idea to achieve proof of quantum knowledge property for a bounded

concurrent QZK for QMA system. In Figure 4.4.3, include a quantum proof of classical knowledge system for NP (for instance, the one we constructed in Section 5.3.2) just after the prover sends encoding of the witness state $|\Psi\rangle$, encoded using the key s. Using the quantum proof of classical knowledge system, the prover convinces the verifier of its knowledge of the s. The rest of the protocol is the same as Figure 4.4.3. To see why this satisfies proof of quantum knowledge, note that an extractor can extract s with probability negligibly close to the acceptance probability and using s, can recover the witness $|\Psi\rangle$.

For the first time, we get proof of quantum knowledge (even in the standalone setting) with $(1 - \text{negl})$-quality if the acceptance probability is negligibly close to 1, where the quality denotes the closeness to the witness state. Previous proof of quantum knowledge [BG20, CVZ20] achieved only $1 - \frac{1}{\text{poly}}$ qualitiy; this is because these works use Unruh's quantum proof of classical knowledge technique [Unr12] and the extraction probability in Unruh is not negligibly close to the acceptance probability.

Chapter 6

Impossibility of Quantum Copy-Protection

A circuit C for which there would be no copy-protection would be one where, given any QPT algorithm U_C and an auxiliary state ρ_C, it is possible to recover a classical description of C. Note that this wouldn't mean that C is a quantum learnable circuit, because having access to (U_C, ρ_C) is qualitatively different than having quantum oracle access to C. This is similar to the black-box and non-black-box distinction in the context of QZK protocols. In particular, we could hope to have a quantum unlearnable circuit C and to use non-black-box techniques that would allow us to recover secrets from (U_C, ρ_C). We introduce the notion of *de-quantumizable* circuits to capture this scenario. These are quantum unlearnable circuits where access to (U_C, ρ_C) is enough to recover a classical functionally equivalent circuit to C.

It is not clear *a priori* whether such circuits exist or not. The goal of recovering a classical description from (U_C, ρ_C) is similar to the goal of extracting secret information from a QPT verifier V given non-black-box access to V. Thus, it is reasonable to assume that extraction mechanism designed in the context of QZK can help us construct de-quantumizable circuits. Our main result is to show that this is indeed the case – we combine lockable obfuscation and QFHE, as done in Section 3.3, to construct a de-quantumizable circuit class.

Before defining de-quantumizable circuits, we start by recalling what we mean by an efficient quantum implementation of a circuit C.

Definition 109 (Quantum Implementation)**.** *We say that a collection of QPT algorithms, $\{U_C, \rho_C\}_{C\in\mathcal{C}}$, computes the circuit class $\mathcal{C}$ if for any $C \in \mathcal{C}$, with input length n and output length m, ρ_C is a* $\mathrm{poly}(n)$*-qubits auxiliary state, and U_C a QPT algorithm satisfying that for all $x \in \{0,1\}^n$,*

$$\mathsf{Pr}[U_C(\rho_C, x) = C(x)] \geq 1 - negl(\lambda),$$

where the probability is over the measurement outcomes of U_C. We also refer to (U_C, ρ_C) as an **efficient quantum implementation** *of C.*

In other words, an efficient quantum implementation of a circuit C is a pair, (U_C, ρ_C), of a QPT algorithm U_C and a quantum state ρ_C, that lets you evaluate C.

6.1 De-quantumizable Circuits

A de-quantumizable class of circuits $\mathcal{C}$ is a class of circuits for which there is a QPT algorithm that given any quantum implementation that computes a circuit $C \in \mathcal{C}$, it finds a (possibly different) classical circuit $C' \in \mathcal{C}$ with the same functionality as C. Of course if $\mathcal{C}$ is learnable, then it could be possible to just observe the input-output behavior of the quantum circuit to find such a C'. To make this notion meaningful, we additionally impose the requirement that $\mathcal{C}$ needs to be quantum unlearnable.

Definition 110 (De-quantumizable circuits)**.** *A class of classical circuits $\mathcal{C}$, associated with a distribution $\mathcal{D}_{\mathcal{C}}$, is said to be* **de-quantumizable** *if the following holds:*

- **Efficient de-quantumization***: There is a QPT algorithm $\mathcal{B}$ such that, for any $\{U_C, \rho_C\}_{C\in\mathcal{C}}$ that computes $\mathcal{C}$, the following holds:*

$$\mathsf{Pr}\left[\bigwedge_{\forall x\in\{0,1\}^n, C(x)=C'(x)}^{C'\in\mathcal{C}} \;:\; \begin{matrix} C\leftarrow\mathcal{D}_{\mathcal{C}} \\ C'(x)\leftarrow\mathcal{B}(U_C,\rho_C)\end{matrix}\right] \geq 1-negl(\lambda)$$

- **ν-Quantum Unlearnability***: For any QPT adversary $\mathcal{A}$, the following holds:*

$$\mathsf{Pr}\left[\forall x, \mathsf{Pr}[U^*(\rho^*,x)=C(x)]\geq\nu \;:\; \begin{matrix} C\leftarrow\mathcal{D}_{\mathcal{C}} \\ (U^*,\rho^*)\leftarrow\mathcal{A}^{C(\cdot)}(1^\lambda)\end{matrix}\right] \leq negl(\lambda)$$

Remark 111. *By the Almost As Good As New Lemma, we can assume that the QPT algorithm U_C also outputs a state $\rho'_{C,x}$ that is negligibly close in trace distance to ρ_C, i.e. for all $C \in \mathcal{C}$ and $x \in \{0,1\}^n$ it holds that*

$$\mathsf{Pr}[U_C(\rho_C, x) = (\rho'_{C,x}, C(x))] \geq 1 - negl(\lambda)$$

and $\left\|\rho'_{C,x} - \rho_C\right\|_{tr} \leq negl(\lambda)$.

Remark 112. *We emphasize that the efficient de-quantumization property requires that the circuit C' output by the adversary should be in the same circuit class $\mathcal{C}$.*

Remark 113. *We can relax the unlearnability condition in the above definition to instead have a distribution over the inputs and have the guarantee that the adversary has to output a circuit (U^*, ρ^*) such that it agrees with C only on inputs drawn from this distribution. Our impossibility result will also rule out this relaxed unlearnability condition; however, for simplicity of exposition, we consider the unlearnability condition stated in the above definition.*

Impossibility of copy-protection. From the definition above, we can see why a de-quantumizable class $\mathcal{C}$ cannot be copy-protected, as there is a QPT $\mathcal{B}$ that takes any (U_C, ρ_C) efficiently computing C, and outputs a functionally equivalent *classical* circuit C', which can be copied. We leave the formal details of the impossibility result after we have constructed de-quantumizable circuits.

6.1.1 Constructing de-quantumizable circuits

We turn our attention to the construction of a de-quantumizable circuits class $(\mathcal{C}, \mathcal{D}_{\mathcal{C}})$. Our family has the property that every circuit in the support of $\mathcal{D}_{\mathcal{C}}$ has a unique representation in $\mathcal{C}$[1].

Constructing de-quantumizable circuits: Challenges. The starting point is the seminal work of Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan, and Yang [BGI+01], who demonstrated a class of functions, where each function is associated with a secret key sk, such that: (a) *Non-black-box secret extraction*: given non-black-box access to any classical circuit implementation of this function, the key can be efficiently recovered, (b) *Classical unlearnability of secrets*: but given black-box access to this circuit, any classical adversary who can only make polynomially many queries to the oracle cannot recover the key.

While the result of Barak et al. has the ingredients suitable for us, it falls short in many respects:

- The proof of non-black-box secret extraction crucially relies upon the fact that we are only given a classical obfuscated circuit. In fact there are inherent difficulties that we face in adapting Barak et al. to the quantum setting; see [AF16].

- As is the case with many black-box extraction techniques, the proof of Barak et al. involves evaluating the obfuscated circuit multiple times in order to recover the secret. As is typically the case with quantum settings, evaluating the same circuit again and again is not always easy – the reason being that evaluating a circuit once could potentially destroy the state thus rendering it impossible to run it again.

- Barak et al. only guarantees extraction of secrets given non-black-box access to the classical circuit implementation of the function. However, our requirement is qualitatively different: given a quantum implementation of the classical circuit, we need to find a (possible different) classical circuit with the same functionality.

- Barak et al.'s unlearnability result only ruled out adversaries who make classical queries to the oracle. On the other hand, we need to argue unlearnability against QPT adversaries who can perform superposition queries to the oracle.

Nonetheless, we show that the techniques introduced in a simplified version of Barak[2] can be suitably adapted for our purpose by using the two tools we already used in Section 3.3: quantum fully homomorphic encryption (QFHE) and lockable obfuscation.

[1] This property will be relevant when extending the impossibility result to SSL.
[2] See [BP13] for a description of this simplified version.

Construction. We present the construction of de-quantumizable circuits.

Theorem 114. *Assuming the quantum hardness of learning with errors (QLWE), and assuming that there is a* QFHE *that supports evaluation of arbitrary polynomial-sized quantum circuits, and has the following two properties: (a) ciphertexts have classical plaintexts have classical descriptions and, (b) classical ciphertexts can be decrypted using a classical circuit,*

there exists a de-quantumizable class of circuits $(\mathcal{C}, \mathcal{D}_{\mathcal{C}})$.

Proof. We define a de-quantumizable class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$, where every circuit in $\mathcal{C}_\lambda$ is defined as shown in Figure 6-1. The circuits are assumed to be suitably padded with zeroes such that all the inputs (resp., outputs) are of the same length n (resp., of the same length m). The distribution associated to $\mathcal{C}$ is described in Figure 6-2.

$\underline{C_{a,b,r,\mathsf{pk},\mathcal{O}}(x)}$:

1. If $x = 0\cdots0$, output $\mathsf{QFHE.Enc}\,(\mathsf{pk}, a; r)\,|\mathcal{O}|\mathsf{pk}$.
2. Else if $x = a$, output b.
3. Otherwise, output $0\cdots0$

Figure 6-1: De-quantumizable circuit class

$\underline{\mathcal{D}_{\mathcal{C}}(\lambda)}$:

1. Sample $a, b, r \xleftarrow{\$} \{0,1\}^\lambda$.
2. Compute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$.
3. Compute $\mathcal{O} \leftarrow \mathsf{LO.Obf}(\mathbf{C}[\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), b, (\mathsf{sk}|r)])$ where $\mathbf{C}$ is a compute-and-compare circuit (see Definition 38).
4. Output $(a, b, r, \mathsf{pk}, \mathcal{O})$.

Figure 6-2: Distribution associated to $\mathcal{C}$

We show that with respect to the distribution in Figure 6-2: (a) $\mathcal{C}$ is quantum unlearnable (Proposition 115) and, (b) $\mathcal{C}$ is efficiently de-quantumizable (Proposition 118).

Proposition 115. *For any non-negligible ν, the circuit class $\mathcal{C}$ is ν-quantum unlearnable with respect to $\mathcal{D}_{\mathcal{C}}$.*

Proof. We first rule out QPT adversaries, who given black-box access to the circuit, can find the secret key sk with non-negligible probability. Once we rule out this type of adversaries, we then show how to reduce a QPT adversary who breaks the quantum unlearnability property of the de-quantumizable class of circuits to one who finds the secret key sk; thus completing the proof.

Claim 116. *For any QPT $\mathcal{A}$ with quantum oracle access to $C_{a,b,r,\mathsf{pk},\mathcal{O}}(\cdot)$ (where the adversary is allowed to make superposition queries), we have*

$$\Pr_{(a,b,r,\mathsf{pk},\mathcal{O})\leftarrow\mathcal{D}_{\mathcal{C}}}\left[\mathsf{sk}\leftarrow\mathcal{A}^{C_{a,b,r,\mathsf{pk},\mathcal{O}}}\left(1^{\lambda}\right)\right]\leq negl(\lambda)$$

Proof. Towards proving this, we make some simplifying assumptions; this is only for simplicity of exposition and they are without loss of generality.

Simplifying Assumptions. Consider the following oracle $O_{a,b,r,\mathsf{pk},\mathcal{O}}$:

$$O_{a,b,r,\mathsf{pk},\mathcal{O}}|x\rangle|z\rangle=\begin{cases}|x\rangle|z\oplus C_{a,b,r,\mathsf{pk},\mathcal{O}}(x)\rangle, & \text{if } x\neq 0\cdots 0\\ |x\rangle|z\rangle, & \text{if } x=0\cdots 0\end{cases}$$

The first simplifying assumption is that the adversary $\mathcal{A}$ is given access to the oracle $O_{a,b,r,\mathsf{pk},\mathcal{O}}$, instead of the oracle $C_{a,b,r,\mathsf{pk},\mathcal{O}}$. In addition, $\mathcal{A}$ is given $\mathsf{Enc}(\mathsf{pk},a;r)$, pk, and $\mathcal{O}$ as auxiliary input.

The second simplifying assumption is that $\mathcal{A}$ is given some auxiliary state $|\xi\rangle$, and that it only performs computational basis measurements right before outputting (i.e. $\mathcal{A}$ works with purified states).

Overview. Our proof follows the adversary method proof technique [Amb02]. We prove this by induction on the number of queries. We show that after every query the following invariant is maintained: the state of the adversary has little amplitude over a. More precisely, we argue that the state of the adversary after the t^{th} query, is neglibly close to the state just before the t^{th} query, denoted by $|\psi^t\rangle$. After the adversary obtains the response to the t^{th} query, it then applies a unitary operation to obtain the state $|\psi^{t+1}\rangle$, which is the state of the adversary just before the $(t+1)^{th}$ query. This observation implies that there is another state $|\phi^{t+1}\rangle$ that: (a) is close to $|\psi^{t+1}\rangle$ (here, we use the inductive hypothesis that $|\phi^t\rangle$ is close to $|\psi^t\rangle$) and, (b) can be prepared without querying the oracle at all.

Let U_i denote the unitary that $\mathcal{A}$ performs right before its i^{th} query, and let $\mathbf{A},\mathbf{X}$, and $\mathbf{Y}$ denote the private, oracle input, and oracle output registers of $\mathcal{A}$, respectively.

Just before the t^{th} query, we denote the state of the adversary to be:

$$|\psi^t\rangle:=U_tO\cdots OU_1|\psi^0\rangle$$

where $|\psi^0\rangle = |\xi\rangle|\mathsf{Enc}(\mathsf{pk}, a; r), \mathcal{O}, \mathsf{pk}\rangle|0\cdots 0\rangle_{\mathbf{X}}|0\cdots 0\rangle_{\mathbf{Y}}$ is the initial state of the adversary. Let $\Pi_a = (|a\rangle\langle a|)_{\mathbf{X}} \otimes I_{\mathbf{Y},\mathbf{A}}$.

Note that any $\mathcal{A}$ that outputs sk with non-negligible probability can also query the oracle on a state $|\psi\rangle$ satisfying $\mathrm{Tr}[\Pi_a|\psi\rangle\langle\psi|] \geq \text{non-negl}(\lambda)$ with non-negligible probability. Since $\mathcal{A}$ outputs sk with non-negligible probability, it can decrypt $\mathsf{Enc}(\mathsf{pk}, a; r)$, to find a and then query the oracle on a. In other words, if there is an adversary $\mathcal{A}$ that finds sk with non-negligible probability, then there is an adversary that at some point queries the oracle with a state $|\psi\rangle$ satisfying $\mathrm{Tr}[\Pi_a|\psi\rangle\langle\psi|] \geq \text{non-negl}(\lambda)$ also with non-negligible probability.

Hence, it suffices to show that for any adversary $\mathcal{A}$ that makes at most $T = \mathrm{poly}(\lambda)$ queries to the oracle, it holds that

$$\mathsf{Pr}[\forall j, \mathrm{Tr}[\Pi_a|\psi^j\rangle\langle\psi^j|] \leq \mathrm{negl}(\lambda)] \geq 1 - \mathrm{negl}(\lambda).$$

This would then imply that $\mathcal{A}$ cannot output sk with non-negligible probability, thus proving Claim 116.

Towards proving the above statement, consider the following claim that states that if $\mathcal{A}$ has not queried the oracle with a state that has large overlap with Π_a, then its next query will also not have large overlap with Π_a.

Claim 117 (No Good Progress)**.** *Let T be any polynomial in λ. Suppose for all $t < T$, the following holds:*

$$Tr\left[\Pi_a|\psi^t\rangle\langle\psi^t|\right] \leq negl(\lambda)$$

Then, $\mathsf{Pr}[Tr[\Pi_a|\psi^T\rangle\langle\psi^T|] \leq negl(\lambda)] \geq 1 - negl(\lambda)$.

Proof. For all j, let $|\phi^j\rangle = U_j U_{j-1} ... U_1|\psi^0\rangle$.

We will proceed by induction on T. Our base case is $T = 1$ (just before the first query to the oracle); that is, $|\psi^1\rangle = |\phi^1\rangle$. Suppose the following holds:

$$\mathsf{Pr}[\mathrm{Tr}[\Pi_a|\psi^1\rangle\langle\psi^1|] \geq \text{non-negl}(\lambda)] \geq \text{non-negl}(\lambda).$$

The first step is to argue that if $\mathcal{A}$ can prepare a state such that $\mathrm{Tr}[\Pi_a|\psi_1\rangle\langle\psi_1|] \geq$ non-negl(λ) given $\mathsf{Enc}(\mathsf{pk}, a; r), \mathsf{pk}$ and $\mathcal{O} \leftarrow \mathsf{LO.Obf}(\mathbf{C}\,[\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), b, (\mathsf{sk}|r)])$ without querying the oracle, then it can also prepare a state with large overlap with Π_a if its given the simulator of the lockable obfuscation instead. We will use $\mathcal{A}$ (specifically, the first unitary that $\mathcal{A}$ applies, U_1) to construct an adversary $\mathcal{B}$ that breaks the security of lockable obfuscation. $\mathcal{B}$ is given a, $\mathsf{Enc}(\mathsf{pk}, a; r)$, pk and $\mathcal{O}$ as well as auxiliary state $|\xi\rangle$. It the prepares $|\psi_{1,\mathcal{O}}\rangle = U_1|\xi\rangle|\mathsf{Enc}(\mathsf{pk}, a; r), \mathcal{O}, \mathsf{pk}\rangle|0\cdots 0\rangle_{\mathbf{X}}|0\cdots 0\rangle_{\mathbf{Y}}$, and measures in computational basis. If the output of this measurement is a, it outputs 1; otherwise, it outputs 0.

Consider the following hybrids.

•Hyb_1 In this hybrid, $\mathcal{B}$ is given a, $\mathsf{Enc}(\mathsf{pk}, a; r), \mathsf{pk}, \mathcal{O} \leftarrow \mathsf{LO.Obf}(\mathbf{C}[\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), b, (\mathsf{sk}|r)])$.

•Hyb_2: In this hybrid, $\mathcal{B}$ is given a, $\mathsf{Enc}(\mathsf{pk}, a; r), \mathsf{pk}$ and $\mathcal{O} \leftarrow \mathsf{Sim}(1^\lambda)$.

Since the lock b is chosen uniformly at random, by security of lockable obfuscation, the probability that $\mathcal{B}$ outputs 1 in the first hybrid is negligibly close to the probability that $\mathcal{B}$ outputs 1 in the second hybrid. This means that if $\mathrm{Tr}[\Pi_a|\psi_{1,\mathcal{O}}\rangle\langle\psi_{1,\mathcal{O}}|] \geq \text{non-negl}(\lambda)$ with non-negligible probability when $\mathcal{O} \leftarrow \mathsf{LO.Obf}(\mathbf{C}[\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), b, (\mathsf{sk}|r)])$, then this still holds when $\mathcal{O} \leftarrow \mathsf{Sim}(1^\lambda)$.

