Post-quantum Zero Knowledge in Constant Rounds*

Nir Bitansky† ‡ Omri Shmueli‡

Abstract

We construct a constant-round zero-knowledge classical argument for \( \text{NP} \) secure against quantum attacks. We assume the existence of Quantum Fully-Homomorphic Encryption and other standard primitives, known based on the Learning with Errors Assumption for quantum algorithms. As a corollary, we also obtain a constant-round zero-knowledge quantum argument for \( \text{QMA} \).

At the heart of our protocol is a new no-cloning non-black-box simulation technique.

---

*This work was supported in part by ISF grants 18/484 and 19/2137, by Len Blavatnik and the Blavatnik Family Foundation, and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482).

†Tel Aviv University, nirbitan@tau.ac.il. Member of the Check Point Institute of Information Security. Supported also by the Alon Young Faculty Fellowship.

‡Tel Aviv University, omrishmueli@mail.tau.ac.il. Supported also by the Zevulun Hammer Scholarship from the Council for Higher Education in Israel.
1 Introduction

Zero-knowledge protocols allow to prove statements without revealing anything but the mere fact that they are true. Since their introduction by Goldwasser, Micali, and Rackoff [GMR89] they have had a profound impact on modern cryptography and theoretical computer science at large. Following more than three decades of exploration, zero-knowledge protocols are now quite well understood in terms of their expressiveness and round complexity. In particular, under standard computational assumptions, arbitrary NP statements can be proved in only a constant number of rounds [GMW86, GK96a].

In this work, we consider classical zero-knowledge protocols with post-quantum security, namely, protocols that can be executed by classical parties, but where both soundness and zero knowledge are guaranteed even against efficient quantum adversaries. Here our understanding is far more restricted than in the classical setting. Indeed, not only are we faced with stronger adversaries, but also have to deal with the fact that quantum information behaves in a fundamentally different way than classical information, which summons new challenges in the design of zero-knowledge protocols.

In his seminal work [Wat09], Watrous developed a new quantum simulation technique and used it to show that classical zero-knowledge protocols for NP, such as the Goldreich-Micali-Wigderson 3-coloring protocol [GMW86], are also zero knowledge against quantum verifiers, assuming commitments with post-quantum hiding. These protocols are, in fact, proof systems meaning that soundness holds against unbounded adversarial provers, let alone efficient quantum ones. As in the classical setting, to guarantee a negligible soundness error (the gold standard in cryptography) these protocols require a polynomial number of rounds.

Watrous’ technique does not apply for classical constant-round protocols. In fact, constant-round zero-knowledge protocols with post-quantum security remains an open question, even when the honest parties and communication are allowed to be quantum. The gap between classical and quantum zero knowledge stems from fundamental aspects of quantum information such as the no-cloning theorem [WZ82] and quantum state disturbance [FP96]. These pose a substantial barrier for classical zero-knowledge simulation techniques, a barrier that has so far been circumvented only in specific settings (such as, [Wat09]). Overcoming these barriers in the context of constant-round zero-knowledge seems to require a new set of techniques.

1.1 Results

Under standard computational assumptions, we resolve the above open question — we construct a classical, post-quantumly secure, computational-zero-knowledge argument for NP in a constant number of rounds (with a negligible soundness error). That is, the honest verifier and prover (given a witness) are efficient classical algorithms. In terms of security, both zero-knowledge and soundness hold against polynomial-size quantum circuits with non-uniform quantum advice.

Our construction is based on fully-homomorphic encryption supporting the evaluation of quantum circuits (QFHE) as well as additional standard classical cryptographic primitives. All are required to be secure against efficient quantum algorithms with non-uniform quantum advice. QFHE was recently constructed [Mah18a, Bra18] based on the assumption that the Learning with Errors Problem [Reg09] is hard for the above class of algorithms (from hereon, called QLWE) and a circular security assumption (analogous to the assumptions required for multi-key FHE in the classical setting). All other required primitives can be based on the QLWE assumption.

Theorem 1.1 (informal). Assuming QLWE and QFHE, there exist a classical, post-quantumly secure, computational-zero-knowledge argument in a constant number of rounds for any $L \in NP$.

Combining our zero-knowledge protocol with previous work by Broadbent et al. [BJSW16, BG19], yields constant-round zero-knowledge arguments for QMA with quantum honest parties.
Corollary 1.1 (informal). Assuming QLWE and QFHE, there exist a quantum, post-quantumly secure, computational-zero-knowledge argument in a constant number of rounds for any $L \in \mathcal{QMA}$.