But we show that if $\mathrm{Tr}[\Pi_a|\psi_{1,\mathcal{O}}\rangle\langle\psi_{1,\mathcal{O}}|] \geq \text{non-negl}(\lambda)$, when $\mathcal{O}$ is generated as $\mathcal{O} \leftarrow \mathsf{Sim}(1^\lambda)$, then QFHE is insecure.

- Consider the following QFHE adversary who is given $|\xi\rangle$ as auxiliary information, and chooses two messages $m_0 = 0 \cdots 0$ and $m_1 = a$, where a is sampled uniformly at random from $\{0,1\}^\lambda$. It sends (m_0, m_1) to the challenger.
- The challenger of QFHE then generates $\mathsf{ct}_d = \mathsf{Enc}(\mathsf{pk}, m_d)$, for some bit $d \in \{0,1\}$ and sends it to the QFHE adversary.
- The QFHE adversary computes $\mathcal{O} \leftarrow \mathsf{Sim}(1^\lambda)$.
- The QFHE adversary then prepares the state $|\psi_d\rangle = U_1\left(|\xi\rangle|\mathsf{ct}_d, \mathcal{O}, \mathsf{pk}\rangle|0 \cdots 0\rangle_{\mathbf{X}}|0 \cdots 0\rangle_{\mathbf{Y}}\right)$ and measures register $\mathbf{X}$ in the computational basis.

If $d = 0$, the probability that the QFHE adversary obtains a as outcome is negligible; since a is independent of U_1, pk, $|\xi\rangle$, and $\mathcal{O}$. But from our hypothesis ($\mathsf{Pr}[\mathrm{Tr}[\Pi_a|\psi^1\rangle\langle\psi^1|] \geq \text{non-negl}(\lambda)] \geq \text{non-negl}(\lambda)$), the probability that the QFHE adversary obtains a as outcome is non-negligible for the case when $d = 1$. This contradicts the security of QFHE as the adversary can use a to distinguish between these two cases.

To prove the induction hypothesis, suppose that for all $t < T$, the following two conditions hold:

1. $\mathrm{Tr}[\Pi_a|\psi^t\rangle\langle\psi^t|] \leq \text{negl}(\lambda)$
2. $|\langle\phi^t|\psi^t\rangle| = 1 - \delta_t$

for some negligible $\delta_1, ..., \delta_{T-1}$. We can write

$$|\langle\phi^T|\psi^T\rangle| = |\langle\phi^{T-1}|O|\psi^{T-1}\rangle|$$

By hypothesis (2) above, we have $|\phi^{T-1}\rangle = (1-\delta_{T-1})e^{i\alpha}|\psi^{T-1}\rangle + \sqrt{2\delta_{T-1} - \delta_{T-1}^2}|\widetilde{\psi}^{T-1}\rangle$, here α is some phase, and $|\widetilde{\psi}^{T-1}\rangle$ is some state orthogonal to $|\psi^{T-1}\rangle$. Then

$$\begin{aligned}
|\langle\phi^T|\psi^T\rangle| &= |(1-\delta_{T-1})e^{i\alpha}\langle\psi^{T-1}|O|\psi^{T-1}\rangle + \sqrt{2\delta_{T-1} - \delta_{T-1}^2}\langle\widetilde{\psi}^{T-1}|O|\psi^{T-1}\rangle| \\
&\geq |(1-\delta_{T-1})e^{i\alpha}\langle\psi^{T-1}|O|\psi^{T-1}\rangle| - \sqrt{2\delta_{T-1} - \delta_{T-1}^2} \\
&\geq (1-\delta_{T-1})|\langle\psi^{T-1}|O|\psi^{T-1}\rangle| - \sqrt{2\delta_{T-1} - \delta_{T-1}^2}
\end{aligned}$$

By hypothesis (1) above, and since the oracle acts non-trivially only on a, we have $|\langle\psi^{T-1}|O|\psi^{T-1}\rangle| \geq 1 - \mathrm{negl}(\lambda)$, which gives us

$$|\langle\phi^T|\psi^T\rangle| \geq 1 - \mathrm{negl}(\lambda).$$

Now we want to show that $\mathrm{Tr}[\Pi_a|\psi^T\rangle\langle\psi^T|] \leq \mathrm{negl}(\lambda)$. This follows from the security of lockable obfuscation and QFHE similarly to $T = 1$ case. Since $|\langle\phi^T|\psi^T\rangle| \geq 1 - \mathrm{negl}(\lambda)$, we have that

$$\mathrm{Tr}[\Pi_a|\phi^T\rangle\langle\phi^T|] \leq \mathrm{negl}(\lambda) \implies \mathrm{Tr}[\Pi_a|\psi^T\rangle\langle\psi^T|] \leq \mathrm{negl}(\lambda).$$

From a similar argument to the $T = 1$ case but using $U_T U_{T-1} \cdots U_1$ instead of just U_1, we have that $\mathsf{Pr}[\mathrm{Tr}[\Pi_a|\phi^T\rangle\langle\phi^T|] \leq \mathrm{negl}(\lambda)] \geq 1 - \mathrm{negl}(\lambda)$. □

Let E_i denote the event that $\mathrm{Tr}[\Pi_a|\psi^i\rangle\langle\psi^i|] \leq \mathrm{negl}(\lambda)$. Let p_T be the probability that $\mathrm{Tr}[\Pi_a|\psi^t\rangle\langle\psi^t|] \leq \mathrm{negl}(\lambda)$ for all the queries $t \leq T$. Using the previous claim, we have that

$$\begin{aligned} p_T &= \prod_{t=1}^{T} \mathsf{Pr}[E_t|\forall j < t, E_j] \\ &\geq (1 - \mathrm{negl}(\lambda))^T \\ &\geq (1 - T \cdot \mathrm{negl}(\lambda)) \end{aligned}$$

□

Suppose that there is a QPT $\mathcal{B}$ that can learn $\mathcal{C}$ with respect to $\mathcal{D}_{\mathcal{C}}$ with non-negligible probability δ. In other words, for all inputs x,

$$\mathsf{Pr}\left[U(\rho, x) = C_{a,b,r,\mathsf{pk},\mathcal{O}}(x) : \substack{C_{a,b,r,\mathsf{pk},\mathcal{O}} \leftarrow \mathcal{D}_{\mathcal{C}} \\ (U,\rho) \leftarrow \mathcal{B}^{C_{a,b,r,\mathsf{pk},\mathcal{O}}}(1^\lambda)}\right] = \delta$$

We use $\mathcal{B}^{C_{a,b,r,\mathsf{pk},\mathcal{O}}}$ to construct a QPT $\mathcal{A}^{C_{a,b,r,\mathsf{pk},\mathcal{O}}}$ that can find sk with probability negligibly close to δ, contradicting Claim 116. To do this, $\mathcal{A}$ first prepares $(U, \rho) \leftarrow \mathcal{B}^{C_{a,b,r,\mathsf{pk},\mathcal{O}}}(1^\lambda)$. Then, $\mathcal{A}^{C_{a,b,r,\mathsf{pk},\mathcal{O}}}$ queries the oracle on input $0 \cdots 0$, obtaining $\mathsf{ct}_1 = \mathsf{QFHE.Enc}(\mathsf{pk}, a; r)$ along with pk and $\mathcal{O} = \mathsf{LO.Obf}(\mathbf{C}[\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), b, (\mathsf{sk}|r)])$. Finally, it homomorphically computes $\mathsf{ct}_2 \leftarrow \mathsf{QFHE.Eval}(U(\rho, \cdot), \mathsf{ct}_1)$. Then it computes $\mathsf{sk}'|r' = \mathcal{O}(\mathsf{ct}_2)$, and outputs sk'.

By the correctness of the QFHE and because $U(\rho, a) = b$ holds with probability δ, we have that $\mathsf{QFHE.Dec}_{\mathsf{sk}}(\mathsf{ct}_2) = b$ with probability negligibly close to δ. By correctness of lockable obfuscation $\mathcal{O}(\mathsf{ct}_2)$ will output the right message sk. This means that output of $\mathcal{A}$ is sk with probability negligibly close to δ.

□

Proposition 118. $(\mathcal{C}, \mathcal{D}_{\mathcal{C}})$ *is efficiently de-quantumizable.*

Proof. We will start with an overview of the proof.

Overview: Given a quantum circuit (U_C, ρ_C) that computes $C_{a,b,r,\mathsf{pk},\mathcal{O}}(\cdot)$, first compute on the input $x = 0 \cdots 0$ to obtain $\mathsf{QFHE.Enc}(\mathsf{pk}, a; r)|\mathcal{O}|\mathsf{pk}$. We then homomorphically evaluate the quantum circuit on $\mathsf{QFHE.Enc}(\mathsf{pk}, a; r)$ to obtain $\mathsf{QFHE.Enc}(\mathsf{pk}, b')$, where b' is the output of the quantum circuit on input a; this is part where we crucially use the fact that we are given (U_C, ρ_C) and not just black-box access to the functionality computing (U_C, ρ_C). But b' is nothing but b! Given QFHE encryption of b, we can then use the lockable obfuscation to recover sk; since the lockable obfuscation on input a valid encryption of b outputs sk. Using sk we can then recover the original circuit $C_{a,b,r,\mathsf{pk},\mathcal{O}}(\cdot)$. Formal details follow.

For any $C \in \mathcal{C}$, let (U_C, ρ_C) be any QPT algorithm (with auxiliary state ρ_C) satisfying that for all $x \in \{0,1\}^n$,

$$\mathsf{Pr}\left[U_C(\rho_C, x) = \left(\rho'_{C,x}, C(x)\right)\right] \geq 1 - \mathrm{negl}(\lambda),$$

where the probability is over the measurement outcomes of U_C, and $\rho'_{C,x}$ is negligibly close in trace distance to ρ_C (see Remark 111). We will show how to constuct a QPT $\mathcal{B}$ to de-quantumize $(\mathcal{C}, \mathcal{D}_{\mathcal{C}})$.

$\mathcal{B}$ will perform a QFHE evaluation, which we describe here. Given $\mathsf{QFHE.Enc}(\mathsf{pk}, x)$, we want to homomorphically evaluate $C(x)$ to obtain $\mathsf{QFHE.Enc}(\mathsf{pk}, C(x))$. To do this, first prepare $\mathsf{QFHE.Enc}(\mathsf{pk}, \rho_C, x)$, then evaluate U_C homomorphically to obtain the following:

$$\mathsf{QFHE.Enc}(\mathsf{pk}, \rho'_{C,x}, C(x)) = \mathsf{QFHE.Enc}(\mathsf{pk}, \rho'_{C,x}) \big| \mathsf{QFHE.Enc}(\mathsf{pk}, C(x))$$

Consider the following QPT algorithm $\mathcal{B}$ that is given (U_C, ρ_C) for any $C \in \mathcal{C}$.

$\mathcal{B}(U_C, \rho_C)$**:**

1. Compute $(\rho', \mathsf{ct}_1|\mathcal{O}'|\mathsf{pk}') \leftarrow U_C(\rho_C, 0 \cdots 0)$.
2. Compute $\sigma|\mathsf{ct}_2 \leftarrow \mathsf{QFHE.Eval}(U_C(\rho', \cdot), \mathsf{ct}_1)$
3. Compute $\mathsf{sk}'|r' \leftarrow \mathcal{O}(\mathsf{ct}_2)$
4. Compute $a' \leftarrow \mathsf{QFHE.Dec}(\mathsf{sk}', \mathsf{ct}_1)$, $b' \leftarrow \mathsf{QFHE.Dec}(\mathsf{sk}', \mathsf{ct}_2)$.
5. Output $C_{a',b',r',\mathsf{pk}',\mathcal{O}'}$.

We claim that with probability negligibly close to 1, $(a', b', r', \mathsf{pk}', \mathcal{O}') = (a, b, r, \mathsf{pk}, \mathcal{O})$ when $C := C_{a,b,r,\mathsf{pk},\mathcal{O}} \leftarrow \mathcal{D}_{\mathcal{C}}$. This would finish our proof.

Lets analyze the outputs of $\mathcal{B}$ step-by-step.

- After Step (1), with probability neglibibly close to 1, we have that $\mathsf{ct}_1 = \mathsf{QFHE.Enc}(\mathsf{pk}, a; r)$, $\mathsf{pk}' = \mathsf{pk}$, and $\mathcal{O}' = \mathcal{O} \leftarrow \mathsf{LO.Obf}(\mathbf{C}[\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), b, (\mathsf{sk}|r)])$. Furthermore, we have that ρ' is negligibly close in trace distance to ρ_C.

- Conditioned on Step (1) computing $C(0\cdots 0)$ correctly, we have that $\mathsf{QFHE.Eval}(U_C(\rho',.),\mathsf{ct}_1)$ computes correctly with probability negligibly close to 1. This is because $\|\rho'-\rho_C\|_{\mathrm{tr}}\leq \mathrm{negl}(\lambda)$, and by correctness of both QFHE and (U_C,ρ_C). Conditioned on $\mathsf{ct}_1=\mathsf{QFHE.Enc}(\mathsf{pk},a;r)$, when Step (2) evaluates correctly, we have $\mathsf{ct}_2=\mathsf{QFHE.Enc}(\mathsf{pk},C(a))=\mathsf{QFHE.Enc}(\mathsf{pk},b)$

- Conditioned on $\mathsf{ct}_2=\mathsf{QFHE.Enc}(\mathsf{pk},b)$, by correctness of lockable obfuscation, we have that $\mathcal{O}(\mathsf{ct}_2)$ outputs $\mathsf{sk}|r$. Furthermore, by correctness of QFHE, decryption is correct: $\mathsf{QFHE.Dec}(\mathsf{sk},\mathsf{ct}_1)$ outputs a with probability negligibly close to 1, and $\mathsf{QFHE.Dec}(\mathsf{sk},\mathsf{ct}_2)$ outputs b with probability negligibly close to 1.

With probability negligibly close to 1, we have shown that $(a',b',r',\mathsf{pk}',\mathcal{O}')=(a,b,r,\mathsf{pk},\mathcal{O})$.

Note that it is also possible to recover ρ'' that is neglibly close in trace distance to ρ_C. This is because $\sigma=\mathsf{QFHE.Enc}(\mathsf{pk},\rho'')$ for some ρ'' satisfying $\|\rho''-\rho_C\|_{\mathrm{tr}}$. Once $\mathsf{sk}'=\mathsf{sk}$ has been recovered, it is possible to also decrypt σ and obtain ρ''. To summarize, we have shown a QPT $\mathcal{B}$ satisfying

$$\mathsf{Pr}[\mathcal{B}(U_C,\rho_C)=(\rho'',C)\ :\ C\leftarrow\mathcal{D}_{\mathcal{C}}]\geq 1-\mathrm{negl}(\lambda)$$

where $\|\rho''-\rho_C\|_{\mathrm{tr}}\leq\mathrm{negl}(\lambda)$. □

□

6.2 Impossibility of Copy-Protection and QVBB

We have constructed a class $\mathcal{C}$ and an associated distribution $\mathcal{D}_{\mathcal{C}}$ that is efficient de-quantumizable. In particular, this means that there is no copy-protection for $\mathcal{C}$. If for all inputs x, there is a QPT (U_C,ρ_C) to compute $U_C(\rho_C,x)=C(x)$ with probability $1-\varepsilon$ for some negligible ε, then it is possible to find, with probability close to 1, a circuit C' that computes the same functionality as C. We also proved that $(\mathcal{C},\mathcal{D}_{\mathcal{C}})$ is quantum unlearnable. We summarize the result in the following corollary,

Corollary 119. *There is $(\mathcal{C},\mathcal{D}_{\mathcal{C}})$ that is quantum unlearnable, but $\mathcal{C}$ cannot be copy-protected against $\mathcal{D}_{\mathcal{C}}$. Specifically, for any $C\leftarrow\mathcal{D}_{\mathcal{C}}$ with input length n, and for any QPT algorithm (U_C,ρ_C) satisfying that for all $x\in\{0,1\}^n$,*

$$\mathsf{Pr}[U_C(\rho_C,x)=C(x)]\geq 1-\varepsilon$$

for some negligible ε, there is a QPT algorithm (pirate) that outputs a circuit C', satisfying $C'(x)=C(x)$ for all $x\in\{0,1\}^n$, with probability negligibly close to 1.

Further Discussion. Notice that in our proof that $\mathcal{C}$ is efficient de-quantumizable, we just need to compute $U_C(\rho_C,x)$ at two different points $x_1=0\cdots 0$ and $x_2=a$, where the evaluation at x_2 is done homomorphically. This means that any scheme that lets a user evaluate a circuit C at least 2 times (for 2 possibly different inputs) with non-negligible probability cannot be copy-protected. Such a user would be able

to find all the parameters of the circuit, $(a, b, r, \mathsf{pk}, \mathcal{O})$, succesfully with non-negligible probability, hence it can prepare as many copies of a functionally equivalent circuit C'.

In our proof, we make use of the fact that (U_C, ρ_C) evaluates correctly with probability close to 1. This is in order to ensure that the pirate can indeed evaluate at 2 points by uncomputing after it computes $C(0\cdots 0)$. Since any copy-protection scheme can be amplified to have correctness neglibly close to 1 by providing multiple copies of the copy-protected states, our result also rules out copy-protection for non-negligible correctness parameter ε. As long as the correctness of (U_C, ρ_C) can be amplified to neglibily close to 1 by providing $\rho_C^{\otimes k}$ for some $k = \mathrm{poly}(\lambda)$, a pirate can get a hold of many copies and evaluate at the 2 points necessary to break the scheme.

Impossibility of Quantum VBB with single unclonable state. Our techniques also rule out the possibility of quantum VBB for classical circuits. In particular, this rules the possibility of quantum VBB for classical circuits with the obfucated circuit being a single unclonable state, thus resolving an open problem by Alagic and Fefferman [AF16].

Proposition 120. *Assuming the quantum hardness of learning with errors and assuming that there is a QFHE satisfying the properties described in Theorem 114, there exists a circuit class $\mathcal{C}$ such that any quantum VBB for $\mathcal{C}$ is insecure.*

Proof. We construct a circuit class $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$, where every circuit in $\mathcal{C}_\lambda$ is of the form $C_{a,b,r,\mathsf{pk},\mathcal{O}}$ defined in the proof of Theorem 114.

Given any quantum VBB of $C_{a,b,r,\mathsf{pk},\mathcal{O}}$, there exists an adversary $\mathcal{A}$ that recovers b and outputs the first bit of b. The adversary $\mathcal{A}$ follows steps 1-4 of $\mathcal{B}$ defined in the proof of Proposition 118 and then outputs the first bit of b'. In the same proof, we showed that the probability that $b' = b$ is negligibly close to 1 and thus, the probability it outputs the first bit of b is negligibly close to 1.

On the other hand, any QPT simulator Sim with superposition access to $C_{a,b,r,\mathsf{pk},\mathcal{O}}$ can recover b with probability negligibly close to $1/2$. To prove this, we rely upon the proof of Claim 116. We will start with the same simplifying assumptions as made in the proof of Claim 116. Suppose T is the number of superposition queries made by Sim to $C_{a,b,r,\mathsf{pk},\mathcal{O}}$. Let $|\psi^0\rangle$ is the initial state of Sim and more generally, let $|\psi^t\rangle$ be the state of Sim after t queries, for $t \leq T$.

We define an alternate QPT simulator Sim' which predicts the first bit of b with probability negligibly close to Sim. Before we describe Sim', we give the necessary preliminary background. Define $|\phi^t\rangle = U_t U_{t-1}\cdots U_1|\psi^0\rangle$. We proved the following claim.

Claim 121. *$|\langle\phi^t|\psi^t\rangle| = 1 - \delta_t$ for every $t \in [T]$.*

Sim' starts with the initial state $|\psi^0\rangle$. It then computes $|\phi^T\rangle$. If U is a unitary matrix Sim applies on $|\psi^T\rangle$ followed by a measurement of a register $\mathbf{D}$ then Sim' also performs U on $|\phi^T\rangle$ followed by a measurement of $\mathbf{D}$. By the above claim, it then follows that the probability that Sim' outputs 1 is negligibly close to the probability that Sim

outputs 1. But the probability that Sim' predicts the first bit of b is $1/2$. Thus, the probability that Sim predicts the first bit of b is negligibly close to $1/2$. $\square$

Chapter 7

Secure Software Leasing

7.1 Introduction

Almost all proprietary software requires a legal document, called software license, that governs the use against illegal distribution of software, also referred to as pirating. The main security requirement from such a license is that any malicious user no longer has access to the functionality of the software after the lease associated with the software license expires. While ad hoc solutions existed in the real world, for a long time, no theoretical treatment of this problem was known. This was until Aaronson, who in his seminal work [Aar09] introduced and formalized the notion of quantum software copy-protection, a quantum cryptographic primitive that uses quantum no-cloning techniques to prevent pirating of software by modeling software as boolean functions. Quantum copy-protection would prevent a pirate from being able to create a new software from his own copy and re-distribute it; of course it can circulate its own copy to others but it will lose access to its own copy.

Need for Alternate Notions. While quantum copy-protection does provide a solution for software piracy, constructing quantum copy-protection has been notoriously difficult. Despite being introduced more than a decade ago, not much is known on the existence of quantum copy-protection. There are no known provably secure constructions of quantum copy-protection for *any* class of circuits. All the existing constructions of quantum copy-protection are either proven in an oracle model [Aar09, ALL+20] or are heuristic[1] candidates for very simple functions such as point functions [Aar09]. In a recent blog post, Aaronson [Aar] even mentioned constructing quantum copy-protection from cryptographic assumptions as one of the five big questions he wishes to solve.

This not only prompted us to explore the possibility of copy-protection but also look for alternate notions to protect against software piracy. Specifically, we look for application scenarios where the full power of quantum copy-protection is not needed and it suffices to settle for weaker notions. Let us consider one such example.

[1] That is, there is no known reduction to concrete cryptographic assumptions.

Example: Anti-Piracy Solutions for Microsoft Office. Microsoft Office is one of the most popular software tools used worldwide. Since Microsoft makes a sizeable portion of their revenue from this tool [rev], it is natural to protect Microsoft Office from falling prey to software piracy. A desirable requirement is that pirated copies cannot be sold to other users such that these copies can run successfully on other Microsoft Windows systems. Importantly, it does not even matter if the pirated copies can be created as long as they cannot be executed on other Windows systems; this is because, only the pirated copies that run on Windows systems are the ones that bite into the revenue of Microsoft. Indeed, there are open source versions of Office publicly available but our aim is to prevent these open source versions from being sold off as authentic versions of Microsoft Office software.

This suggests that instead of quantum copy-protection – which prevents the adversary from creating *any* pirated copy of the copy-protected software – we can consider a weaker variant that only prevents the adversary from being able to create *authenticated* pirated copies (for instance, that runs only on specific operating systems). To capture this, we present a new definition called secure software leasing.

Our Work: Secure Software Leasing (SSL). Roughly speaking, a secure leasing scheme allows for an authority (the lessor[2]) to lease a classical circuit C to a user (the lessee[3]) by providing a corresponding quantum state ρ_C. The user can execute ρ_C to compute C on any input it desires. Leases can expired, requiring ρ_C to be returned at a later point in time, specified by the lease agreement. After it returns the state, we require the security property that the lessee can no longer compute C.

In more detail, a secure software leasing scheme (SSL) for a family of circuits $\mathcal{C}$ is a collection, $(\mathsf{Gen}, \mathsf{Lessor}, \mathsf{Run}, \mathsf{Check})$, of quantum polynomial-time algorithms (QPT) satisfying the following conditions. $\mathsf{Gen}(1^\lambda)$, on input a security parameter λ, outputs a secret key sk that will be used by a lessor to validate the states being returned after the expiration of the lease. For any circuit $C : \{0,1\}^n \to \{0,1\}^m$ in $\mathcal{C}$, $\mathsf{Lessor}(\mathsf{sk}, C)$ outputs a quantum state ρ_C, where ρ_C allows Run to evaluate C. Specifically, for any $x \in \{0,1\}^n$, we want that $\mathsf{Run}(\rho_C, x) = C(x)$; this algorithm is executed by the lessee. Finally, $\mathsf{Check}(\mathsf{sk}, \rho_C)$ checks if ρ_C is a valid leased state. Any state produced by the lessor is a valid state and will pass the verification check.

A SSL scheme can have two different security guarantees depending on whether the leased state is supposed to be returned or not.

- *Infinite-Term Lessor Security*: In this setting, there is no time duration associated with the leased state and hence, the user can keep this leased state forever[4]. Informally, we require the guarantee that the lessee, using the leased state, cannot produce two *authenticated* copies of the leased state. Formally speaking, any (malicious) QPT user $\mathcal{A}$ holding a leased state $\mathcal{A}(\rho_C)$ (produced using classical circuit C) cannot output a (possibly entangled) bipartite state

[2]The person who leases the software to another.
[3]The person to whom the software is being leased to.
[4]Although the lessor will technically be the owner of the leased state.

σ^* such that both $\sigma_1^* = \mathrm{Tr}_2[\sigma^*]$ and $\sigma_2^* = \mathrm{Tr}_1[\sigma^*]$ can be used to compute C with Run.

- *Finite-Term Lessor Security*: On the other hand, we could also consider a weaker setting where the leased state is associated with a fixed term. In this setting, the lessee is obligated to return back the leased state after the term expires. We require the property that after the lessee returns back the state, it can no longer produce another *authenticated* state having the same functionality as the leased state.

  Formally speaking, we require that any (malicious) QPT user $\mathcal{A}$ holding a leased state ρ_C (produced using C) cannot output a (possibly entangled) bipartite states σ^* such that $\sigma_1^* := \mathrm{Tr}_2[\sigma^*]$[5] passes the lessor's verification ($\mathsf{Check}(\mathsf{sk}, \sigma_1^*) = 1$) and such that the the resulting state, after the first register has been verified by the lessor, on the second register, σ_2^*, can also be used to evaluate C with the Run algorithm, $\mathsf{Run}(\sigma_2^*, x) = C(x)$.

A SSL scheme satisfying infinite-term security would potentially be useful to tackle the problem of developing anti-piracy solutions for Microsoft Office. However, there are scenarios where finite-term security suffices. We mention two examples below.

Trial Versions. Before releasing the full version of a program C, a software vendor might want to allow a selected group of people[6] to run a beta version of it, C_β, in order to test it and get user feedback. Naturally, the vendor would not want the beta versions to be pirated and distributed more widely. Again, they can lease the beta version C_β, expecting the users to return it back when the beta test is over. At this point, they would know if a user did not return their beta version and they can penalize such a user according to their lease agreement.

Subscription Models. Another example where finite-term SSL would be useful is for companies that use a subscription model for their revenue. For example, Microsoft has a large library of video games for their console, the Xbox, which anyone can have access to for a monthly subscription fee. A malicious user could subscribe in order to have access to the collection of games, then make copies of the games intending to keep them after cancelling the subscription. The same user will not be able to make another copy of a game that also runs on Xbox.

Remark 122. *Following our work, other weakenings of copy-protection (similar to SSL) have been studied. Security guarantee against a pirate that intents to prepare two copies, one that can be maliciously evaluated and one that has to be honestly evaluated, was studied in [BJL+21]. In this context, SSL can be seen as having a security guarantee when both copies have to be evaluated honestly. In [ALL+20], the*

[5] This denotes tracing out the second register.

[6] For instance, they could be engineers assigned to test whether the beta version contains bugs.

notion of copy-detection was defined using the framework of projective implementations from [Zha20]. While the syntax/definition is slightly different, copy-detection is similar to SSL with infinite-term security.

7.1.1 Construction overview

While we construct SSL with infinite-term security, in this overview we will use the syntax of finite-term lessor security. Our ideas can be easily adapted to the infinite-term lessor security.

To construct a SSL scheme in the setup model ($\mathsf{Setup}, \mathsf{Gen}, \mathsf{Lessor}, \mathsf{Run}, \mathsf{Check}$) against arbitrary quantum poly-time (QPT) pirates, we first focus on two weaker class of adversaries, namely, *duplicators* and *maulers*. Duplicators are adversaries who, given ρ_C generated by the lessor for a circuit C sampled from a distribution $\mathcal{D}_{\mathcal{C}}$, produce $\rho_C^{\otimes 2}$; that is, all they do is replicate the state. Maulers, who given ρ_C, output $\rho_C \otimes \rho_C^*$, where ρ_C^* is far from ρ_C in trace distance and ρ_C is the copy returned by the mauler back to the lessor; that is the second copy it produces is a modified version of the original copy.

While our construction is secure against arbitrary pirates, it will be helpful to first focus on these restricted type of adversaries. We propose two schemes: the first scheme is secure against QPT maulers and the second scheme against QPT duplicators. Once we discuss these schemes, we will then show how to combine the techniques from these two schemes to obtain a construction secure against arbitrary pirates.

SSL against Maulers. To protect SSL against a mauler, we attempt to construct a scheme using only classical cryptographic techniques. The reason why it could be possible to construct such a scheme is because maulers never produce a pirated copy ρ_C^* that is the same as the original copy ρ_C.

A natural attempt to construct a SSL scheme is to use virtual black-box obfuscation [BGI+01] (VBB): this is a compiler that transforms a circuit C into another functionally equivalent circuit $\widetilde{C}$ such that $\widetilde{C}$ only leaks the input-output behavior of C and nothing more. This is a powerful notion and implies almost all known cryptographic primitives. We generate the leased state ρ_C to be the VBB obfuscation of C, namely $\widetilde{C}$. The hope is that a mauler will not output another leased state ρ_C^* that is different from $\widetilde{C}$.

Unfortunately, this scheme is insecure. A mauler on input $\widetilde{C}$, obfuscates $\widetilde{C}$ once more to obtain $\widetilde{\widetilde{C}}$ and outputs this re-obfsuscated circuit. Moreover, note that the resulting re-obfuscated circuit still computes C. This suggests that program obfuscation is insufficient for our purpose. In hindsight, this should be unsurprising: VBB guarantees that given an obfuscated circuit, an efficient adversary should not learn anything about the implementation of the circuit, but this doesn't prevent the adversary from being able to re-produce modified copies of the obfuscated circuit.

To rectify this issue, we devise the following strategy:

- Instead of VBB, we start with a different obfuscation scheme that has the following property: given an obfuscated circuit $\widetilde{C}$, where C corresponds to an evasive function, it is computationally infeasible to determine an accepting input for C.

- We then combine this with a special proof system that guarantees the property: suppose an adversary, upon receiving $\widetilde{C}$ and a proof, outputs a *different* but functionally equivalent obfuscated circuit $\widetilde{C}^*$ along with a new proof. Then we can extract an accepting input for $\widetilde{C}$ from the adversary's proof. But this would contradict the above bullet and hence, it follows that its computationally infeasible for the adversary to output a different circuit $\widetilde{C}^*$.

To realize the above strategy, we need two separate cryptographic tools, that we define below.

Input-Hiding Obfuscators [BBC+14]: We recall the notion of input-hiding obfuscators [BBC+14]. An input-hiding obfuscator guarantees that given an obfuscated circuit $\widetilde{C}$, any efficient adversary cannot find an accepting input x, i.e., an input x such that $\widetilde{C}(x) = 1$. Of course this notion is only meaningful for an evasive class of functions: a function is evasive if given oracle access to this function, any efficient adversary cannot output an accepting point. The work of Barak, Bitansky, Canetti, Kalai, Paneth, and Sahai [BBC+14] proposed candidates for input-hiding obfuscators.

Simulation-Extractable NIZKs [Sah99, DSDCO+01]: Another primitive we consider is simulation-extractable non-interactive zero-knowledge [Sah99, DSDCO+01] (seNIZKs). A seNIZK system is a non-interactive protocol between a prover and a verifier with the prover trying to convince the verifier that a statement belongs to the NP language. By non-interactive we mean that the prover only sends one message to the verifier and the verifier is supposed to output the decision bit: accept or reject. Moreover, this primitive is defined in the common reference string model. In this model, there is a trusted setup that produces a common reference string and both the prover and the verifier have access to this common reference string.

As in a traditional interactive protocol, we require a seNIZK to satisfy the completeness property. Another property we require is simulation-extractability. Simulation-extractability, a property that implies both zero-knowledge and soundness, guarantees that if there exists an efficient adversary $\mathcal{A}$ who upon receiving a *simulated* proof[7] for an instance x, produces an accepting proof for a different instance x', i.e., $x' \neq x$, then there also exists an adversary $\mathcal{B}$ that given the same simulated proof produces an accepting proof for x' along with simultaneously producing a valid witness for x'.

[7] A simulated proof is one that is generated by an efficient algorithm, called a simulator, who has access to some private coins that was used to generate the common reference string. Moreover, a simulated proof is indistinguishable from an honestly generated proof. A simulator has the capability to generate simulated proofs for YES instances even without knowing the corresponding witness for these instances.

Combining Simulation-Extractable NIZKs and Input-Hiding Obfuscators: We now combine the two tools we introduced above to obtain a SSL scheme secure against maulers. Our SSL scheme will be associated with searchable circuits; given a description of a searchable circuit C, an input x can be determined efficiently such that $C(x) = 1$.

To lease a circuit C, do the following:

- Compute an input-hiding obfuscation of C, denoted by $\widetilde{C}$,
- Produce a seNIZK proof π that proves knowledge of an input x such that $C(x) = 1$. Note that we can find this input using the searchability property.

Output $(\widetilde{C}, \pi)$ as the leased circuit. To evaluate on any input x, we first check if π is a valid proof and if so, we compute $\widetilde{C}$ on x to obtain $C(x)$.

To see why this scheme is secure against maulers, suppose an adversary $\mathcal{A}$ given $(\widetilde{C}, \pi)$ produces $(\widetilde{C}^*, \pi^*)$, where $\widetilde{C}^* \neq \widetilde{C}$. Since $\mathcal{A}$ is a valid mauler we are guaranteed that $\widetilde{C}^*$ is functionally equivalent to C. We first run the seNIZK simulator to simulate π and once this is done, we no longer need x to generate π. Now, we invoke the simulation-extractability property to convert $\mathcal{A}$ into one who not only produces $(\widetilde{C}^*, \pi^*)$ but also simultaneously produces x such that $\widetilde{C}^*(x) = 1$. Since $\widetilde{C}^*$ is functionally equivalent to C, it follows that $C(x) = 1$ as well. But this violates the input-hiding property which says that no efficient adversary given $\widetilde{C}$ can produce an accepting input.

Issue: Checking Functional Equivalence. There is a subtlety we skipped in the proof above. The maulers that we consider have multi-bit output which is atypical in the cryptographic setting where the focus is mainly on boolean adversaries. This causes an issue when we switch from the honestly generated proof to a simulated proof. Upon receiving the honestly generated proof, $\mathcal{A}$ outputs $(\widetilde{C}^*, \pi^*)$ such that $\widetilde{C}^*$ is functionally equivalent to C but upon receiving the simulated proof, the adversary outputs $(\widetilde{C}^*, \pi^*)$ where $\widetilde{C}^*$ differs from C on one point. From $\mathcal{A}$, we need to extract one bit that would help distinguish the real and simulated proofs. To extract this bit, we rely upon sub-exponential security. Given $\widetilde{C}^*$, we run in time 2^n, where n is the input length, and check if $\widetilde{C}^*$ is still functionally equivalent to C; if indeed $\widetilde{C}^*$ is not functionally equivalent to C then we know for a fact that the adversary was given a simulated proof, otherwise it received an honestly generated proof. We set the security parameter in the seNIZK system to be sufficiently large (for eg, poly(n)) such that the seNIZK is still secure against adversaries running in time 2^n.

SSL against Duplicators. Next we focus on constructing SSL secure against duplicators. If our only goal was to protect against duplicators, we could achieve this with a simple scheme. The lessor, in order to lease C, will output $(|\psi\rangle, C)$ where $|\psi\rangle$ is a random quantum state generated by applying a random polynomial sized quantum circuit U on input $|0^{\otimes\lambda}\rangle$. Run on input $(|\psi\rangle, C, x)$ ignores the quantum state $|\psi\rangle$, and outputs $C(x)$. By quantum no-cloning, an attacker cannot output two copies of $(|\psi\rangle, C)$, which means that this scheme is already secure against duplicators.

Recall that we focused on designing SSL for duplicators in the hope that it will be later helpful for designing SSL for arbitrary pirates. But any SSL scheme in which Run ignores the quantum part would not be useful for obtaining SSL secure against arbitrary pirates; an attacker can simply replace the quantum state as part of the leased state with its own quantum state and copy the classical part. To overcome this insufficiency, we need to design SSL schemes where the Run algorithm only computes correctly when the input leased state belongs to a sparse set of quantum states. This suggests that the Run algorithm implicitly satisfies a verifiability property; it should be able to verify that the input quantum state lies in this sparse set.

Publicly Verifiable Unclonable States. We wish to construct a family of efficiently preparable states $\{|\psi_s\rangle\}_s$ with the following verifiability property. For any state $|\psi_s\rangle$ in the family, there is a way to sample a classical description d_s for $|\psi_s\rangle$ in such a way that it can be verified that d_s is a corresponding description of $|\psi_s\rangle$. To be more precise, there should be a verification algorithm $\mathsf{Ver}(|\psi_s\rangle, d)$ that accepts if d is a valid description for $|\psi_s\rangle$. Furthermore, we want the guarantee that given a valid pair $(|\psi_s\rangle, d_s)$, no QPT adversary can produce $|\psi_s\rangle^{\otimes 2}$.

Our requirement has the same flavor as public-key quantum money, but a key difference is that we do not require any secret parameters associated with the scheme. Moreover, we allow anyone to be able to generate such tuples $(|\psi_s\rangle, d_s)$ and not just the minting authority (bank).

Given such verifiable family, we can define the Run algorithm as follows,

$\mathsf{Run}(C, (|\psi_s\rangle, d), x)$:

- If $\mathsf{Ver}(|\psi_s\rangle, d) = 0$, output $\bot$.
- Otherwise, output $C(x)$.

Any lessor can now lease a state $(|\psi_s\rangle, d_s, C)$, which would allow anyone to compute C using Run. Of course, any pirate that is given $(|\psi_s\rangle, d_s, C)$ can prepare their own $(|\psi_{s'}\rangle, d_{s'})$ and then input $(|\psi_{s'}\rangle, d_{s'}, C)$ into Run. But recall that we are only interested in ruling out *duplicators*. From the public verifiable property of the quantum states, we have the fact that no QPT pirate could prepare $|\psi_s\rangle^{\otimes 2}$ from $(|\psi_s\rangle, d_s)$ and thus, it is computationally infeasible to duplicate the leased state.