Main Technical Contribution: Non-Black-Box Quantum Extraction. Our main technical contribution is a new technique for extracting information from quantum circuits in a constant number of rounds. The technique circumvents the quantum information barriers previously mentioned. A key feature that enables this is using the adversary’s circuit representation in a non-black-box manner.

The technique, in particular, yields a constant round extractable commitment. In such a commitment protocol, the verifier can commit to a classical (polynomially long) string. This commitment is perfectly binding, and hiding against efficient quantum receivers. Furthermore, it guarantees the existence of a simulator, which given non-black-box access to the sender’s code, can simulate its view while extracting the committed plaintext. Further details are given in the technical overview below.

1.2 Technical Overview

We next discuss the main challenges in the design of post-quantum zero knowledge in constant rounds, and our main technical ideas toward overcoming these challenges.

1.2.1 Classical Protocols and the Quantum Barrier

To understand the challenges behind post-quantum zero knowledge, let us first recall how classical constant-round protocols work, and identify why they fail in the quantum setting. Classical constant-round protocols typically involve three main steps: (1) a prover commitment $\alpha$ to a set of bits, (2) a verifier challenge $\beta$, and (3) a prover response $\gamma$, in which it opens the commitments corresponding to the challenge $\beta$. For instance, in the 3-coloring protocol of [GMW86], the prover commits to the (randomly permuted) vertex colors, the verifier picks some challenge edge, and the prover opens the commitments corresponding to the vertices of that edge. To guarantee a negligible soundness error, this is repeated in parallel a polynomial number of times.

As described so far, the protocol satisfies a rather weak zero-knowledge guarantee — a simulator can efficiently simulate the verifier’s view in the protocol if it knows the verifier’s challenge $\beta$ ahead of time.

To obtain an actual zero-knowledge protocol, we need to exhibit a simulator for any malicious verifier, including ones who may arbitrarily choose their challenge depending on the prover’s message $\alpha$. For this purpose, an initial step (0) is added where the verifier commits ahead of time to its challenge, later opening it in step (2) [GK96a].

The added step allows the simulator to obtain the verifier’s challenges ahead of time by means of rewinding. Specifically, having obtained the verifier commitment, the simulator takes a snapshot of the verifier’s state and then runs it twice: first it generates a bogus prover commitment, and obtains the verifier challenge, then with the challenges at hand, it returns to the snapshot (effectively rewinding the verifier) and runs the verifier again to generate the simulated execution. The binding of the verifier’s commitment guarantees that it will never use a different challenge, and thus simulation succeeds.

Barriers to Post-Quantum Security. By appropriately instantiating the verifier commitment, the above protocol can be shown to be sound against unbounded provers, and in particular efficient quantum provers. One could expect that by instantiating the prover’s commitments so to guarantee hiding against quantum adversaries, we would get post-quantum zero knowledge. However, we do not know how to prove that such a protocol is zero knowledge against quantum verifiers. Indeed, the simulation strategy described above fails due to two basic concepts of quantum information theory:

- **No Cloning:** General quantum states cannot be copied. In particular, the simulator cannot take a snapshot of the verifier’s state.
• **Quantum State Disturbance:** General quantum circuits, which in particular perform measurements, are not reversible. Once the simulator evaluates the verifier’s quantum circuit to obtain its challenge, the verifier’s original state (prior to this bogus execution) has already been disturbed and cannot be recovered.

Watrous [Wat09] showed that in certain settings the rewinding barrier can be circumvented. He presents a *quantum rewinding lemma* that roughly, shows how non-rewinding simulators that succeed in simulating only with some noticeable probability can be amplified into full-fledged simulators. The quantum rewinding lemma allows proving that classical protocols, like the GMW protocol are post-quantum zero knowledge (assuming commitments with hiding against quantum adversaries). The technique is insufficient, however, to prove post-quantum zero knowledge of existing constant-round protocols with a *negligible soundness error*, such as the GK protocol described above. For such protocols, non-rewinding simulators with a noticeable success probability are not known.

**Can Non-Black-Box Techniques Cross the Quantum Barriers?** Rewinding is, in fact, often an issue also in the classical setting. Starting with the work of Goldreich and Krawczyk [GK96b], it was shown that constant-round zero-knowledge protocols with certain features, such as a public-coin verifier, cannot be obtained using simulators that only use the verifier’s next message function as a black box. That is, simulators that are based solely on rewinding. Surprisingly, Barak [Bar01] showed that these barriers can be circumvented using *non-black-box techniques*. He constructed a constant-round public-coin zero-knowledge protocol where the simulator takes advantage of the explicit circuit representation of the verifier. Following Barak’s work, different non-black-box techniques have been introduced to solve various problems in cryptography (c.f., [DGS09, CLP13, Goy13, BP15, CPS16]).