Publicly Verifiable Unclonable States from Subspace Hiding Obfuscation. The notion of publicly verifiable unclonable states was first constructed without oracles by Zhandry [Zha19]. Zhandry's idea is to instantiate the quantum money scheme from hidden subspace [AC12] by introducing a type of obfuscation called subspace hiding obfuscation. Roughly speaking, a subspace hiding obfuscator (shO) takes as input a description of a linear subspace A, and outputs a circuit that computes the membership function for A, i.e. $\mathsf{shO}(A)(x) = 1$ iff $x \in A$. Zhandry shows that for a uniformly random $\frac{\lambda}{2}$-dimensional subspace $A \subset \mathbb{Z}_q^\lambda$, given $|A\rangle := \frac{1}{\sqrt{q^{\lambda/2}}} \sum_{a \in A} |a\rangle$ along with $\widetilde{g} \leftarrow \mathsf{shO}(A), \widetilde{g_\perp} \leftarrow \mathsf{shO}(A^\perp)$, no QPT algorithm can prepare $|A\rangle^{\otimes 2}$ with non-negligible

probability. Nevertheless, because $\widetilde{g}$ and $\widetilde{g_\perp}$ compute membership for A and $A^\perp$ respectively, it is possible to project onto $|A\rangle\langle A|$ using $(\widetilde{g}, \widetilde{g_\perp})$. This lets anyone check the tuple $(|\psi\rangle, (\widetilde{g}, \widetilde{g_\perp}))$ by measuring $|\psi\rangle$ with the projectors $\{|A\rangle\langle A|, I - |A\rangle\langle A|\}$.

Main Template: SSL against Pirates. Our goal is to construct SSL against arbitrary QPT pirates and not just duplicators or maulers. To achieve this goal, we combine the techniques we have developed so far.

To lease a circuit C, do the following:

1. First prepare the state the state $|A\rangle = \frac{1}{\sqrt{q^{\lambda/2}}} \sum_{a \in A} |a\rangle$, along with $\tilde{g} \leftarrow \mathsf{shO}(A)$ and $\widetilde{g_\perp} \leftarrow \mathsf{shO}(A^\perp)$.

2. Compute an input-hiding obfuscation of C, namely $\widetilde{C}$.

3. Let x be an accepting point of C. This can be determined using the searchability condition.

4. Compute a seNIZK proof π such that: (1) the obfuscations $(\widetilde{g}, \widetilde{g_\perp}, \widetilde{C})$ were computed correctly, as a function of $(A, A^\perp, C)$, and, (2) $C(x) = 1$.

5. Output $|\psi_C\rangle = (|A\rangle, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi)$.

The Run algorithm on input $(|\psi_C\rangle, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi)$ and x, first checks the proof π, and outputs $\perp$ if it does not accept the proof. If it accepts the proof, it knows that $\widetilde{g}$ and $\widetilde{g_\perp}$ are subspace obfuscators for some subspaces A and $A^\perp$ respectively; it can use them to project $|\psi_C\rangle$ onto $|A\rangle\langle A|$. This checks whether $|\psi_C\rangle$ is the same as $|A\rangle$ or not. If it is not, then it outputs $\perp$. If it has not output $\perp$ so far, then it computes $\widetilde{C}$ on x to obtain $C(x)$.

Proof Intuition: To prove the lessor security of the above scheme, we consider two cases depending on the behavior of the pirate:

- *Duplicator*: in this case, the pirate produces a new copy that is of the form $(\sigma^*, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi)$; that is, it has the same classical part as before. If σ^* is close to $|A\rangle\langle A|$, it would violate the no-cloning theorem. On the other hand, if σ^* is far from $|A\rangle\langle A|$, we can argue that the execution of Run on the copy produced by the pirate will not compute C. The reason being that at least one of the two subspace obfuscators $\widetilde{g}, \widetilde{g_\perp}$ will output $\perp$ on the state σ^*.

- *Mauler*: suppose the pirate produces a new copy that is of the form $(\sigma^*, \widetilde{g}^*, \widetilde{g_\perp}^*, \widetilde{C}^*, \pi^*)$ such that $(\widetilde{g}^*, \widetilde{g_\perp}^*, \widetilde{C}^*) \neq (\widetilde{g}, \widetilde{g_\perp}, \widetilde{C})$. We invoke the simulation-extractability property to find an input x such that $\widetilde{C}^*(x) = 1$. Since $\widetilde{C}^*$ is assumed to have the same functionality as C, this means that $C(x) = 1$. This would contradict the security of input-hiding obfuscation, since any QPT adversary even given $\widetilde{C}$ should not be able to find an accepting input x such that $C(x) = 1$.

7.2 Definition

We present the definition of secure software leasing schemes. A **secure software leasing (SSL) scheme** for a class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ consists of the following QPT algorithms.

- **Private-key Generation,** $\mathsf{Gen}(1^\lambda)$: On input security parameter λ, outputs a private key sk.

- **Software Lessor,** $\mathsf{Lessor}(\mathsf{sk}, C)$: On input the private key sk and a poly(n)-sized classical circuit $C \in \mathcal{C}_\lambda$, with input length n and output length m, outputs a quantum state ρ_C.

- **Evaluation,** $\mathsf{Run}(\rho_C, x)$: On input the quantum state ρ_C and an input $x \in \{0,1\}^n$, outputs y, and some state $\rho'_{C,x}$.

- **Check of Returned Software,** $\mathsf{Check}(\mathsf{sk}, \rho^*_C)$: On input the private key sk and the state ρ^*_C, it checks if ρ^*_C is a valid leased state and if so it outputs 1, else it outputs 0.

Setup. In this work, we only consider SSL schemes in the setup model. In this model, all the lessors in the world have access to a common reference string generated using a PPT algorithm Setup. The difference between Setup and Gen is that Setup is run by a trusted third party whose output is used by all the lessors while Gen is executed by each lessor separately. We note that our impossibility result rules out SSL schemes for all quantum unlearnable class of circuits even in the setup model.

We define this notion below.

Definition 123 (SSL with Setup)**.** *A secure software leasing scheme* $(\mathsf{Gen}, \mathsf{Lessor}, \mathsf{Run}, \mathsf{Check})$ *is said to be in the common reference string (CRS) model if additionally, it has an algorithm* Setup *that on input* 1^λ *outputs a string* crs.

Moreover, the algorithm Gen *now takes as input* crs *instead of* 1^λ *and* Run *additionally takes as input* crs.

We require that a SSL scheme, in the setup model, satisfies the following properties.

Definition 124 (Correctness)**.** *A SSL scheme* $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Lessor}, \mathsf{Run}, \mathsf{Check})$ *for* $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *is* ε**-correct** *if for all* $C \in \mathcal{C}_\lambda$*, with input length* n*, the following two properties holds for some negligible function* ε*:*

- **Correctness of Run***:*

$$\Pr\left[\forall x \in \{0,1\}^n,\ y = C(x) \ : \begin{array}{c} \mathsf{crs}\leftarrow\mathsf{Setup}\left(1^\lambda\right), \\ \mathsf{sk}\leftarrow\mathsf{Gen}(\mathsf{crs}), \\ \rho_C\leftarrow\mathsf{Lessor}(\mathsf{sk},C) \\ \left(\rho'_{C,x},y\right)\leftarrow\mathsf{Run}(\mathsf{crs},\rho_C,x) \end{array}\right] \geq 1-\varepsilon$$

- **Correctness of Check***:*

$$\Pr\left[\mathsf{Check}\left(\mathsf{sk},\rho_C\right)=1 \;:\; \begin{array}{c}\mathsf{crs}\leftarrow\mathsf{Setup}\left(1^\lambda\right),\\ \mathsf{sk}\leftarrow\mathsf{Gen}(\mathsf{crs})\\ \rho_C\leftarrow\mathsf{Lessor}(\mathsf{sk},C)\end{array}\right]\geq 1-\varepsilon$$

Reusability. A desirable property of a SSL scheme is reusability: the lessee should be able to repeatedly execute Run on multiple inputs. A SSL scheme does not necessarily guarantee reusability; for instance, Run could destroy the state after executing it just once. But fortunately, we can transform this scheme into another scheme that satisfies reusability.

We define reusability formally.

Definition 125. *(Reusability) A SSL scheme* $(\mathsf{Setup},\mathsf{Gen},\mathsf{Lessor},\mathsf{Run},\mathsf{Check})$ *for* $\mathcal{C}=\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *is said to be reusable if for all* $C\in\mathcal{C}$ *and for all* $x\in\{0,1\}^n$*,*

$$\left\|\rho'_{C,x}-\rho_C\right\|_{tr}\leq negl(\lambda).$$

Note that the above requirement $\left\|\rho'_{C,x}-\rho_C\right\|_{\mathrm{tr}}\leq \mathrm{negl}(\lambda)$ would guarantee that an evaluator can evaluate the leased state on multiple inputs; on each input, the original leased state is only disturbed a little which means that the resulting state can be reused for evaluation on other inputs.

The following proposition states that any SSL scheme can be converted into one that is reusable.

Proposition 126. *Let* $(\mathsf{Setup},\mathsf{Gen},\mathsf{Lessor},\mathsf{Run},\mathsf{Check})$ *be any SSL scheme (not necessarily satisfying the reusability condition). Then, there is a QPT algorithm* Run' *such that* $(\mathsf{Setup},\mathsf{Gen},\mathsf{Lessor},\mathsf{Run}',\mathsf{Check})$ *is a reusable SSL scheme.*

Proof. For any $C\in\mathcal{C}$ and for any $x\in\{0,1\}^n$, we have that $\mathsf{Run}(\mathsf{crs},\rho_C,x)$ outputs $C(x)$ with probability $1-\varepsilon$. By the Almost As Good As New Lemma, there is a way to implement Run such that it is possible to obtain $C(x)$, and then recover a state $\widetilde{\rho_C}$ satisfying $\|\widetilde{\rho_C}-\rho_C\|_{\mathrm{tr}}\leq\sqrt{\varepsilon}$. We let Run' be this operation. □

Thus, it suffices to just focus on the correctness property when constructing a SSL scheme.

7.2.1 Security

Our notion intends to capture the different scenarios discussed in the introduction. In particular, we want to capture the security guarantee that given an authorized (valid) copy ρ_C, no pirate can output two authorized copies. We will assume that these valid copies contain a quantum state and a classical string. The Run algorithm expects valid copies to have this form; without loss of generality, the classical part can always be measured before executing Run.

Finite-Term Lessor Security

We require the following security guarantee: suppose a QPT adversary (pirate) receives a leased copy of C generated using Lessor; denote this by ρ_C. We require that the pirate cannot produce a bipartite state σ^* on registers R_1 and R_2, such that $\sigma_1^* := \mathrm{Tr}_2[\sigma^*]$ passes the verification by Check, and the resulting *post-measurement* state on R_2, which we denote by $P_2(\sigma^*)$, still computes C by $\mathsf{Run}(P_2(\sigma^*), x) = C(x)$.

Before formally stating the definition, let us fix some notation. We will use the following notation for the state that the pirate keeps after the initial copy has been returned and verified. If the pirate outputs the bipartite state σ^*, then we will write

$$P_2(\mathsf{sk}, \sigma^*) \propto \mathrm{Tr}_1 \left[\Pi_1[\mathsf{Check}(\mathsf{sk}, \cdot)_1 \otimes I_2 (\sigma^*)]\right]$$

for the state that the pirate keeps *after* the first register has been returned and verified. Here, Π_1 denotes projecting the output of Check onto 1, and where $\mathsf{Check}(\mathsf{sk}, \cdot)_1 \otimes I_2(\sigma^*)$ denotes applying the Check QPT onto the first register, and the identity on the second register of σ^*. In other words, $P_2(\mathsf{sk}, \sigma^*)$ is used to denote the post-measurement state on R_2 conditioned on $\mathsf{Check}(\mathsf{sk}, \cdot)$ accepting on R_1.

Definition 127 (Finite-Term Perfect Lessor Security)**.** *We say that a SSL scheme* $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Lessor}, \mathsf{Run}, \mathsf{Check})$ *for a class of circuits* $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *is said to satisfy* $(\beta, \gamma, \mathcal{D}_\mathcal{C})$**-perfect finite-term lessor security**, *with respect to a distribution* $\mathcal{D}_\mathcal{C}$ *on* $\mathcal{C}$*, if for every QPT adversary* $\mathcal{A}$ *(pirate) that outputs a bipartite (possibly entangled) quantum state on two registers,* R_1 *and* R_2*, the following holds:*

$$\Pr\left[\begin{array}{c} \mathsf{Check}\left(\mathsf{sk}, \sigma_1^*\right)=1 \\ \wedge \\ \forall x,\ \Pr[\mathsf{Run}(\mathsf{crs}, P_2(\mathsf{sk}, \sigma^*), x) = C(x)] \geq \beta \end{array} : \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}\left(1^\lambda\right), \\ C \leftarrow \mathcal{D}_\mathcal{C}(\lambda), \\ \mathsf{sk} \leftarrow \mathsf{Gen}(\mathsf{crs}), \\ \rho_C \leftarrow \mathsf{Lessor}(\mathsf{sk}, C), \\ \sigma^* \leftarrow \mathcal{A}(\mathsf{crs}, \rho_C) \\ \sigma_1^* = Tr_2[\sigma^*] \end{array}\right] \leq \gamma$$

Remark 128. *The reason why we use the word perfect here is because we require* $\mathsf{Run}(P_2(\sigma^*), x) = C(x)$ *to hold with probability at least* β *on* every *input* x*. Note that* Run *is not necessarily deterministic (for instance, it could perform measurements) and thus we allow it to output the incorrect value with some probability.*

7.2.2 Infinite-Term Lessor Security

In the infinite-term lease case, we want the following security notion: given (σ_1^*, σ_2^*) generated by a pirate $\mathcal{A}(\rho_C)$, guarantees that if one copy satisfies the correctness,

$$\forall x \Pr[\mathsf{Run}(\mathsf{crs}, \sigma_1^*, x) = C(x)] \geq \beta$$

for some non-negligible β, then after successfully evaluating $C(x)$ using σ_1^* on any input x^*, it should be the case that the resulting state on the second register, which

we will denote by $\mathcal{E}^{(2)}_{x^*}(\sigma^*)$, cannot also satisfy

$$\forall x \Pr[\mathsf{Run}(\mathsf{crs}, \mathcal{E}^{(2)}_{x^*}(\sigma^*), x) = C(x)] \geq \beta.$$

In other words, if one of the copies has already been succesful in computing C in Run, then there will be inputs in which the second copy cannot evaluate C with better than negligible probability.

This security notion would rule out the following scenario. Eve gets a copy of ρ_C and gives σ^*_1 to Alice and σ^*_2 to Bob. Alice now chooses an input x_A, and Bob an input x_B. It cannot be the case that for all inputs (x_A, x_B) they choose, they will compute $(C(x_A), C(x_B))$ with non-negligible probability.

Definition 129 (Infinite-term Perfect Lessor Security)**.** *We say that a SSL scheme* $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Lessor}, \mathsf{Run}, \mathsf{Check})$ *for a class of circuits* $\mathcal{C} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ *is said to be* $(\gamma, \beta, \mathcal{D}_\mathcal{C})$-***infinite-term perfect lessor secure****, with respect to a distribution* $\mathcal{D}_\mathcal{C}$*, if for every QPT adversary* $\mathcal{A}$ *(pirate) that outputs a bipartite (possibly entangled) quantum state on two registers,* R_1 *and* R_2*, the following holds:*

$$\mathsf{Pr}\left[\begin{array}{c}\forall x, \left(\begin{array}{c}\mathsf{Pr}\left[(\mathsf{Run}(\mathsf{crs},x,\sigma^*_1)=C(x)\right]\geq\beta \\ \wedge \\ \forall x', \mathsf{Pr}\left[\mathsf{Run}(\mathsf{crs},x',\mathcal{E}^{(2)}_x(\sigma^*))=C(x')\right]\geq\beta\end{array}\right) \\ : \\ \begin{array}{c}\mathsf{crs}\leftarrow\mathsf{Setup}(1^\lambda), \\ C\leftarrow\mathcal{D}_\mathcal{C}(\lambda), \\ \mathsf{sk}\leftarrow\mathsf{Gen}(\mathsf{crs}), \\ \rho_C\leftarrow\mathsf{Lessor}(\mathsf{sk},C), \\ \sigma^*\leftarrow\mathcal{A}(\mathsf{crs},\rho_C) \\ \sigma^*_1 = Tr_2[\sigma^*]\end{array}\end{array}\right] \leq \gamma.$$

Remark 130. *Both finite and infinite-term security can be extended to the case where the pirate is given multiple copies,* $\rho_C^{\otimes m}$*, where* ρ_C *is the output of* Lessor *on* C*. In the finite-term case, we require the following: if a pirate outputs* $m+1$ *copies and moreover, the* m *initial copies are returned and succesfully checked, computing* Run *on the remaining copy (that the pirate did not return) will not be functionally equivalent to the circuit* C*. In the infinite-term case, the pirate cannot output* $m+1$ *copies where* Run *on each of the* $m+1$ *copies can be used to successfully compute* C*.*

7.3 Impossibility of SSL

Before presenting our SSL construction, we show how the impossibility of copy-protection from Chapter 6 extends to SSL. In the following theorem we will show that if every circuit $C \in \mathcal{C}$ have a unique representation in $\mathcal{C}$, then it is also not possible to have SSL for this circuit class. To see why we need an additional condition, lets consider a QPT pirate $\mathcal{A}$ that wants to break SSL given (Run, ρ_C) computing $C \in \mathcal{C}$. Then, $\mathcal{A}$ can run $\mathcal{B}$ to obtain a circuit $C' \in \mathcal{C}$, but in the proccess it could have destroyed ρ_C, hence it wouldn't be able to return the initial copy. If $\mathcal{B}$ takes as input (Run, ρ_C) and outputs a *fixed* C' with probability neglibly close to 1, then by the Almost As Good As New Lemma, it could uncompute and recover ρ_C. The definition of de-quantumizable class does not guarantee that $\mathcal{B}$ will output a fixed circuit C', unless each circuit in the family has a unique representation in $\mathcal{C}$. If each circuit has

a unique representation, the pirate would obtain $C' = C$ with probability negligibly close to 1, and uncompute to recover ρ_C. At this point, the pirate can generate its own leasing keys sk', and run $\mathsf{Lessor}(\mathsf{sk}', C')$ to obtain a valid leased state $\rho'_{C'}$. The pirate was able to generate a new valid leased state for C, while preserving the initial copy ρ_C, which it can later return to the lessor.