A natural question is whether we can leverage classical protocols with non-black-box simulators, such as Barak’s, in order to circumvent the discussed barriers in the quantum setting. Trying to answer this question reveals several challenges. One inherent challenge is that classical non-black-box techniques naturally involve cryptographic tools that support classical computations. Obtaining zero knowledge against quantum verifiers would require analogous tools for quantum computations. As an example, Barak relies on the existence of constant-round succinct proof systems for the correctness of classical computations; to obtain post-quantum zero knowledge, such a protocol would need to support also quantum computations, while (honest) verification should remain classical. Existing protocols for classical verification of quantum computations [Mah18b] are neither constant round nor succinct.

Another family of non-black-box techniques [BP15, BKP19], different from that of Barak, is based on fully-homomorphic encryption. Here (as mentioned above) constructions for homomorphic evaluation of quantum computations exist [Mah18a, Bra18]. The problem is that the mentioned non-black-box techniques do perform state cloning. Roughly speaking, starting from the same state, they evaluate the verifier’s computation (at least) twice: once homomorphically, under the encryption, and once in the clear.\(^3\) An additional hurdle is proving soundness against quantum provers. Known non-black-box techniques are sound against efficient classical provers, and often use tools that are not known in the quantum setting, such as constant-round knowledge extraction (which is further discussed below).

Our main technical contribution is devising a non-black-box technique that copes with the above challenges. We next explain the main ideas behind the technique.

1.2.2 **Our Technique: A No-Cloning Extraction Procedure**

Toward describing the technique, we restrict attention to a more specific problem. Specifically, constructing a constant-round post-quantum zero-knowledge protocol can be reduced to the problem of

\(^3\)In fact, Barak’s technique also seems to require state cloning. Roughly speaking, the same verifier state is used once for simulating the main verifier execution and once when computing the proof for the verifier’s computation.
constructing constant-round quantumly-extractable commitments. We recall what such commitments are and why they are sufficient, and then move to discuss the commitments we construct.

A quantumly-extractable commitment is a classical protocol between a sender $\text{Sen}$ and a receiver $\text{Rec}$. The protocol satisfies the standard (statistical) binding and post-quantum hiding, along with a plaintext extraction guarantee. Extraction requires that there exists an efficient quantum simulator $\text{Ext}$ that given any malicious sender $\text{Sen}^*$, represented by a polynomial-size quantum circuit, can simulate the view of $\text{Sen}^*$ in the commitment protocol while extracting the committed plaintext message. Specifically, $\text{Ext}(\text{Sen}^*)$ outputs a classical transcript $T$, a quantum state $|\tilde{\psi}\rangle$, and an extracted plaintext $\tilde{m}$ that are computationally indistinguishable from a real transcript, state, and plaintext $(T, |\psi\rangle, m)$, where $T$ and $|\psi\rangle$ are the transcript and sender state generated at the end of a real interaction between the receiver Rec and sender $\text{Sen}^*$, and $m$ is the plaintext fixed by the commitment transcript $T$.

Such commitments allow enhancing the classical four-step protocol described before to satisfy post-quantum zero-knowledge. We simply instantiate the verifier’s commitment to the challenge $\beta$ in step (0) with a quantumly-extractable commitment. To simulate a malicious quantum verifier $V^*$, the zero-knowledge simulator can then invoke the commitment simulator $\text{Ext}(V^*)$, with $V^*$ acting as the sender, to obtain a simulated commitment as well as the corresponding challenge $\beta$. Now the simulator knows the challenge ahead of time, before producing the prover message $\alpha$ in step (1), and using the (simulated) verifier state $|\tilde{\psi}\rangle$, can complete the simulation, without any state cloning. (Proving soundness is actually tricky on its own due to malleability concerns. We remain focused on zero knowledge for now).

The challenge is of course to obtain constant-round commitments with no-cloning extraction. Indeed, classically-extractable commitments have been long known in constant rounds under minimal assumptions, based on rewinding (and thus state cloning) [PRS02]. We next describe our non-black-box technique and how it enables quantum extraction without state cloning.

The Non-Black-Box Quantum Extraction Technique: A Simple Case. To describe the technique, we first focus on a restricted class of adversarial senders that are non-aborting and explainable. The notion of non-aborting explainable senders considers senders $\text{Sen}^*$ whose messages can always be explained as a behavior of the honest (classical) sender with respect to some plaintext and randomness (finding this explanation may be inefficient); in particular, they never abort. The notion further restricts that of (aborting) explainable adversaries from [BKP19], which also allows aborts. To even further simplify our exposition, we first address classical (rather than quantum) senders, but crucially, while avoiding any form of state cloning. Later on, we shall address general quantum adversaries.

Our protocol is inspired by [BP15, BKP19] and relies on two basic tools. The first is fully-homomorphic encryption (FHE) — an encryption scheme that allows to homomorphically apply any polynomial-size circuit $C$ to an encryption of $x$ to obtain a new encryption of $C(x)$, proportional in size to the result $|C(x)|$ (the size requirement is known as compactness). The second is compute-and-compare program obfuscation (CCO). A compute-and-compare program $\text{CC}[f, s, z]$ is given by a function $f$ (represented as a circuit), a target string $s$ in its range, and a message $z$; it outputs $z$ on every input $x$ such that $f(x) = s$, and rejects all other inputs. A corresponding obfuscator compiles any such program into a program $\text{CC}$ with the same functionality. In terms of security, provided that the target $s$ has high entropy conditioned on $f$ and $z$, the obfuscated program is computationally indistinguishable from a simulated dummy program, independent of $(f, s, z)$. Such post-quantumly-secure obfuscators are known under QLWE [GKW17, WZ17, GKVW19].

To commit to a message $m$, the protocol consists of three steps:

1. The sender $\text{Sen}$ samples:
   - two random strings $s$ and $t$,
   - a secret key $sk$ for an FHE scheme,
   - an FHE encryption $c_t = \text{FHE.Enc}_{sk}(t)$ of $t$,
• an obfuscation $\tilde{\mathcal{C}}$ of $\mathcal{C}[f, s, z]$, where $z = (m, sk)$ and $f = \text{FHE.Dec}_{sk}$ is the FHE decryption circuit.

It then sends $(ct_t, \tilde{\mathcal{C}})$ to the receiver $R$.

2. The receiver $\text{Rec}$ sends a guess $t'$.

3. $\text{Sen}$ rewards a successful guess: if $t = t'$, it sends back $s$ (and otherwise $\perp$).

The described commitment protocol comes close to our objective. First, it is binding — the obfuscation $\tilde{\mathcal{C}}$ uniquely determines $z = (m, sk)$. Second, it is hiding — a receiver (even if malicious) gains no information about the message $m$. To see this, we argue that no receiver sends $t' = t$ at the second message, but with negligible probability. Indeed, given only the first sender message $(ct_t, \mathcal{C})$, the receiver obtains no information about $s$. Hence, we can invoke the CCO security and replace the obfuscation $\mathcal{C}$ with a simulated one, which is independent of the secret FHE key $sk$. This, in turn, allows us to invoke the security of encryption to argue that the first message $(ct_t, \mathcal{C})$ hides $t$. It follows that the third sender message is $\perp$ (rather than the target $s$) with overwhelming probability, which again by CCO security implies that the entire view of the receiver can be simulated independently of $m$.

Lastly, a non-black-box simulator, given the circuit representation of an explainable sender $\text{Sen}^*$, can simulate the sender’s view, while extracting $m$. It first runs the sender to obtain the first message $(ct_t, \mathcal{C})$. At this point, it can use the sender’s circuit $\text{Sen}^*$ to continue the emulation of $\text{Sen}^*$ homomorphically under the encryption $ct_t$. The key point is that, under the encryption, we do have $t$. We can (homomorphically) feed $t$ to the sender, and obtain an encryption $ct_s$ of $s$. Now, the simulator feeds $ct_s$ to the obfuscation $\mathcal{C}$, and gets back $z = (m, sk)$. (Note that here the compactness of FHE is crucial — the sender $\text{Sen}^*$ could be of arbitrary polynomial size, whereas $\tilde{\mathcal{C}}$ and thus also $ct_s$ are of fixed size.)

Having extracted $m$, it remains to simulate the inner (for now, classical) state $\psi$ of the sender $S^*$ and the full interaction transcript $T$. These are actually available, but in encrypted form, as a result of the previous homomorphic computation. Here we use the fact that the extracted $z$ also includes the decryption key $sk$, allowing us to obtain the state $\psi$ and transcript $T$ in the clear.

An essential difference between the above extraction procedure and previous non-black-box extraction techniques (e.g., [BP15, BKP19]) is that it does not perform any state cloning. As explained earlier, previous procedures would perform the same computation twice, once under the encryption, and once in the clear. Here we perform the computation once, partially in the clear, and partially homomorphically. Crucially, we have a mechanism to peel off the encryption at the end of second part so that we do not have to redo the computation in the clear.