Theorem 131. *Let $(\mathcal{C}, \mathcal{D}_{\mathcal{C}})$ be a de-quantumizable class of circuits in which every circuit in the support of $\mathcal{D}_{\mathcal{C}}$ has a unique representation in $\mathcal{C}$. Then there is no SSL scheme $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Lessor}, \mathsf{Run}, \mathsf{Check})$ (in CRS model) for $\mathcal{C}$ satisfying ε-correctness and $(\beta, \gamma, \mathcal{D}_{\mathcal{C}})$-perfect finite-term lessor security for any negligible γ, and any $\beta \leq (1-\varepsilon)$.*

Proof. Consider the QPT algorithm $\mathcal{A}$ (pirate) that is given $\rho_C \leftarrow \mathsf{Lessor}(\mathsf{sk}, C)$ for some $C \leftarrow \mathcal{D}_{\mathcal{C}}$. The pirate will run $\mathcal{B}$, the QPT that de-quantumizes $(\mathcal{C}, \mathcal{D}_{\mathcal{C}})$, on input (Run, ρ_C) to obtain a functionally equivalent circuit $C' \in \mathcal{C}$. Because C has a unique representation in $\mathcal{C}$, we have $C' = C$. Since this succeeds with probability negligibly close to 1, by the Almost As Good As New Lemma, it can all be done in a way such that it is possible to obtain C and to recover a state $\widetilde{\rho_C}$ satisfying $\|\widetilde{\rho_C} - \rho_C\|_{\mathrm{tr}} \leq \mathrm{negl}(\lambda)$. At this point, the pirate generates its own key $\mathsf{sk}' \leftarrow \mathsf{Gen}(\mathsf{crs})$, and prepares $\rho'_C \leftarrow \mathsf{Lessor}(\mathsf{sk}', C)$. It outputs $\widetilde{\rho_C} \otimes \rho'_C$.

This means that ρ'_C is a valid leased state and by correctness of the SSL scheme,

$$\Pr\left[\forall x \in \{0,1\}^n,\ \mathsf{Run}\left(\mathsf{crs}, \rho'_C, x\right) = C(x) \ : \ \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}\left(1^\lambda\right), \\ \mathsf{sk}' \leftarrow \mathsf{Gen}(\mathsf{crs}), \\ \rho'_C \leftarrow \mathsf{Lessor}(\mathsf{sk}', C) \end{array}\right] \geq 1 - \varepsilon$$

Furthermore, since $\|\widetilde{\rho_C} - \rho_C\|_{\mathrm{tr}} \leq \mathrm{negl}(\lambda)$, the probability that $\widetilde{\rho_C}$ passes the return check is neglibly close to 1. Putting these together, we have

$$\mathsf{Pr}\left[\begin{array}{c} \mathsf{Check}(\mathsf{sk}, \widetilde{\rho_C}) = 1 \\ \wedge \\ \forall x,\ \Pr\left[\mathsf{Run}(\mathsf{crs}, \rho'_C, x) = C(x)\right] \geq 1 - \varepsilon \end{array} \ : \ \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}\left(1^\lambda\right), \\ C \leftarrow \mathcal{D}_{\mathcal{C}}(\lambda), \\ \mathsf{sk} \leftarrow \mathsf{Gen}(\mathsf{crs}), \\ \rho_C \leftarrow \mathsf{Lessor}(\mathsf{sk}, C), \\ \widetilde{\rho_C} \otimes \rho'_C \leftarrow \mathcal{A}(\mathsf{crs}, \rho_C) \end{array}\right] \geq 1 - \mathrm{negl}(\lambda)$$

□

7.4 Evasive circuits

The circuit class we consider in our construction of SSL is a subclass of evasive circuits. We recall the definition of evasive circuits below.

Evasive Circuits. Informally, a class of circuits is said to be evasive if a circuit drawn from a suitable distribution outputs 1 on a fixed point with negligible probability.

Definition 132 (Evasive Circuits)**.** *A class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$, associated with a distribution $\mathcal{D}_\mathcal{C}$, is said to be* **evasive** *if the following holds: for every $\lambda \in \mathbb{N}$, every $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$,*

$$\Pr_{C\leftarrow\mathcal{D}_\mathcal{C}}[C(x) = 1] \leq negl(\lambda),$$

Searchability. For our construction of SSL for $\mathcal{C}$, we crucially use the fact that given a circuit $C \in \mathcal{C}$, we can read off an input x from the description of C such that $C(x) = 1$. We formalize this by defining a search algorithm $\mathcal{S}$ that on input a circuit C outputs an accepting input for C. For many interesting class of functions, there do exist a corresponding efficiently implementable class of circuits associated with a search algorithm $\mathcal{S}$.

Definition 133 (Searchability)**.** *A class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ is said to be* **$\mathcal{S}$-searchable***, with respect to a PPT algorithm $\mathcal{S}$, if the following holds: on input C, $\mathcal{S}(C)$ outputs x such that $C(x) = 1$.*

In the next section, we will show how to construct SSL for searchable evasive circuits that have quantum input hiding obfuscators. An example of such a family of circuits is searchable compute-and-compare circuits. In Appendix B, we show that there is quantum secure input hiding obfuscators for them.

Compute-and-compare Circuits. A compute-and-compare circuit is of the following form: $\mathbf{C}[C, \alpha]$, where α is called a lock and C has output length $|\alpha|$, is defined as follows:

$$\mathbf{C}[C,\alpha](x) = \begin{cases} 1, & \text{if } C(x)=\alpha, \\ 0, & \text{otherwise} \end{cases}$$

Multi-bit compute-and-compare circuits. We can correspondingly define the notion of multi-bit compute-and-compare circuits. A multi-bit compute-and-compare circuit is of the following form:

$$\mathbf{C}[C,\alpha,\mathsf{msg}](x) = \begin{cases} \mathsf{msg}, & \text{if } C(x)=\alpha, \\ 0, & \text{otherwise} \end{cases},$$

where msg is a binary string.
We consider two types of distributions as defined by [WZ17].

Definition 134 (Distributions for Compute-and-Compare Circuits)**.** *We consider the following distributions on $\mathcal{C}_{\mathsf{cnc}}$:*

- *$\mathcal{D}_{\mathsf{unpred}}(\lambda)$: For any $(\mathbf{C}[C,\alpha])$ along with aux sampled from this unpredictable distribution, it holds that α is computationally unpredictable given (C, aux).*
- *$\mathcal{D}_{\mathsf{pseud}}(\lambda)$: For any $\mathbf{C}[C,\alpha]$ along with aux sampled from this distribution, it holds that $\mathbf{H}_{\mathsf{HILL}}(\alpha|(C,\mathsf{aux})) \geq \lambda^\varepsilon$, for some constant $\epsilon > 0$, where $\mathbf{H}_{\mathsf{HILL}}(\cdot)$ is the HILL entropy [HILL99].*

Note that with respect to the above distributions, the compute-and-compare class of circuits $\mathcal{C}_{\mathsf{cnc}}$ is evasive.

Searchable Compute-and-Compare Circuits: Examples. There are natural and interesting classes of searchable compute-and-compare circuits. For completeness, we state them below with additional examples [WZ17].

- Point circuits $C(\alpha, \cdot)$: the circuit $C(\alpha, \cdot)$ is a point circuit if it takes as input x and outputs $C(\alpha, x) = 1$ iff $x = \alpha$. If we define the class of point circuits suitably, we can find α directly from C_α; for instance, α can be the value assigned to the input wires of C.

- Conjunctions with wild cards $C(S, \alpha, \cdot)$: the circuit $C(S, \alpha, \cdot)$ is a conjunction with wild cards if it takes as input x and outputs $C(S, \alpha, x) = 1$ iff $y = \alpha$, where y is such that $y_i = x_i$ for all $i \in S$. Again, if we define this class of circuits suitably, we can find S and α directly from the description of $C(S, \alpha, \cdot)$. Once we find S and α, we can find the accepting input.

- Affine Tester: the circuit $C(\mathbf{A}, \alpha, \cdot)$ is an affine tester, with $\mathbf{A}, \mathbf{y}$ where $\mathbf{A}$ has a non-trivial kernel space, if it takes as input $\mathbf{x}$ and outputs $C(\mathbf{A}, \alpha, \mathbf{x}) = 1$ iff $\mathbf{A} \cdot \mathbf{x} = \alpha$. By reading off $\mathbf{A}$ and α and using Gaussian elimination we can find $\mathbf{x}$ such that $\mathbf{A} \cdot \mathbf{x} = \alpha$.

- Plaintext equality checker $C(\mathsf{sk}, \alpha, \cdot)$: the circuit $C(\mathsf{sk}, \alpha, \cdot)$, with hardwired values decryption key sk associated with a private key encryption scheme, message α, is a plaintext equality checker if it takes as input a ciphertext ct and outputs $C(\mathsf{sk}, \alpha, \mathsf{ct}) = 1$ iff the decryption of ct with respect to sk is α. By reading off α and sk, we can find a ciphertext such that ct is an encryption of α.

7.5 SSL for evasive circuits

In this section, we present the main construction of SSL satisfying infinite-term perfect lessor security.

Let $\mathcal{C} = \{\mathcal{C}_\lambda\}$ be the class of $\mathcal{S}$-searchable circuits associated with SSL. We denote $s(\lambda) = \mathrm{poly}(\lambda)$ to be the maximum size of all circuits in $\mathcal{C}_\lambda$. And let $\mathcal{D}_\mathcal{C}$ be the distribution associated with $\mathcal{C}$.

Tools.

- q-Input-hiding obfuscator $\mathsf{qIHO} = (\mathsf{qIHO.Obf}, \mathsf{qIHO.Eval})$ for $\mathcal{C}$ (Section 2.4.4).

- Subspace hiding obfuscation $\mathsf{shO} = (\mathsf{shO.Obf}, \mathsf{shO.Eval})$ (Section 2.4.4). The field associated with shO is $\mathbb{Z}_q$ and the dimensions will be clear below.

- q-simulation-extractable non-interactive zero-knowledge system $\mathsf{qseNIZK} = (\mathsf{CRSGen}, P, \mathcal{V})$ for NP (Section 2.4.7) with sub-exponential security as guaranteed in Lemma 48.

Construction. We describe the scheme of SSL below.
We describe the scheme of SSL below.

- $\mathsf{Setup}(1^\lambda)$: Compute $\mathsf{crs} \leftarrow \mathsf{CRSGen}\left(1^{\lambda_1}\right)$, where $\lambda_1 = \lambda + n$ and n is the input length of the circuit. Output crs.

- $\mathsf{Gen}(\mathsf{crs})$: On input common reference string crs, choose a random $\frac{\lambda}{2}$-dimensional subspace $A \subset \mathbb{Z}_q^\lambda$. Set $\mathsf{sk} = A$.

- $\mathsf{Lessor}(\mathsf{sk} = A, C)$: On input secret key sk, circuit $C \in \mathcal{C}_\lambda$, with input length n,

  1. Prepare the state $|A\rangle = \frac{1}{\sqrt{q^{\lambda/2}}} \sum_{a \in A} |a\rangle$.
  2. Compute $\widetilde{C} \leftarrow \mathsf{qIHO.Obf}(C; r_o)$
  3. Compute $\widetilde{g} \leftarrow \mathsf{shO}(A; r_A)$.
  4. Compute $\widetilde{g_\perp} \leftarrow \mathsf{shO}(A^\perp; r_{A^\perp})$.
  5. Let $x = \mathcal{S}(C)$; that is, x is an accepting point of C.
  6. Let L be the NP language defined by the following NP relation.

$$\mathcal{R}_L := \left\{ \left(\left(\widetilde{g}, \widetilde{g_\perp}, \widetilde{C}\right),\ (A, r_o, r_A, r_{A^\perp}, C, x) \right) \;\middle|\; \begin{smallmatrix} \widetilde{g}=\mathsf{shO}(A;r_A) \\ \widetilde{g_\perp}=\mathsf{shO}(A^\perp;r_{A^\perp}) \\ \widetilde{C}=\mathsf{qIHO.Obf}(C;r_o), \\ C(x)=1 \end{smallmatrix} \right\}.$$

  Compute $\pi \leftarrow P\left(\mathsf{crs}, \left(\widetilde{g}, \widetilde{g_\perp}, \widetilde{C}\right), (A, r_o, r_A, r_{A^\perp}, C, x)\right)$

  7. Output $\rho_C = |\Phi_C\rangle\langle\Phi_C| = \left(|A\rangle\langle A|, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi\right)$.

- $\mathsf{Run}(\mathsf{crs}, \rho_C, x)$:

  1. Parse ρ_C as $\left(\rho, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi\right)$. In particular, measure the last 4 registers.
     Note: This lets us assume that the input to those registers is just classical, since anyone about to perform Run *might as well measure those registers themselves.*
  2. We denote the operation $\mathsf{shO.Eval}(\widetilde{g}, |x\rangle|y\rangle) = |x\rangle|y \oplus \mathbb{1}_A(x)\rangle$ by $\widetilde{g}[|x\rangle|y\rangle]$, where $\mathbb{1}_A(x)$ is an indicator function that checks membership in A. Compute $\widetilde{g}[\rho \otimes |0\rangle\langle 0|]$ and measure the second register. Let a denote the outcome bit, and let ρ' be the post-measurement state.
  3. As above, we denote the operation $\mathsf{shO.Eval}(\widetilde{g_\perp}, |x\rangle|y\rangle) = |x\rangle|y \oplus \mathbb{1}_A(x)\rangle$ by $\widetilde{g_\perp}[|x\rangle|y\rangle]$. Compute $\widetilde{g_\perp}[\mathsf{FT}\rho'\mathsf{FT}^\dagger \otimes |0\rangle\langle 0|]$ and measure the second register. Let b denote the outcome bit.
     Note: in Step 2 and 3, Run *is projecting ρ onto $|A\rangle\langle A|$ if $a = 1$ and $b = 1$.*

4. Afterwards, perform the Fourier Transform again on the first register of the post-measurement state, let ρ'' be the resulting state.
5. Compute $c \leftarrow \mathcal{V}\left(\mathsf{crs}, \left(\widetilde{g}, \widetilde{g_\perp}, \widetilde{C}\right), \pi\right)$
6. If either $a = 0$ or $b = 0$ or $c = 0$, reject and output $\perp$.
7. Compute $y \leftarrow \mathsf{qIHO.Eval}\left(\widetilde{C}, x\right)$.
8. Output $\left(\rho'', \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi\right)$ and y.

- $\mathsf{Check}(\mathsf{sk} = A, \rho_C)$:
  1. Parse ρ_C as $\left(\rho, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi\right)$.
  2. Perform the measurement $\{|A\rangle\langle A|, I - |A\rangle\langle A|\}$ on ρ. If the measurement outcome corresponds to $|A\rangle\langle A|$, output 1. Otherwise, output 0.

Lemma 135 (Overwhelming probability of perfect correctness)**.** *The above scheme satisfies $\epsilon = negl(\lambda)$ correctness.*

Proof. We first argue that the correctness of Run holds. Since qIHO is perfectly correct, it suffices to show that Run will not output $\perp$. For this to happen, we need to show that $a, b, c = 1$. Since $\widetilde{g} = \mathsf{shO}(A)$, $\widetilde{g_\perp} = \mathsf{shO}(A^\perp)$, and the input state is $|A\rangle\langle A|$, then $a = 1$ and $b = 1$ with probability negligibly close to 1 by correctness of shO. If π is a correct proof, then by perfect correctness of $\mathsf{qseNIZK}$, we have that $\mathsf{Pr}[c = 1] = 1$.

To see that the correctness of Check also holds, note that the leased state is $\rho = |A\rangle\langle A|$, which will pass the check with probability 1.

□

Lemma 136. *Fix $\beta = \mu(\lambda)$, where $\mu(\lambda)$ is any non-negligible function. Assuming the security of $\mathsf{qIHO}, \mathsf{qseNIZK}$ and shO, the above scheme satisfies $(\beta, \gamma, \mathcal{D}_\mathcal{C})$-infinite-term perfect lessor security, where γ is a negligible function.*

Proof. For any QPT adversary $\mathcal{A}$, define the following event.

$\underline{\mathsf{Process}(1^\lambda)}$:

- $\mathsf{crs} \leftarrow \mathsf{Setup}\left(1^\lambda\right)$,
- $\mathsf{sk} \leftarrow \mathsf{Gen}(\mathsf{crs})$,
- $C \leftarrow \mathcal{D}_\mathcal{C}(\lambda)$,
- $\left(\rho_C = \left(|A\rangle\langle A|, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi\right)\right) \leftarrow \mathsf{Lessor}\left(\mathsf{sk}, r\right)$

- $\rho^* = \left(\widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}, \pi^{(1)}, \widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \pi^{(2)}, \sigma^*\right) \leftarrow \mathcal{A}\left(\mathsf{crs}, \rho_C\right)$

  That is, $\mathcal{A}$ outputs two copies; the classical part in the first copy is $\left(\widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}, \pi^{(1)}\right)$ and the classical part in the second copy is $\left(\widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \pi^{(2)}\right)$. Moreover, it outputs a single density matrix σ^* associated with two registers R_1 and R_2; the state in R_1 is associated with the first copy and the state in R_2 is associated with the second.

- $\sigma_1^* = \mathrm{Tr}_2[\sigma^*]$

- $\rho_C^{(1)} = \left(\sigma_1^*, \widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}, \pi^{(1)}\right) \bigwedge \rho_C^{(2)} = \left(\Pi_2(\sigma^*), \widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \pi^{(2)}\right)$ where

$$\Pi_2(\sigma^*) = \frac{\mathrm{Tr}_1\left[(\Pi_{(\widetilde{g^{(1)}}, \widetilde{g_\perp^{(1)}})} \otimes I)\sigma^*\right]}{\mathrm{Tr}\left[(\Pi_{(\widetilde{g^{(1)}}, \widetilde{g_\perp^{(1)}})} \otimes I)\sigma^*\right]}$$

  and where $\Pi_{(\widetilde{g^{(1)}}, \widetilde{g_\perp^{(1)}})}$ is the projection onto the subspace obfuscated by $(\widetilde{g^{(1)}}, \widetilde{g_\perp^{(1)}})$. In other words, $\Pi_2(\sigma^*)$ is the quantum state on register 2 conditioned on Run not outputting $\perp$ when applied to register 1.

To prove the lemma, we need to prove the following:

$$\mathsf{Pr}\left[\begin{array}{c} \forall x, \mathsf{Pr}\left[(\mathsf{Run}(\mathsf{crs}, x, \sigma_1^*) = C(x)\right] \geq \beta \\ \bigwedge \\ \forall x, x', \mathsf{Pr}\left[\mathsf{Run}(\mathsf{crs}, x', \mathcal{E}_x^{(2)}(\sigma^*)) = C(x')\right] \geq \beta \end{array} : \mathsf{Process}\left(1^\lambda\right)\right] \leq \gamma.$$

Note that for all x, $\mathcal{E}_x^{(2)}(\sigma^*) = \Pi_2(\sigma^*)$, since the only quantum operation that Run performs is projecting the first register of σ^* onto the subspace corresponding to $\widetilde{g}^{(1)}$. Consider the following:

- Define γ_1 as follows:

$$\mathsf{Pr}\left[\begin{array}{c} \forall x, \mathsf{Pr}\left[(\mathsf{Run}(\mathsf{crs}, x, \sigma_1^*) = C(x)\right] \geq \beta \\ \bigwedge \\ \forall x', \mathsf{Pr}[\mathsf{Run}(\mathsf{crs}, x', \Pi_2(\sigma^*)) = C(x')] \geq \beta \\ \bigwedge \\ \left(\widetilde{C}, \widetilde{g}, \widetilde{g_\perp}\right) = \left(\widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}\right) \\ \bigwedge \\ \left(\widetilde{C}, \widetilde{g}, \widetilde{g_\perp}\right) = \left(\widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}\right) \end{array} : \mathsf{Process}\left(1^\lambda\right)\right] = \gamma_1$$

- The other possible case is the case where at least one of the copies $\left(\widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}\right)$ or $\left(\widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}\right)$ is not equal to the corresponding resgisters of the original

copy. Without loss of generality, we will assume that the the second copy is not the same. Define γ_2 as follows:

$$\mathsf{Pr}\left[\begin{array}{c}\forall x,\mathsf{Pr}\left[(\mathsf{Run}(\mathsf{crs},x,\sigma_1^*)=C(x)\right]\geq\beta\\ \wedge\\ \forall x',\mathsf{Pr}[\mathsf{Run}(\mathsf{crs},x',\Pi_2(\sigma^*))=C(x')]\geq\beta\\ \wedge\\ \left(\widetilde{C},\widetilde{g},\widetilde{g_\perp}\right)\neq\left(\widetilde{C}^{(2)},\widetilde{g}^{(2)},\widetilde{g_\perp}^{(2)}\right)\end{array}\ :\ \mathsf{Process}\left(1^\lambda\right)\right]=\gamma_2$$

Note that $\gamma=\gamma_1+\gamma_2$. In the next two propositions, we prove that both γ_1 and γ_2 are negligible which will complete the proof of the lemma.