**Indistinguishability through Secure Function Evaluation.** The described protocol does not quite achieve our objective. The simulated interaction is, in fact, easy to distinguish from a real one. Indeed, in a simulated interaction the simulator’s guess in the second message is $t' = t$, whereas the receiver cannot produce this value. To cope with this problem, we augment the protocol yet again, and perform the second step under a secure function evaluation (SFE) protocol. This can be thought of as homomorphic encryption with an additional circuit privacy guarantee, which says that the result of homomorphic evaluation of a circuit, reveals nothing about the evaluated circuit to the decryptor, except of course from the result of evaluation.
The augmented protocol is similar to the previous one, except for the last two steps, now done using SFE:

1. The sender $Sen$ samples:
   - two random strings $s$ and $t$,
   - a secret key $sk$ for an FHE scheme,
   - an FHE encryption $ct_{t} = \text{FHE.Enc}_{sk}(t)$ of $t$,
   - an obfuscation $\overline{CC}$ of $CC[f, s, z]$, where $z = (m, sk)$ and $f = \text{FHE.Dec}_{sk}$ is the FHE decryption circuit.

   It then sends $(ct_{t}, \overline{CC})$ to the receiver $Rec$.

2. The receiver $Rec$ sends $ct'_{t}$, a guess $t'$ encrypted using SFE. (The honest receiver sets $t'$ arbitrarily.)

3. $Sen$ homomorphically evaluates the function that given input $t$, returns $s$ (and otherwise $\bot$). $Sen$ then returns the resulting ciphertext to $Rec$.

The homomorphic computation done by the simulator in the new protocol is augmented accordingly — instead of sending $t$ and obtaining $s$ directly, it now sends an SFE encryption of $t$ and obtains back an SFE encryption of $s$, which it can then decrypt to obtain $s$. Thus, as before, the homomorphic computation results in an FHE encryption of $s$. Indistinguishability of the simulated sender view from the real sender view now follows since the SFE encryption $ct'_{t}$ hides $t'$. The SFE circuit privacy guarantees that the homomorphic SFE evaluation does not leak any information about the target $s$, as long as the receiver does not send an SFE encryption of $t$.

A Malleability Problem and its Resolution. While we could argue before that a malicious receiver cannot output $t$ in the clear, arguing that it does not output an SFE encryption of $t$ is more tricky. In particular, the receiver might be able to somehow maul the FHE encryption $ct_{t}$ to get an SFE encryption $ct'_{t}$ of the value $t$, without actually “knowing” the value $t$. Classically, such malleability problems are solved using extraction. If we could efficiently extract the value encrypted in the SFE encryption $ct'_{t}$, then we could rely on the previous argument. However, as explained before, efficient extraction is classically achieved using rewinding and thus state cloning. While so far we have focused on avoiding state cloning for the sake of simulating the sender, we should also avoid state cloning when proving hiding of the commitment as we are dealing with quantum receivers. It seems like we are back to square one.

To circumvent the problem, we rely on the fact that the hiding requirement of the commitment is relatively modest — commitments to different plaintexts should be indistinguishable. This is in contrast the efficient simulation requirement for the sender (needed for efficient zero knowledge simulation). Here one commonly used solution is complexity leveraging — we can design the SFE, FHE, and CCO so that extraction from SFE encryptions can be done in brute force, without any state cloning, and without compromising the security of the FHE and CCO. This comes at the cost of assuming subexponential (rather than just polynomial) hardness of the primitives in use.

A different solution, which is also the one we use in the body of the paper, relies on hardness against efficient quantum adversaries with non-uniform quantum advice (instead of subexponential hardness). Specifically, the receiver sends a commitment to the SFE encryption key in the beginning of the protocol. The reduction establishing the hiding of the protocol gets as non-uniform advice the initial receiver (quantum) state that maximizes the probability of breaking hiding, along with the corresponding SFE key. This allows for easy extraction from SFE encryptions, without any state cloning.

The full solution contains additional steps meant to establish that the receiver’s messages are appropriately structured (e.g., the receiver’s commitment defines a valid SFE key, and the SFE encryption later indeed uses that key). This is done using standard techniques based on witness-indistinguishable proofs, which exist in a constant number of rounds [GMW86] assuming commitments with post-quantum hiding (and in particular, QLWE).
Dealing with Quantum Adversaries. Above, we have assumed for simplicity that the sender is classical and have shown a simulation strategy that requires no state cloning. We now explain how the protocol is augmented to deal with quantum senders (for now still restricting attention to non-aborting explainable senders). The first natural requirement in order to deal with quantum senders is that the cryptographic tools in use (e.g., SFE encryption) will be post-quantum secure. This can be guaranteed assuming QLWE.