Proposition 137. $\gamma_1 \leq negl(\lambda)$

Proof. The run algorithm first projects σ^* into $|A\rangle^{\otimes 2}$, and outputs $\perp$ if σ^* is not $(|A\rangle\langle A|)^{\otimes 2}$. Suppose that $\langle A|\sigma_1^*|A\rangle$ is negligible, then Run will output $\perp$ on the first register with probability negligibly close to 1, and we would have γ_1 negligible as desired.

On the contrary, suppose that $\langle A|\sigma_1^*|A\rangle$ is non-negligible, and we have that

$$\Pi_2(\sigma^*)=\frac{\mathrm{Tr}_1\left[(|A\rangle\langle A|\otimes I)\sigma^*\right]}{\mathrm{Tr}\left[(|A\rangle\langle A|\otimes I)\sigma^*\right]}$$

i.e. the state in the second register after Run succesfully projects σ_1^* onto $|A\rangle\langle A|$.

We will prove the following claim, which implies that at least one of the two copies will output $\perp$ under Run with probability neglibly close to 1.

Claim 138. $\langle A|\Pi_2(\sigma^*)|A\rangle \leq negl(\lambda)$

Proof. Suppose not. Then, we can use $\mathcal{A}$ to break quantum no-cloning. Specifically, Zhandry [Zha19] showed that no QPT algorithm on input $(|A\rangle, \widetilde{g} := \mathsf{shO}(A), \widetilde{g_\perp} := \mathsf{shO}(A^\perp))$ can prepare the state $|A\rangle^{\otimes 2}$ with non-negligible probability. We will show that $\mathcal{A}$ allows us to do exactly this if $\langle A|\Pi_2(\sigma^*)|A\rangle$ is non-negligible.

Consider the following adversary $\mathcal{B}'$. It runs $\mathcal{A}$ and then projects the output of $\mathcal{A}$ onto $(|A\rangle\langle A|)^{\otimes 2}$; the output of the projection is the output of $\mathcal{B}'$.

$\mathcal{B}'(C)$**:**

1. Compute $\mathsf{crs}, \mathsf{sk}$ as in the construction
2. Compute $\rho_C \leftarrow \mathsf{Lessor}(\mathsf{sk}, C)$. Let $\rho_C = \left(|A\rangle\langle A|, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi\right)$.
3. Compute $\mathcal{A}(\mathsf{crs}, \rho_C)$ to obtain $\left(\widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}, \pi^{(1)}, \widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \pi^{(2)}, \sigma^*\right)$.
4. Then, project σ^* onto $(|A\rangle\langle A|)^{\otimes 2}$ by using $\widetilde{g}$ and $\widetilde{g_\perp}$. Let m be the outcome of this projection, so $m=1$ means that the post measured state is $(|A\rangle\langle A|)^{\otimes 2}$.
5. Output m.

The projection $(|A\rangle\langle A|)^{\otimes 2}$ can be done by first projecting the first register onto $|A\rangle\langle A|$ and then the second register. Conditioned on the first register not outputting $\perp$, means that σ_1^* is succesfully projected onto $|A\rangle\langle A|$. By our assumption that $\langle A|\sigma_1^*|A\rangle$ is non-negligible, this will happen with non-negligible probability. Conditioned on this being the case, if $\langle A|\Pi_2(\sigma^*)|A\rangle$ is non-negligible, then projecting the second register onto $|A\rangle\langle A|$ will also succeed with non-negligible probability. This means that $m = 1$ with non-negligible probability.

Consider the following adversary. It follows the same steps as $\mathcal{B}'$ except in preparing the states $|A\rangle$ and computing obfuscations $\widetilde{g}$, $\widetilde{g_\perp}$; it gets these quantities as input. Moreover, it simulates the proof π instead of computing the proof using the honest prover. This is because unlike $\mathcal{B}'$, the adversary $\mathcal{B}$ does not have the randomness used in computing $\widetilde{g}$ and $\widetilde{g_\perp}$ and hence cannot compute the proof π honestly.

$\mathcal{B}(|A\rangle, \widetilde{g}, \widetilde{g_\perp})$:

1. Sample randomness r_o and compute $\tilde{C} \leftarrow \mathsf{qIHO.Obf}(C; r_o)$.
2. Let FkGen and Sim be associated with the simulation-extractability propety of $\mathsf{qseNIZK}$. Compute $(\widetilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{FkGen}(1^\lambda)$.
3. Compute $(\pi, \mathsf{st}) \leftarrow \mathsf{Sim}\left(\widetilde{\mathsf{crs}}, \mathsf{td}, \left(\widetilde{g}, \widetilde{g_\perp}, \widetilde{C}\right)\right)$
4. Let $\rho_C = (|A\rangle\langle A|, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi)$
5. Run $\mathcal{A}(\widetilde{\mathsf{crs}}, \rho_C)$ to obtain $\left(\widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}, \pi^{(1)}, \widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \pi^{(2)}, \widetilde{\sigma^*}\right)$.
6. Then, project $\widetilde{\sigma^*}$ onto $(|A\rangle\langle A|)^{\otimes 2}$ by using $\widetilde{g}$ and $\widetilde{g_\perp}$. Let m be the outcome of this projection, so $m = 1$ means that the post measured state is $(|A\rangle\langle A|)^{\otimes 2}$.
7. Output m.

Note that from the q-simulation-extractability property[8] of $\mathsf{qseNIZK}$, it follows that the probability that $\mathcal{B}$ outputs 1 is negligibly close to the probability that $\mathcal{B}'$ outputs 1 because everything else is sampled from the same distribution. This implies that $\mathcal{B}$ on input $(|A\rangle, \widetilde{g}, \widetilde{g_\perp})$ outputs $|A\rangle^{\otimes 2}$ with non-negligible probability, contradicting [Zha19]. □

At this point, we want to show that if $\left(\widetilde{g^{(2)}}, \widetilde{g_\perp^{(2)}}\right) = (\widetilde{g}, \widetilde{g_\perp})$, and $\langle A|\Pi_2(\sigma^*)|A\rangle \leq \mathrm{negl}(\lambda)$, then the probability that $\mathsf{Run}(\mathsf{crs}, \Pi_2(\sigma^*), x)$ evaluates C correctly is negligible.

By correctness of shO, we have

$$\Pr[\forall x\ \widetilde{g^{(2)}}(x) = \mathbb{1}_A(x)] \geq 1 - \mathrm{negl}(\lambda)$$

[8] We don't need the full-fledged capability of q-simulation-extractability to argue this part; we only need q-zero-knowledge property which is implied by q-simulation-extractability.

$$\mathsf{Pr}[\forall x\ \widetilde{g_{\perp}^{(2)}}(x) = \mathbb{1}_{A^{\perp}}(x)] \geq 1 - \mathrm{negl}(\lambda)$$

This means that with probability negligibly close to 1, the first thing that the Run algorithm does on input $\rho_C^{(2)} = (\Pi_2(\sigma^*), \widetilde{g^{(2)}}, \widetilde{g_{\perp}^{(2)}}, \widetilde{C}, \pi)$ is to measure $\{|A\rangle\langle A|, I - |A\rangle\langle A|\}$ on $\Pi_2(\sigma^*)$. If $I - |A\rangle\langle A|$ is obtained, then the Run algorithm will output $\perp$. By Claim 138, the probability that this happens is negligibly close to 1. Formally, when $\widetilde{g}$ and $\widetilde{g_{\perp}}$ are subspace obfuscations of A and $A^{\perp}$ respectively, the check $a = 1$ and $b = 1$ performed by the Run algorithm is a projection onto $|A\rangle\langle A|$.

$$\begin{aligned}
\mathsf{Pr}[a = 1, b = 1] &= \mathrm{Tr}[\mathsf{FT}^{\dagger}\Pi_{A^{\perp}}\mathsf{FT}\Pi_A\Pi_2(\sigma^*)] \\
&= \mathrm{Tr}[|A\rangle\langle A|\Pi_2(\sigma^*)] \\
&= \langle A|\Pi_2(\sigma^*)|A\rangle \\
&\leq \mathrm{negl}(\lambda)
\end{aligned}$$

where $\Pi_A = \sum_{a \in A} |a\rangle\langle a|$ and $\Pi_{A^{\perp}} = \sum_{a \in A^{\perp}} |a\rangle\langle a|$. From this, we have that $\mathsf{Pr}[\mathsf{Run}(\mathsf{crs}, \rho_C^{(2)}, x) = \perp] \geq 1 - \mathrm{negl}(\lambda)$, and we have $\mathsf{Pr}[\mathsf{Run}(\mathsf{crs}, \rho_C^{(2)}, x) = C(x)] \leq \mathrm{negl}(\lambda)$ with probability negligibly close to 1.

This finishes our proof that if β is non-negligible, then $\gamma_1 \leq \mathrm{negl}(\lambda)$. □

Proposition 139. $\gamma_2 \leq \mathit{negl}(\lambda)$.

Proof. We consider the following hybrid process.

$\underline{\mathsf{HybProcess}_1(1^{\lambda})}$:

- $(\widetilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{FkGen}\left(1^{\lambda}\right)$,
- $\mathsf{sk} \leftarrow \mathsf{Gen}(\mathsf{crs})$,
- $C \leftarrow \mathcal{D}_{\mathcal{C}}(\lambda)$,
- Sample a random $\frac{\lambda}{2}$-dimensionall sub-space $A \subset \mathbb{Z}_q^{\lambda}$. Prepare the state $|A\rangle = \frac{1}{\sqrt{q^{\lambda/2}}} \sum_{a \in A} |a\rangle$.
- Compute $\widetilde{g} \leftarrow \mathsf{shO}\left(A; r_A\right)$,
- Compute $\widetilde{g_{\perp}} \leftarrow \mathsf{shO}\left(A^{\perp}; r_{A^{\perp}}\right)$,
- Compute $\widetilde{C} \leftarrow \mathsf{qIHO.Obf}(C; r_o)$
- $(\pi, \mathsf{st}) \leftarrow \mathsf{Sim}_1\left(\mathsf{crs}, \mathsf{td}, \left(\widetilde{g}, \widetilde{g_{\perp}}, \widetilde{C}\right)\right)$
- Set $\rho_C = \left(|A\rangle\langle A|, \widetilde{g}, \widetilde{g_{\perp}}, \widetilde{C}, \pi\right)$.
- $\left(\widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_{\perp}}^{(1)}, \pi^{(1)}, \widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_{\perp}}^{(2)}, \pi^{(2)}, \sigma^*\right) \leftarrow \mathcal{A}\left(\mathsf{crs}, \rho_C\right)$

- Set $\sigma_1^* = \mathrm{Tr}_2[\sigma^*]$
- Set $\rho_C^{(1)} = \left(\sigma_1^*, \widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}, \pi^{(1)}\right)$ and $\rho_C^{(2)} = \left(\Pi_2(\sigma^*), \widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \pi^{(2)}\right)$
- $\left(A^*, r_o^*, r_A^*, r_{A^\perp}^*, C^*, x^*\right) \leftarrow \mathsf{Sim}_2\left(\mathsf{st}, \left(\widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \widetilde{C}^{(2)}\right), \pi^{(2)}\right)$.

The proof of the following claim follows from the q-simulation-extractactability property of $\mathsf{qseNIZK}$.

Claim 140. *Assuming that* $\mathsf{qseNIZK}$ *satisfies q-simulation extractability property secure against QPT adversaries running in time* 2^n*, we have:*

$$\mathsf{Pr}\left[\begin{array}{c} \left(\left(\widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \widetilde{C}^{(2)}\right),\ \left(A^*, r_o^*, r_A^*, r_{A^\perp}^*, C^*, x^*\right)\right) \in \mathcal{R}(L) \\ \wedge \\ \forall x',\ \mathsf{Pr}\left[\mathsf{Run}\left(\mathsf{crs}, \rho^{(2)}, x'\right) = C(x')\right] \geq \beta \\ \wedge \\ \left(\widetilde{C}, \widetilde{g}, \widetilde{g_\perp}\right) \neq \left(\widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}\right) \end{array} : \mathsf{HybProcess}_1\left(1^\lambda\right)\right] = \delta_1$$

Then, $|\delta_1 - \gamma_2| \leq negl(\lambda)$.

Remark 141. *Note that* n *is smaller than the length of the NP instance and thus, we can invoke the sub-exponential security of the seNIZK system guaranteed in Lemma 48.*

Proof of Claim 140. Consider the following $\mathsf{qseNIZK}$ adversary $\mathcal{B}$:

- It gets as input crs.
- It samples and computes $(C, A, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C})$ as described in $\mathsf{HybProcess}_1(1^\lambda)$. It sends the following instance-witness pair to the challenger of seNIZK:

$$\left(\left(C, A, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}\right),\ \left((A, r_o, r_A, r_{A^\perp}, C, x)\right),\right.$$

  where $r_o, r_A, r_{A^\perp}$ is, respectively, the randomness used to compute obfuscations $\widetilde{g}$, $\widetilde{g_\perp}$ and $\widetilde{C}$.
- The challenger returns back π.
- $\mathcal{B}$ then sends $\left(|A\rangle, \widetilde{g}, \widetilde{g_\perp}, \widetilde{C}, \pi\right)$ to $\mathcal{A}$.
- $\mathcal{A}$ then outputs $\left(\widetilde{C}^{(1)}, \widetilde{g}^{(1)}, \widetilde{g_\perp}^{(1)}, \pi^{(1)}, \widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}, \pi^{(2)}, \sigma^*\right)$.
- $\mathcal{B}$ sets $\sigma_1^* = \mathrm{Tr}_2[\sigma^*]$.
- Finally, $\mathcal{B}$ performs the following checks:
  - *Verify if the classical parts are different*: Check if $\left(\widetilde{C}, \widetilde{g}, \widetilde{g_\perp}\right) = \left(\widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\perp}^{(2)}\right)$; if so output $\perp$, otherwise continue.

– *Verify if second copy computes C*: If the measurement above does not output $\bot$, set $\rho_C^{(2)} = \left(\Pi_2(\sigma^*), \widetilde{C}^{(2)}, \widetilde{g}^{(2)}, \widetilde{g_\bot}^{(2)}, \pi^{(2)}\right)$. For every x, check if $\widetilde{C}^{(2)}(x) = C(x)$. If for any x, the check fails, output $\bot$. *// Note that this step takes time* $2^{O(n+\log(n))}$.

- Output $\left(\left(\widetilde{g}^{(2)}, \widetilde{g_\bot}^{(2)}, \widetilde{C}^{(2)}\right), \pi^{(2)}\right)$.

Note that $\mathcal{B}$ is a valid $\mathsf{qseNIZK}$ adversary: it produces a proof on an instance different from one for which it obtained a proof (either real or simulated) and moreover, the proof produced by $\mathcal{B}$ (conditioned on not $\bot$) is an accepting proof.

If $\mathcal{B}$ gets as input honest CRS and honestly generated proof π then this corresponds to $\mathsf{Process}_1(1^\lambda)$ and if $\mathcal{B}$ gets as input simulated CRS and simulated proof π then this corresponds to $\mathsf{HybProcess}_1(1^\lambda)$.

Thus, from the security of q-simulation-extractable NIZKs, we have that $|\gamma_2 - \delta_1| \leq \mathrm{negl}(\lambda)$. □

We first prove the following claim.

Claim 142.

$$\left(\left(\left(\widetilde{g}^{(2)}, \widetilde{g_\bot}^{(2)}, \widetilde{C}^{(2)}\right),\ (A^*, r_o^*, r_A^*, r_{A^\bot}^*, C^*, x^*)\right) \in \mathcal{R}(L) \bigwedge \forall x,\ \mathsf{Pr}\left[\mathsf{Run}\left(\mathsf{crs}, \rho_C^{(2)}, x\right) = C(x)\right] \geq \beta\right)$$
$$\implies C(x^*) = 1,$$

Proof. We first claim that $\forall x,\ \mathsf{Pr}\left[\mathsf{Run}\left(\mathsf{crs}, \rho_C^{(2)}, x\right) = C(x)\right] \geq \beta$ implies that $\widetilde{C}^{(2)} \equiv C$, where $\equiv$ denotes functional equivalence. Suppose not. Let x' be an input such that $\widetilde{C}^{(2)}(x') \neq C(x')$ then this means that $\mathsf{Run}(\mathsf{crs}, \rho_C^{(2)}, x')$ *always* outputs a value different from $C(x')$; follows from the description of Run. This means that $\mathsf{Pr}[\mathsf{Run}\left(\mathsf{crs}, \rho_C^{(2)}, x'\right) = C(x')] = 0$, contradicting the hypothesis.

Moreover, $\left(\left(\widetilde{g}^{(2)}, \widetilde{g_\bot}^{(2)}, \widetilde{C}^{(2)}\right),\ \left(A^*, r_o^*, r_A^*, r_{A^\bot}^*, C^*, x^*\right)\right) \in \mathcal{R}(L)$ implies that $\widetilde{C}^{(2)} = \mathsf{qIHO}(1^\lambda, C^*; r_o^*)$ and $C^*(x^*) = 1$. Furthermore, perfect correctness of qIHO implies that $\widetilde{C}^{(2)} \equiv C^*$.

So far we have concluded that $\widetilde{C}^{(2)} \equiv C$, $\widetilde{C}^{(2)} \equiv C^*$ and $C^*(x^*) = 1$. Combining all of them together, we have $C(x^*) = 1$.

□

Consider the following inequalities.

$$\delta_1 = \mathsf{Pr}\left[\begin{array}{c}\left(\left(\widetilde{g}^{(2)},\widetilde{g_\perp}^{(2)},\widetilde{C}^{(2)}\right),\ \left(A^*,r_o^*,r_A^*,r_{A^\perp}^*,C^*,x^*\right)\right)\in\mathcal{R}(L)\\ \wedge\\ \forall x',\ \mathsf{Pr}\left[\mathsf{Run}\left(\mathsf{crs},\rho_C^{(2)},x'\right)=C(x')\right]\geq\beta\\ \wedge\\ \left(\widetilde{C},\widetilde{g},\widetilde{g_\perp}\right)\neq\left(\widetilde{C}^{(2)},\widetilde{g}^{(2)},\widetilde{g_\perp}^{(2)}\right)\end{array} : \mathsf{HybProcess}_1\right]$$

$$= \mathsf{Pr}\left[\begin{array}{c}C(x^*)=1\\ \wedge\\ \left(\widetilde{C},\widetilde{g},\widetilde{g_\perp}\right)\neq\left(\widetilde{C}^{(2)},\widetilde{g}^{(2)},\widetilde{g_\perp}^{(2)}\right)\end{array} : \mathsf{HybProcess}_1\right]$$

$$\leq \mathsf{Pr}\left[C\left(x^*\right)=1\ :\ \mathsf{HybProcess}_1\right]$$

Let $\mathsf{Pr}\left[C\left(x^*\right)=1\ :\ \mathsf{HybProcess}_1\right]=\delta_2$.