As already mentioned earlier in the introduction, post-quantum security alone is not enough — we need to make sure that our non-black-box extraction technique can also work with quantum, rather than classical, circuits representing the sender $Sen^*$. For this purpose, we use quantum fully-homomorphic encryption (QFHE). In a QFHE scheme, the encryption and decryption keys are (classical) strings and the encryption and decryption algorithms are classical provided that the plaintext is classical (and otherwise quantum). Most importantly, QFHE allows to homomorphically evaluate quantum circuits. Such QFHE schemes were recently constructed in [Mah18a, Bra18] based on QLWE and a circular security assumption (analogous to the assumptions required for multi-key FHE in the classical setting).

The augmented protocol simply replaces the FHE scheme with a QFHE scheme (other primitives, such as the SFE and compute-and-compare are completely classical in terms of functionality and only need to be post-quantum secure). In the augmented protocol, the honest sender and receiver still act classically. In contrast, the non-black-box simulator described before is now quantum — it homomorphically evaluates the quantum sender circuit $Sen^*$. A technical point is that QFHE should support the evaluation of a quantum circuit with an additional quantum auxiliary input — in our case the quantum sender $Sen^*$ and its inner state after it sends the first message. This is achieved by existing QFHE schemes (for instance, by using their public key encryption mode, and encrypting the initial state prior to the computation).

Dealing with Aborts. So far, we have dealt with explainable senders that are non-aborting. This is indeed a strong restriction and in fact, quantumly-extractable commitments against this class of senders can be achieved using black-box techniques (see more in the related work section). However, considering an adversary who, with noticeable probability, may abort at some stage of the protocol, existing black-box techniques completely fail (even if the adversary is explainable up to the abort). In contrast, as we shall see, our non-black-box technique will enable simulation also for aborting senders.

In our protocol, an aborting sender $Sen^*$ may refuse to perform the SFE evaluation in the last step of the protocol. In this case, the simulator will get stuck — the simulated transcript and sender state $|\psi\rangle$ will remain forever locked under the encryption (since the simulator cannot use the obfuscation $\tilde{CC}$ to get the decryption key $sk$). Accordingly, the described simulator successfully simulates senders that never abort, but fails to simulate senders that abort (noticeably often). We next observe that there is, in fact, a non-re-winding simulation strategy also for the other extreme, namely for senders $Sen^*$ that (almost) always abort. Here the simulator would simply send in the clear (rather than under FHE) an SFE encryption $ct'_t$ of an arbitrary string $t'$, just like the honest receiver Rec. In this case, the simulated sender view is identical to its view in a real interaction (and since the sender $Sen^*$ aborts, there is no need to extract the plaintext message).

We show that the two simulators described, $Sim_{na}$ for never-aborting senders and $Sim_{aa}$ for always-aborting senders, can be combined into a simulator for general senders (which sometimes abort). This is enabled by the fact that simulated receiver messages $ct'_{t'}$ generated by the two simulators are indistinguishable due to the hiding of SFE encryptions. Accordingly, the sender’s choice of whether to abort or not is (computationally) independent of whether we are simulating using the first simulator $Sim_{na}$ or the second $Sim_{aa}$. This gives rise to a combined simulator $Sim_{comb}$, which flips a random coin $b \leftarrow \{na, aa\}$ to predict whether an abort will occur, and then runs $Sim_b$. The combined simulator $Sim_{comb}$ succeeds if it guessed correctly, which occurs with probability (negligibly close to) half.

Applying Watrous’ Quantum Rewinding Lemma. The above is reminiscent of the simulation strategy in classical 3-message zero-knowledge protocols (with a large soundness error), such as the GMW 3-coloring protocol [GMW87]. In these protocols, for each possible verifier challenge $\beta$ there exists a
non-rewinding simulator $\text{Sim}_\beta$, and the combined simulator $\text{Sim}_{\text{comb}}$ tries to guess the challenge $\beta$ and apply the corresponding simulator. Similarly to the combined simulator in our protocol, the verifier’s choice of challenge $\beta$ is (computationally) independent of $\text{Sim}_{\text{comb}}$’s guess, and thus the simulator $\text{Sim}_{\text{comb}}$ succeeds in simulating with some fixed noticeable probability (specifically $2^{-|\beta|}$).

The advantage of such simulators (non-rewinding and successful with fixed noticeable probability) is that they can be amplified to full-fledged simulators, both classically and quantumly. In the classical setting, a full-fledged simulator $\text{Sim}$ can be obtained by rerunning $\text{Sim}_{\text{comb}}$ until it succeeds. We can, in fact, apply the same rerunning strategy also for quantum verifiers. However, this does not guarantee zero knowledge against verifiers with quantum auxiliary input (since each execution of $\text{Sim}_{\text{comb}}$ may disturb the verifier’s auxiliary state). To obtain zero knowledge against verifiers with quantum auxiliary input, we apply Watrous’ quantum rewinding lemma [Wat09], which shows how to faithfully amplify the combined simulator $\text{Sim}_{\text{comb}}$ in the presence of quantum auxiliary input.

From Explainable Adversaries to Malicious Ones. The only remaining gap is the assumption that senders are explainable; that is, the messages they send (up to the point that they possibly abort), can always be explained as messages that would be sent by the honest (classical) sender for some plaintext and randomness. The simulator $\text{Sim}_{\text{na}}$ (for never-aborting verifiers) crucially relies on this; in particular, the CCO $\tilde{\text{CC}}$ and the FHE ciphertext $ct_t$ must be formed consistently with each other for the simulator to work. Importantly, it suffices that there exists an explanation for the messages, and we do not have to efficiently extract it as part of the simulation; indeed, efficient quantum extraction is exactly the problem we are trying to solve.

The commitment protocol against explainable senders naturally gives rise to a zero-knowledge protocol against explainable verifiers. As is often the case in the design of zero knowledge protocols (see discussion in [BKP19]), dealing with explainable verifiers is actually the hard part of designing zero-knowledge protocols. Indeed, we use a generic transformation of [BKP19], slightly adapted to our setting, which converts zero-knowledge protocols against explainable verifiers to ones against arbitrary malicious verifiers. The transformation is based on constant-round (post-quantumly-secure) witness-indistinguishable proofs, which as mentioned before can be obtained based on QLWE.

1.3 More Related Work on Post-Quantum Zero Knowledge

The study of post-quantum zero-knowledge (QZK) protocols was initiated by van de Graaf [VDGC97], who first observed that traditional zero-knowledge simulation techniques, based on rewinding, fail against quantum verifiers. Subsequent work has further explored different flavors of zero knowledge and their limitations [Wat02], and also demonstrated that relaxed notions such as zero-knowledge with a trusted common reference string can be achieved [Kob03, DFS04]. Later on, Peikert and Shiehian constructed non-interactive post-quantum zero knowledge from QLWE in the common random string model [PS19]. Watrous [Wat09] was the first to show that the barriers of quantum information theory can be crossed, demonstrating a post-quantum zero-knowledge protocol for $\text{NP}$ in a polynomial number of rounds (in the plain model).

Zero Knowledge for QMA. Another line of work aims at constructing quantum (rather than classical) protocols for QMA (rather than $\text{NP}$). Following a sequence of works [BOCG+06, Liu06, DNS10, DNS12, MHNF15], Broadbent, Ji, Song and Watrous [BJSW16] show a zero-knowledge quantum proof system for all of QMA (in a polynomial number of rounds).

Quantum Proofs and Arguments of Knowledge. Extracting knowledge from quantum adversaries was investigated in a sequence of works [Unr12, HSS11, LN11, ARU14]. A line of works considered different variants of quantum proofs and arguments of knowledge (of the witness), proving both feasibility

\footnote{This is in contrast to other restrictions of the adversary considered in the literature, like semi-honest and semi-malicious adversaries [GMW87, HIK+11, BGJ+13].}
results and limitations. In particular, Unruh [Unr12] shows that assuming post-quantum injective one-way functions, some existing systems are a quantum proof of knowledge. He identifies a certain strict soundness requirement that suffices for such an implication. Ambainis, Rosmanis and Unruh [ARU14] give evidence that this requirement may be necessary.

Based on QLWE, Hallgren, Smith, and Song [HSS11] and Lunemann and Nielsen [LN11] show argument of knowledge where it is also possible to simulate the prover’s state (akin to our simulation requirement of the sender’s state). Unruh further explores arguments of knowledge in the context of computationally binding quantum commitments [Unr16b, Unr16a]. All of the above require a polynomial number of rounds to achieve a negligible knowledge error.

**Zero-Knowledge Multi-Prover Interactive Proofs.** Two recent works by Chiesa et al. [CFGS18] and by Grilo, Slofstra, and Yuen [GSY19] show that NEXP and MIP∗, respectively, have perfect zero-knowledge multi-prover interactive proofs (against entangled quantum provers).