Claim 143. *Assuming the q-input-hiding property of* qIHO*, we have* $\delta_2 \leq negl(\lambda)$

Proof. Suppose δ_2 is not negligible. Then we construct a QPT adversary $\mathcal{B}$ that violates the q-input-hiding property of qIHO, thus arriving at a contradiction.

$\mathcal{B}$ now takes as input $\widetilde{C}$ (an input-hiding obfuscator of C), computes $(\widetilde{\mathsf{crs}},\mathsf{td})\leftarrow\mathsf{FkGen}\left(1^\lambda\right)$ and then computes $\rho_C=\left(|A\rangle,\widetilde{g},\widetilde{g_\perp},\widetilde{C},\pi\right)$ as computed in $\mathsf{HybProcess}_1$. It sends $(\widetilde{\mathsf{crs}},\rho_C)$ to $\mathcal{A}$ who responds with $\left(\widetilde{C}^{(1)},\widetilde{g}^{(1)},\widetilde{g_\perp}^{(1)},\pi^{(1)},\widetilde{C}^{(2)},\widetilde{g}^{(2)},\widetilde{g_\perp}^{(2)},\pi^{(2)},\sigma^*\right)$. Compute $\left(A^*,r_o^*,r_A^*,r_{A^\perp}^*,C^*,x^*\right)$ by generating $\mathsf{Sim}_2(\mathsf{st},(\widetilde{g}^{(2)},\widetilde{g_\perp}^{(2)},\widetilde{C}^{(2)}),\pi^{(2)})$, where st is as defined in $\mathsf{HybProcess}_1$. Output x^*.

Thus, $\mathcal{B}$ violates the q-input-hiding property of qIHO with probability δ_2 and thus δ_2 has to be negligible. □

Combining the above observations, we have that $\gamma_2 \leq \mathrm{negl}(\lambda)$ for some negligible function negl. This completes the proof.

□

□

Appendix A

Instantiation of qseNIZK

Before we prove Lemma 48, we first state the necessary preliminary background.

Definition 144 (q-Non-Interactive Zero-Knowledge)**.** *A non-interactive system* $(\mathsf{CRSGen}, P, \mathcal{V})$ *defined for a NP language* $\mathcal{L}$ *is said to be* **q-non-interactive zero-knowledge (qNIZK)** *if it satisfies Definition 45 and additionally, satisfies the following properties:*

- *Adaptive Soundness: For any malicious QPT prover* P^**, the following holds:*

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathsf{crs},x,\pi)\ \ accepts \\ \wedge \\ x'\notin\mathcal{L} \end{array} : \begin{array}{c} \mathsf{crs}\leftarrow\mathsf{CRSGen}\left(1^\lambda\right) \\ (x,\pi)\leftarrow P^*(\mathsf{crs}) \end{array}\right] \leq negl(\lambda)$$

- *Adaptive (Multi-Theorem) Zero-knowledge: For any QPT verifier* $\mathcal{V}^*$*, there exists two QPT algorithms* FkGen *and simulator* Sim*, such that the following holds:*

$$\left|\Pr\left[\begin{array}{c} 1\leftarrow\mathcal{V}^*\left(\mathsf{st},\{\pi\}_{i\in[q]}\right) \\ \wedge \\ \forall i\in[q],\ (x_i,w_i)\in\mathcal{R}(\mathcal{L}) \end{array} : \begin{array}{c} \mathsf{crs}\leftarrow\mathsf{CRSGen}(1^\lambda) \\ \left(\{(x_i,w_i)\}_{i\in[q]},\mathsf{st}\right)\leftarrow\mathcal{V}^*(\mathsf{crs}) \\ \forall i\in[q],\ \pi_i\leftarrow P(\mathsf{crs},x_i,w_i) \end{array}\right]\right.$$

$$\left.-\Pr\left[\begin{array}{c} 1\leftarrow\mathcal{V}^*\left(\mathsf{st},\{\pi\}_{i\in[q]}\right) \\ \wedge \\ \forall i\in[q],\ (x_i,w_i)\in\mathcal{R}(\mathcal{L}) \end{array} : \begin{array}{c} (\mathsf{crs},\mathsf{td})\leftarrow\mathsf{FkGen}(1^\lambda) \\ \left(\{(x_i,w_i)\}_{i\in[q]},\mathsf{st}\right)\leftarrow\mathcal{V}^*(\mathsf{crs}) \\ \{\pi_i\}_{i\in[q]}\leftarrow\mathsf{Sim}(\mathsf{crs},\mathsf{td},\{x_i\}_{i\in[q]}) \end{array}\right]\right| \leq negl(\lambda)$$

If both adaptive soundness and adaptive multi-theorem zero-knowledge holds against quantum adversaries running in time $2^{\tilde{O}(T)}$ *then we say that* $(\mathsf{CRSGen}, P, \mathcal{V})$ *is a* T*-sub-exponential qNIZK.*

Remark 145. *q-simulation-extractable NIZKs imply qNIZKs since simulation-extractability implies both soundness and zero-knowledge properties.*

Definition 146 (q-CCA2-secure PKE)**.** *A public-encryption scheme* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *(defined below) is said to satify* **q-CCA2-security** *if every QPT adversary* $\mathcal{A}$ *wins in* $\mathsf{Expt}_{\mathcal{A}}$ *(defined below) only with negligible probability.*

- $\mathsf{Setup}(1^\lambda)$: *On input security parameter* λ*, output a public key* pk *and a decryption key* sk*.*
- $\mathsf{Enc}(\mathsf{pk}, x)$: *On input public-key* pk*, message* x*, output a ciphertext* ct*.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: *On input decryption key* sk*, ciphertext* ct*, output* y*.*

For any $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$*, we have* $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, x)) = x$*.*

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

- *Challenger generates* $\mathsf{Setup}(1^\lambda)$ *to obtain* $(\mathsf{pk}, \mathsf{sk})$*. It sends* pk *to* $\mathcal{A}$*.*
- $\mathcal{A}$ *has (classical) access to a decryption oracle that on input* ct*, outputs* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$*. It can make polynomially many queries.*
- $\mathcal{A}$ *then submits* (x_0, x_1) *to the challenger which then returns* $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, x_b)$*.*
- $\mathcal{A}$ *is then given access to the same oracle as before. The only restriction on* $\mathcal{A}$ *is that it cannot query* ct^**.*
- *Output* b' *where the output of* $\mathcal{A}$ *is* b'*.*

$\mathcal{A}$ *wins in* $\mathsf{Expt}_{\mathcal{A}}$ *with probability* $\mu(\lambda)$ *if* $\mathsf{Pr}\left[b = b' \ : \ \substack{b \xleftarrow{\$} \{0,1\} \\ \mathsf{Expt}_{\mathcal{A}}(1^\lambda)}\right] = \frac{1}{2} + \mu(\lambda)$*.*

If the above q-CCA2 security holds against quantum adveraries running in time $2^{\tilde{O}(T)}$ *then we say that* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *is a* T*-sub-exponential q-CCA2-secure PKE scheme.*

Remark 147. *One could also consider the setting when the CCA2 adversary has superposition access to the oracle. However, for our construction, it suffices to consider the setting when the adversary only has classical access to the oracle.*

Consider the following lemma.

Lemma 148. *Consider a language* $\mathcal{L}_\ell \in NP$ *such that every* $x \in \mathcal{L}_\ell$ *is such that* $|x| = \ell$*.*

Under the ℓ*-sub-exponential QLWE assumption, there exists a q-simulation-extractable NIZKs for* $\mathcal{L}_\ell$ *satisfying perfect completeness.*

Proof. We first state the following proposition that shows how to generically construct a q-simulation-extractable NIZK from qNIZK and a CCA2-secure public-key encryption scheme.

Proposition 149. *Consider a language* $\mathcal{L}_\ell \in NP$ *such that every* $x \in \mathcal{L}_\ell$ *is such that* $|x| = \ell$*.*

Assuming ℓ*-sub-exponential qNIZKs for NP and* ℓ*-sub-exponential q-CCA2-secure PKE schemes, there exists a* ℓ*-sub-exponential qseNIZK system for* $\mathcal{L}_\ell$*.*

Proof. Let qPKE be a ℓ-sub-exponential qCCA2-secure PKE scheme. Let qNIZK be a ℓ-sub-exponential qNIZK for the following relation.

$$\mathcal{R}_{\mathsf{qNIZK}} = \left\{((\mathsf{pk}, \mathsf{ct}_w, x),\ (w, r_w)) \ :\ \left((x,w) \in \mathcal{R}(\mathcal{L}_\ell) \bigwedge\ \mathsf{ct}_w = \mathsf{Enc}(\mathsf{pk}, (x,w); r_w)\right)\right\}$$

We present the construction (quantum analogue of [Sah99, DSDCO+01]) of q-simulation-extractable NIZK for $\mathcal{L}_\ell$ below.

- $\mathsf{CRSGen}(1^\lambda)$: On input security parameter λ,
  - Compute $\mathsf{qNIZK.CRSGen}(1^{\lambda_1})$ to obtain $\mathsf{qNIZK.crs}$, where $\lambda_1 = \mathrm{poly}(\lambda, \ell)$ is chosen such that qNIZK is a ℓ-sub-exponential q-non-interactive zero-knowledge argument system.
  - Compute $\mathsf{qPKE.Setup}(1^{\lambda_2})$ to obtain $(\mathsf{pk}, \mathsf{sk})$, where $\lambda_2 = \mathrm{poly}(\lambda, \ell)$ is chosen such that qPKE is a ℓ-sub-exponential q-CCA2-secure PKE scheme.

  Output $\mathsf{crs} = (\mathsf{pk}, \mathsf{qNIZK.crs})$.
- $P(\mathsf{crs}, x, w)$: On input common reference string crs, instance x, witness w,
  - Parse crs as $(\mathsf{pk}, \mathsf{qNIZK.crs})$.
  - Compute $\mathsf{ct}_w \leftarrow \mathsf{qPKE.Enc}(\mathsf{pk}, (x,w); r_w)$, where $r_w \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$.
  - Compute $\mathsf{qNIZK}.\pi \leftarrow \mathsf{qNIZK}.P(\mathsf{qNIZK.crs}, (\mathsf{pk}, \mathsf{ct}_w, x), (w, r_w))$.

  Output $\pi = (\mathsf{qNIZK}.\pi, \mathsf{ct}_w)$.
- $\mathcal{V}(\mathsf{crs}, x, \pi)$: On input common reference string crs, NP instance x, proof π,
  - Parse crs as $(\mathsf{pk}, \mathsf{ct}, \mathsf{qNIZK.crs})$.
  - Output $\mathsf{qNIZK}.\mathcal{V}(\mathsf{qNIZK.crs}, (\mathsf{pk}, \mathsf{ct}_w, x), \pi)$.

We prove that the above argument system satisfies q-simulation-extractability. We describe the algorithms FkGen and $\mathsf{Sim} = (\mathsf{Sim}_1.\mathsf{Sim}_2)$ below. Let $\mathsf{qNIZK.FkGen}$ and $\mathsf{qNIZK.Sim}$ be the QPT algorithms associated with the zero-knowledge property of qNIZK.

$\underline{\mathsf{FkGen}(1^\lambda)}$: Compute $(\mathsf{qNIZK.crs}, \tau) \leftarrow \mathsf{qNIZK.FkGen}\left(1^\lambda\right)$. Compute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{qPKE.Setup}(1^\lambda)$. Output $\mathsf{crs} = (\mathsf{qNIZK.crs}, \mathsf{pk}, \mathsf{ct})$ and $\mathsf{td} = (\tau, \mathsf{sk})$.

$\underline{\mathsf{Sim}_1\left(\mathsf{crs}, \mathsf{td}, \{x_i\}_{i\in[q]}\right)}$: Compute $\mathsf{qNIZK.Sim}\left(\mathsf{qNIZK.crs}, \tau, (\mathsf{pk}, \mathsf{ct}, x_i)\right)$ to obtain $\mathsf{qNIZK}.\pi_i$, for every $i \in [q]$. Output $\{\mathsf{qNIZK}.\pi_1, \ldots, \mathsf{qNIZK}.\pi_q\}$ and $\mathsf{st} = \left(\mathsf{td}, \mathsf{crs}, \left(\{x_i\}_{i\in[q]}\right)\right)$.

$\underline{\mathsf{Sim}_2\left(\mathsf{st}, x', \pi'\right)}$: On input $\mathsf{st} = \left(\mathsf{td} = (\tau, \mathsf{sk}), \mathsf{crs}, \left(\{x_i\}_{i\in[q]}\right)\right)$, instance x', proof $\pi' = (\mathsf{qNIZK}.\pi', \mathsf{ct}'_w)$, compute $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}'_{w'})$ to obtain w'. Output w'.

Suppose $\mathcal{A}$ be a quantum adversary running in time $2^{\widetilde{O}(\ell)}$ such that the following holds:

$$\mathsf{Pr}\left[\begin{array}{c} \mathcal{V}(\mathsf{crs},x',\pi') \text{ accepts} \\ \wedge \\ (\forall i\in[q],\ (x_i,w_i)\in\mathcal{R}(\mathcal{L})) \\ \wedge \\ (\forall i\in[q],\ x'\neq x_i) \end{array} : \begin{array}{c} \mathsf{crs}\leftarrow\mathsf{CRSGen}(1^\lambda), \\ \left(\{(x_i,w_i)\}_{i\in[q]},\mathsf{st}_{\mathcal{A}}\right)\leftarrow\mathcal{A}_1(\mathsf{crs}) \\ \forall i\in[q],\ \pi_i\leftarrow P(\mathsf{crs},\mathsf{td},x_i) \\ (x',\pi')\leftarrow\mathcal{A}_2(\mathsf{st}_{\mathcal{A}},\pi_1,\ldots,\pi_q) \end{array}\right] = \varepsilon$$

Let δ be such that the following holds:

$$\mathsf{Pr}\left[\begin{array}{c} \mathcal{V}(\mathsf{crs},x',\pi') \text{ accepts} \\ \wedge \\ (\forall i\in[q],(x_i,w_i)\in\mathcal{R}(\mathcal{L})) \\ \wedge \\ (x',w')\in\mathcal{R}(L) \\ \wedge \\ (\forall i\in[q],\ x'\neq x_i) \end{array} : \begin{array}{c} (\mathsf{crs},\mathsf{td})\leftarrow\mathsf{FkGen}(1^\lambda), \\ \left(\{(x_i,w_i)\}_{i\in[q]},\mathsf{st}_{\mathcal{A}}\right)\leftarrow\mathcal{A}_1(\mathsf{crs}) \\ (\pi_1,\ldots,\pi_q,\mathsf{st}_{\mathsf{Sim}})\leftarrow\mathsf{Sim}_1\left(\mathsf{crs},\mathsf{td},\{x_i\}_{i\in[q]}\right) \\ (x',\pi')\leftarrow\mathcal{A}_2(\mathsf{st}_{\mathcal{A}},\pi_1,\ldots,\pi_q) \\ w'\leftarrow\mathsf{Sim}_2(\mathsf{st}_{\mathsf{Sim}},x',\pi') \end{array}\right] = \delta$$

We prove using a standard hybrid argument that $|\delta-\varepsilon|\leq \mathsf{negl}(\lambda)$.

$\underline{\mathsf{Hybrid}_1}$: $\mathcal{A}$ is given $\pi_1,\ldots,\pi_q$, where $\pi_i\leftarrow P(\mathsf{crs},x_i,w_i)$. Let (x',π') is the output of $\mathcal{A}$ and parse $\pi'=(\mathsf{qNIZK}.\pi',\mathsf{ct}'_w)$. Decrypt ct'_w using sk to obtain (x^*,w').

From the adaptive soundness of qNIZK, the probability that $(x',w')\in\mathcal{R}(\mathcal{L}_\ell)$ and $x^*=x'$ is negligibly close to ε.

$\underline{\mathsf{Hybrid}_2}$: $\mathcal{A}$ is given $\pi_1,\ldots,\pi_q$, where the proofs are generated as follows: first compute $(\mathsf{qNIZK}.\pi_1,\ldots,\mathsf{qNIZK}.\pi_q)\leftarrow\mathsf{qNIZK.Sim}(\mathsf{crs},\mathsf{td},\{x_i\}_{i\in[q]})$, where $(\mathsf{crs},\mathsf{td})\leftarrow\mathsf{qNIZK.FkGen}(1^\lambda)$. Then compute $\mathsf{ct}_{w_i}\leftarrow\mathsf{Enc}(\mathsf{pk},(x_i,w_i))$ for every $i\in[q]$. Set $\pi_i=(\mathsf{qNIZK}.\pi_i,\mathsf{ct}_{w_i})$. The rest of this hybrid is defined as in Hybrid_1.

From the adaptive zero-knowledge property of qNIZK, the probability that $(x',w')\in\mathcal{R}(\mathcal{L}_\ell)$ and $x^*=x'$ in the hybrid $\mathsf{Hybrid}_{2.j}$ is still negligibly close to ε.

$\underline{\mathsf{Hybrid}_3}$: This hybrid is defined similar to the previous hybrid except that $\mathsf{ct}_{w_i}\leftarrow\overline{\mathsf{Enc}(\mathsf{pk},0)}$, for every $i\in[q]$.

From the previous hybrids, it follows that $\mathsf{ct}'_w\neq\mathsf{ct}_{w_i}$, for all $i\in[q]$ with probability negligibly close to ε; this follows from the fact that qPKE is perfectly correct and the fact that $x^*=x'$ holds with probability negligibly close to ε. Thus, we can invoke q-CCA2-security of qPKE, the probability that $(x',w')\in\mathcal{R}(\mathcal{L}_\ell)$ is still negligibly close to ε.

But note that Hybrid_3 corresponds to the simulated experiment and thus we just showed that the probability that we can recover w' such that $(x',w')\in\mathcal{R}(\mathcal{L}_\ell)$ is negligibly close to ε.

□

The primitives in the above proposition can be instantiated from sub-exponential QLWE by starting with existing LWE-based constructions of the above primitive and

suitably setting the parameters of the underlying LWE assumption. We state the following propositions without proof.

Proposition 150 ([PS19])**.** *Assuming ℓ-sub-exponential QLWE (Section 2.3), there exists a ℓ-sub-exponential qNIZK for NP.*

Remark 151. *To be precise, the work of [PS19] constructs a NIZK system satisfying adaptive multi-theorem zero-knowledge and non-adaptive soundness. However, non-adaptive soundness implies adaptive soundness using complexity leveraging; the reduction incurs a security loss of 2^{ℓ}.*

Proposition 152 ([PW11])**.** *Assuming ℓ-sub-exponential QLWE (Section 2.3), there exists a ℓ-sub-exponential q-CCA2-secure PKE scheme.*

□

Appendix B

qIHO for compute-and-compare circuits

To complement the impossibility result, we present a construction of SSL for a subclass of evasive circuits. Specifically, the construction works for circuit classes that have q-Input-Hiding obfuscators. In the following section, we show that there are q-Input-Hiding obfuscators for Compute-and-Compare circuits.

Barak, Bitansky, Canetti, Kalai, Paneth, and Sahai [BBC+14] present a construction of input-hiding obfuscators secure against classical PPT adversaries; however, it is unclear whether their construction is secure against QPT adversaries. Instead we present a construction of input-hiding obfuscators (for a class of circuits different from the ones considered in [BBC+14]) from QLWE. Specifically, we show how to construct a q-input-hiding obfuscator for compute-and-compare circuits $\mathcal{C}_{\mathsf{cnc}}$ with respect to a distribution $\mathcal{D}_{\mathcal{C}}$ defined in Definition 134.

Lemma 153 (qIHO for Compute-and-Compare Circuits)**.** *Consider a class of compute-and-compare circuits $\mathcal{C}_{\mathsf{cnc}}$ associated with a distribution $\mathcal{D}_{\mathcal{C}}$ (Definition 134). Assuming QLWE, there exists qIHO for $\mathcal{C}_{\mathsf{cnc}}$.*

Proof. We prove this in two steps: we first construct a qIHO for the class of point functions and then we use this to build qIHO for compute-and-compare class of circuits.

qIHO for point functions: To prove this, we use a theorem due to [BBC+14] that states that an average-case VBB for circuits with only polynomially many accepting points is already an input-hiding obfuscator for the same class of circuits; their same proof also holds in the quantum setting. Any q-average-case VBB for circuits with only polynomially many accepting points is already a qIHO. As a special case, we have a qIHO for point functions from q-average-case VBB for point functions. Moreover, we can instantiate q-average-case VBB for point functions from QLWE and thus, we have qIHO for point functions from QLWE.

We describe the formal details below. First, we recall the definition of average-case VBB.

Definition 154 (q-Average-Case Virtual Black-Box Obfuscation (VBB))**.** *Consider a class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ associated with a distribution $\mathcal{D}_\mathcal{C}$. We say that* $(\mathsf{Obf}, \mathsf{Eval})$ *is said to be a* **q-average-case virtual black-box obfucsator** *for $\mathcal{C}$ if it holds that for every QPT adversary $\mathcal{A}$, there exists a QPT simulator* Sim *such that for every $\lambda \in \mathbb{N}$, the following holds for every non-uniform QPT distinguisher D:*

$$\left|\Pr\left[1 \leftarrow D\left(\widetilde{C}\right) \ : \ \begin{matrix} C\leftarrow\mathcal{D}_\mathcal{C}(\lambda), \\ \widetilde{C}\leftarrow\mathsf{Obf}\left(1^\lambda, C\right)\end{matrix}\right] - \Pr\left[1 \leftarrow D\left(\widetilde{C}\right) \ : \ \widetilde{C} \leftarrow \mathsf{Sim}\left(1^\lambda\right)\right]\right| \leq negl(\lambda),$$

We consider a quantum analogue of a proposition proven in [BBC+14]. We omit the proof details since this is identical to the proof provided by [BBC+14] albeit in the quantum setting.

Proposition 155. *Consider a class of evasive circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ associated with a distribution $\mathcal{D}_\mathcal{C}$ such that each circuit $C \in \mathcal{C}_\lambda$ has polynomially many accepting points.*

Assuming q-average-case virtual black-box obfuscation for $\mathcal{C}$, there is a qIHO for $\mathcal{C}$.

As a special case, we have qIHO for point functions (defined below) assuming q-average-case VBB for point functions. Moreover, q-average-case VBB for point functions can be instantiated from QLWE (see for example [WZ17, GKW17]). Thus, we have the following proposition.

Proposition 156 (q-Input-Hiding Obfuscator for Point Functions)**.** *Consider the class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ defined as follows: every circuit $C \in \mathcal{C}$, is associated with x such that it outputs 1 on x and 0 on all other points.*

Assuming QLWE, there is a qIHO for $\mathcal{C}$.

qIHO for compute-and-compare circuits from qIHO for point functions: We now show how to construct qIHO for compute-and-compare circuits $\mathcal{C}_{\mathsf{cnc}}$, associated with distribution $\mathcal{D}_{\mathsf{cnc}}$ (Definition 134), from qIHO for point functions. Denote $\mathsf{PO.qIHO}$ to be a qIHO for point functions $\mathcal{G} = \{\mathcal{G}_\lambda\}_{\lambda\in\mathbb{N}}$ associated with distribution $\mathcal{D}_{\mathsf{po}}$, where $\mathcal{D}_{\mathsf{po}}$ is a marginal distribution of $\mathcal{D}_{\mathsf{cnc}}$ on $\{\alpha\}$. We construct qIHO for compute-and-compare circuits below; we denote this by $\mathsf{cnc.qIHO}$.

$\mathsf{cnc.qIHO.Obf}\left(1^\lambda, \mathbf{C}[C,\alpha]\right)$: It takes as input security parameter λ, compute-and-compare circuit $\mathbf{C}[C,\alpha]$, associated with lock α. Compute $\mathsf{PO.qIHO}(1^\lambda, G_\alpha \in \mathcal{G}_\lambda)$ to obtain $\widetilde{G_\alpha}$. Output $\widetilde{\mathbf{C}} = \left(C, \widetilde{G_\alpha(\cdot)}\right)$.

$\mathsf{cnc.qIHO.Eval}\left(\widetilde{\mathbf{C}}, x\right)$: On input obfuscated circuit $\widetilde{\mathbf{C}} = \left(C, \widetilde{G_\alpha}\right)$, input x, do the following:

- Compute $C(x)$ to obtain α'.
- Compute $\mathsf{PO.Eval}\left(\widetilde{G_\alpha}, \alpha'\right)$ to obtain b.
- Output b.

Claim 157. *Assuming* $\mathsf{PO.qIHO}$ *is an input-hiding obfuscator for* $\mathcal{G}$ *associated with* $\mathcal{D}_{\mathsf{po}}$, $\mathsf{cnc.qIHO}$ *is an input-hiding obfuscator for* $\mathcal{C}$ *associated with* $\mathcal{D}_{\mathsf{cnc}}$.

Proof. Suppose there exists a QPT adversary $\mathcal{A}$ such that the following holds:

$$\left|\mathsf{Pr}\left[\widetilde{\mathbf{C}}(x)=1 \ : \ \begin{array}{c}\mathbf{C}[C,\alpha]\leftarrow\mathcal{D}_{\mathsf{cnc}}(\lambda),\\ \widetilde{\mathbf{C}}\leftarrow\mathsf{cnc.qIHO}(1^\lambda,\mathbf{C}[C,\alpha]),\\ x\leftarrow\mathcal{A}\left(1^\lambda,\widetilde{\mathbf{C}}\right)\end{array}\right]\right|=\delta$$

Our first observation is that $\mathsf{Pr}\left[C(x)=\alpha \mid \widetilde{C}(x)=1\right]=1$. Using this, we can construct another adversary $\mathcal{A}'$ that violates the input-hiding property of $\mathsf{PO.qIHO}$. On input $\widetilde{G_\alpha(\cdot)}$, $\mathcal{A}'$ computes $\mathcal{A}\left(\widetilde{\mathbf{C}}=\left(C,\widetilde{G_\alpha(\cdot)}\right)\right)$; denote the output to be x. Finally, $\mathcal{A}'$ outputs $\alpha'=C(x)$.

From the above observations, it holds that $\mathcal{A}'$ breaks the input-hiding property of $\mathsf{PO.qIHO}$ with probability δ. From the security of $\mathsf{PO.qIHO}$, we have that $\delta=\mathrm{negl}(\lambda)$ and thus the proof of the claim follows.

□

Conclusion: Combining Claim 157 and Proposition 156, we have qIHO for compute-and-compare circuits from QLWE.

□

Bibliography

[Aar] Scott Aaronson. Shtetl-Optimized. Ask Me Anything: Apocalypse Edition. https://www.scottaaronson.com/blog/?p=4684#comment-1834174. Comment #283, Posted: 03-24-2020, Accessed: 03-25-2020.

[Aar04] Scott Aaronson. Limitations of quantum advice and one-way communication. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 320–332. IEEE, 2004.

[Aar09] S. Aaronson. Quantum copy-protection and quantum money. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 229–242, 2009.

[ABDS20] Gorjan Alagic, Zvika Brakerski, Yfke Dulek, and Christian Schaffner. Impossibility of quantum virtual black-box obfuscation of classical circuits, 2020.

[ABG+20] Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-quantum multi-party computation in constant rounds. *arXiv preprint arXiv:2005.12904*, 2020.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 41–60, 2012.

[ACGH20] Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography*, pages 153–180, Cham, 2020. Springer International Publishing.

[ACLP21] Prabhanjan Ananth, Kai-Min Chung, and Rolando L. La Placa. On the concurrent composition of quantum zero-knowledge, 2021.

[AF16] Gorjan Alagic and Bill Fefferman. On quantum obfuscation. *arXiv preprint arXiv:1602.01771*, 2016.

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 255–268, 2020.

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

[AK21] Prabhanjan Ananth and Fatih Kaleoglu. Uncloneable encryption, revisited, 2021.

[ALL+20] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection, 2020.

[ALP20] Prabhanjan Ananth and Rolando L. La Placa. Secure quantum extraction protocols. In *Theory of Cryptography Conference*, 2020.

[ALP21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In *EUROCRYPT*. Springer-Verlag, 2021.

[Amb02] Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002.

[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.

[Bar01] 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.

[BB83] Charles H Bennett and Gilles Brassard. Quantum cryptography and its application to provably secure key expansion, public-key distribution, and coin-tossing. In *IEEE International Symposium on Information Theory*, volume 95. St-Jovite: Qebec Press, 1983.

[BB14] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, 2014. Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84.

[BBBW83] Charles H Bennett, Gilles Brassard, Seth Breidbart, and Stephen Wiesner. Quantum cryptography, or unforgeable subway tokens. In *Advances in Cryptology*, pages 267–275. Springer, 1983.

[BBC+14] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Obfuscation for evasive functions. In *Theory of Cryptography Conference*, pages 26–51. Springer, 2014.

[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.

[BCPR16] 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.

[BDS16] Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures. *arXiv preprint arXiv:1609.09047*, 2016.

[BG92] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Annual International Cryptology Conference*, pages 390–420. Springer, 1992.

[BG20] A. Broadbent and A. B. Grilo. Qma-hardness of consistency of local density matrices with applications to quantum zero-knowledge. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 196–205, Nov 2020.

[BGGL01] B. Barak, O. Goldreich, S. Goldwasser, and Y. Lindell. Resettably-sound zero-knowledge and its applications. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 116–125, 2001.

[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.

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In *Annual Cryptology Conference*, pages 344–360. Springer, 2013.

[BI19] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. *arXiv preprint arXiv:1910.03551*, 2019.

[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.

[BJL+21] Anne Broadbent, Stacey Jeffery, Sébastien Lord, Supartha Podder, and Aarthi Sundaram. Secure software leasing without assumptions, 2021.

[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.

[BKS21] Nir Bitansky, Michael Kellner, and Omri Shmueli. Post-quantum resettably-sound zero knowledge. Cryptology ePrint Archive, Report 2021/349, 2021. https://eprint.iacr.org/2021/349.

[BL04] Boaz Barak and Yehuda Lindell. Strict polynomial-time in simulation and extraction. *SIAM Journal on Computing*, 33(4):783–818, 2004.

[BL19] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via random oracles. *arXiv preprint arXiv:1903.00130*, 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.

[BP13] Nir Bitansky and Omer Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 241–250, 2013.

[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.

[Bra05] Gilles Brassard. Brief history of quantum cryptography: A personal perspective. In *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, 2005.*, pages 19–23. IEEE, 2005.

[Bra18] Zvika Brakerski. Quantum fhe (almost) as secure as classical. In *Annual International Cryptology Conference*, pages 67–95. Springer, 2018.

[BS05] Boaz Barak and Amit Sahai. How to play almost any mental game over the net-concurrent composition via super-polynomial simulation. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 543–552. IEEE, 2005.

[BS16] Anne Broadbent and Christian Schaffner. Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography*, 78(1):351–382, 2016.

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In *STOC*, 2020.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[CCLY21] Nai-Hui Chia, Kai-Min Chung, Qipeng Liu, and Takashi Yamakawa. On the impossibility of post-quantum black-box zero-knowledge in constant rounds, 2021.

[CCY20] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Black-box approach to post-quantum zero-knowledge in constant round. *arXiv preprint arXiv:2011.02670*, 2020.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In *Annual International Cryptology Conference*, pages 19–40. Springer, 2001.

[CHN+16] Aloni Cohen, Justin Holmgren, Ryo Nishimaki, Vinod Vaikuntanathan, and Daniel Wichs. Watermarking cryptographic capabilities. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 1115–1127, New York, NY, USA, 2016. Association for Computing Machinery.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 494–503, 2002.

[CLP15] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In *Annual Cryptology Conference*, pages 287–307. Springer, 2015.

[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.

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model, 2020.

[CVZ20] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive zero-knowledge arguments for qma, with preprocessing. In *Annual International Cryptology Conference*, pages 799–828. Springer, 2020.

[DCO99] Giovanni Di Crescenzo and Rafail Ostrovsky. On concurrent zero-knowledge with pre-processing. In *Annual International Cryptology Conference*, pages 485–502. Springer, 1999.

[DGH+20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from cdh or lpn. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 768–797. Springer, 2020.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *Journal of the ACM (JACM)*, 51(6):851–898, 2004.

[DS98] Cynthia Dwork and Amit Sahai. Concurrent zero-knowledge: Reducing the need for timing constraints. In *Annual International Cryptology Conference*, pages 442–457. Springer, 1998.

[DSDCO+01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *Annual International Cryptology Conference*, pages 566–598. Springer, 2001.

[FGH+12] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter Shor. Quantum money from knots. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 276–289, 2012.

[FKP19] Cody Freitag, Ilan Komargodski, and Rafael Pass. Non-uniformly sound certificates with applications to concurrent zero-knowledge. In *Annual International Cryptology Conference*, pages 98–127. Springer, 2019.

[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+05] Oded Goldreich et al. Foundations of cryptography–a primer. *Foundations and Trends® in Theoretical Computer Science*, 1(1):1–116, 2005.

[Gav12] Dmitry Gavinsky. Quantum money with classical verification. In *2012 IEEE 27th Conference on Computational Complexity*, pages 42–52. IEEE, 2012.

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

[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.

[GJJM20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 668–699. Springer, 2020.

[GJO+13] Vipul Goyal, Abhishek Jain, Rafail Ostrovsky, Silas Richelson, and Ivan Visconti. Concurrent zero knowledge in the bounded player model. In *Theory of Cryptography Conference*, pages 60–79. Springer, 2013.

[GK96a] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *J. Cryptology*, 9(3):167–190, 1996.

[GK96b] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 39–56, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[GKVW20] Rishab Goyal, Venkata Koppula, Satyanarayana Vusirikala, and Brent Waters. On perfect correctness in (lockable) obfuscation. 2020.

[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.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC*, pages 291–304, 1985.

[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.

[Gol07] Oded Goldreich. *Foundations of cryptography: volume 1, basic tools*. Cambridge university press, 2007.

[Gol09] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[Got03] Daniel Gottesman. Uncloneable encryption. *Quantum Information & Computation*, 3(6):581–602, 2003.

[GZ20] Marios Georgiou and Mark Zhandry. Unclonable decryption keys. Cryptology ePrint Archive, Report 2020/877, 2020. https://eprint.iacr.org/2020/877.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HSS11] Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In *Annual Cryptology Conference*, pages 411–428. Springer, 2011.

[JKMR06] 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.

[KKS18] Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 34–65. Springer, 2018.

[KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions, 2021.

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

[KW17] Sam Kim and David J. Wu. Watermarking cryptographic functionalities from standard lattice assumptions. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 503–536, Cham, 2017. Springer International Publishing.

[LAF+09] Andrew Lutomirski, Scott Aaronson, Edward Farhi, David Gosset, Avinatan Hassidim, Jonathan Kelner, and Peter Shor. Breaking and making quantum money: toward a new quantum cryptographic protocol. *arXiv preprint arXiv:0912.3825*, 2009.

[Lin03] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 683–692, 2003.

[Lin17] Yehuda Lindell. How to simulate it–a tutorial on the simulation proof technique. *Tutorials on the Foundations of Cryptography*, pages 277–346, 2017.

[LN11] Carolin Lunemann and Jesper Buus Nielsen. Fully simulatable quantum-secure coin-flipping and applications. In Abderrahmane Nitaj and David Pointcheval, editors, *Progress in Cryptology – AFRICACRYPT 2011*, pages 21–40, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[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.

[MST21] Christian Majenz, Christian Schaffner, and Mehrdad Tahmasbi. Limitations on uncloneable encryption and simultaneous one-way-to-hiding, 2021.

[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.

[MY21] Tomoyuki Morimae and Takashi Yamakawa. Classically verifiable (dual-mode) nizk for qma with preprocessing, 2021.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of cryptology*, 4(2):151–158, 1991.

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

[Pas04] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *STOC*, pages 232–241, 2004.

[Pei16] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

[PR03] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 404–413. IEEE, 2003.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *FOCS*, pages 366–375. IEEE, 2002.

[PRSD17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 461–473, 2017.

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

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for np from (plain) learning with errors. In *Annual International Cryptology Conference*, pages 89–114. Springer, 2019.

[PTV14] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent zero knowledge, revisited. *Journal of cryptology*, 27(1):45–66, 2014.

[PTW09] Rafael Pass, Wei-Lung Dustin Tseng, and Douglas Wikström. On the composition of public-coin zero-knowledge protocols. In *Annual International Cryptology Conference*, pages 160–176. Springer, 2009.

[PV08] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. On constant-round concurrent zero-knowledge. In *Theory of Cryptography Conference*, pages 553–570. Springer, 2008.

[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.

[PW11] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *SIAM Journal on Computing*, 40(6):1803–1844, 2011.

[Rab05] Michael O Rabin. How to exchange secrets with oblivious transfer. *IACR Cryptol. ePrint Arch.*, 2005(187), 2005.

[Reg] Oded Regev. The learning with errors problem.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 84–93, 2005.

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

[rev] How microsoft corporation makes most of its money. https://www.fool.com/investing/2017/06/29/how-microsoft-corporation-makes-most-of-its-money.aspx.

[RK99] Ransom Richardson and Joe Kilian. On the concurrent composition of zero-knowledge proofs. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 415–431. Springer, 1999.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 543–553. IEEE, 1999.

[Shm20] Omri Shmueli. Multi-theorem (malicious) designated-verifier nizk for qma, 2020.

[Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[Unr10] Dominique Unruh. Universally composable quantum multi-party computation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–505. Springer, 2010.

[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.

[VW16] Thomas Vidick and John Watrous. Quantum proofs. *Foundations and Trends in Theoretical Computer Science*, 11(1-2):1–215, 2016.

[VZ20] Thomas Vidick and Tina Zhang. Classical zero-knowledge arguments for quantum computations. *Quantum*, 4:266, 2020.

[Wat09] John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983.

[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.

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 408–438. Springer, 2019.

[Zha20] Mark Zhandry. Schrödinger's pirate: How to trace a quantum decoder. Cryptology ePrint Archive, Report 2020/1191, 2020. https://eprint.iacr.org/2020/1191.