**Concurrent Work.** Broadbent and Grilo [BG19] construct quantum sigma protocols for QMA, that is, 3-message protocols that are zero-knowledge but have large soundness error. Relying on their protocol and our zero-knowledge protocol and extractable commitment, we obtain a conceptually simple constant-round zero-knowledge protocol for QMA with a negligible soundness error (in a previous version of our work, we constructed such a protocol based on earlier work of [BJSW16]). Coladangelo, Vidick, and Zhang [CVZ19] construct non-interactive zero-knowledge arguments with preprocessing for QMA in the common reference string model. The challenges tackled and corresponding techniques in our work are substantially different than those in both of the above mentioned works.

Ananth and La Placa [AP] developed a non-black-box quantum extraction protocol that share some of our ideas and is based on similar computational assumptions. They used it to obtain quantum zero-knowledge, but only against explainable non-aborting verifiers.

**A Word on Strict Commitments and Non-Aborting Verifiers.** In [Unr12], Unruh introduces a notion of strict commitments, which are commitments that fix not only the plaintext, but also the randomness (e.g. Blum-Micali [BM84]), and are known to exist based on injective one-way functions. As mentioned in our technical overview, using such commitments it is possible to obtain zero-knowledge in constant rounds against non-aborting explainable verifiers through the GK four-step template we discussed in the overview. Roughly speaking, this is because when considering verifiers that always open their (strict) commitments, we are assured that measuring their answer does not disturb the verifier state, as this answer is information-theoretically fixed. This effectively allows to perform rewinding.

## 2 Preliminaries

We rely on standard notions of classical Turing machines and Boolean circuits:

- A PPT algorithm is a probabilistic polynomial-time Turing machine.

- We sometimes think about PPT algorithms as polynomial-size uniform families of circuits, these are equivalent models. A polynomial-size circuit family $\mathcal{C}$ is a sequence of circuits $\mathcal{C} = \{C_\lambda\}_{\lambda \in \mathbb{N}}$, such that each circuit $C_\lambda$ is of polynomial size $\lambda^{O(1)}$. We say that the family is uniform if there exists a deterministic polynomial-time algorithm $M$ that on input $1^\lambda$ outputs $C_\lambda$.

- For a PPT algorithm $M$, we denote by $M(x; r)$ the output of $M$ on input $x$ and random coins $r$. For such an algorithm and any input $x$, we write $m \in M(x)$ to denote the fact that $m$ is in the support of $M(x; \cdot)$.

We follow standard notions from quantum computation.
• A QPT algorithm is a quantum polynomial-time Turing machine.

• We sometimes think about QPT algorithms as polynomial-size uniform families of quantum circuits, these are equivalent models. A polynomial-size quantum circuit family $\mathcal{C}$ is a sequence of quantum circuits $\mathcal{C} = \{C_\lambda\}_{\lambda \in \mathbb{N}}$, such that each circuit $C_\lambda$ is of polynomial size $\lambda^{O(1)}$. We say that the family is uniform if there exists a deterministic polynomial-time algorithm $M$ that on input $1^\lambda$ outputs $C_\lambda$.

• An interactive algorithm $M$, in a two-party setting, has input divided into two registers and output divided into two registers. For the input, one register $I_m$ is for an input message from the other party, and a second register $I_a$ is an auxiliary input that acts as an inner state of the party. For the output, one register $O_m$ is for a message to be sent to the other party, and another register $O_a$ is again for auxiliary output that acts again as an inner state. For a quantum interactive algorithm $M$, both input and output registers are quantum.

The Adversarial Model. Throughout, efficient adversaries are modeled as quantum circuits with non-uniform quantum advice (i.e. quantum auxiliary input). Formally, a polynomial-size adversary $A^* = \{A^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$, consists of a polynomial-size non-uniform sequence of quantum circuits $\{A^*_\lambda\}_{\lambda \in \mathbb{N}}$, and a sequence of polynomial-size mixed quantum states $\{\rho_\lambda\}_{\lambda \in \mathbb{N}}$.

For an interactive quantum adversary in a classical protocol, it can be assumed without the loss of generality that its output message register (the register containing the message to be sent to the other side, not the register containing output quantum auxiliary information) is always measured in the computational basis at the end of computation. This assumption is indeed without the loss of generality, because whenever a quantum state is sent through a classical channel then qubits decohere and are effectively measured in the computational basis.

Indistinguishability in the Quantum Setting.

• Let $f : \mathbb{N} \to [0, 1]$ be a function.
  - $f$ is negligible if for every constant $c \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n > N$, $f(n) < n^{-c}$.
  - $f$ is noticeable if there exists $c \in \mathbb{N}, N \in \mathbb{N}$ such that for every $n \geq N$, $f(n) \geq n^{-c}$.
  - $f$ is overwhelming if it is in the form $1 - \mu(n)$, for a negligible function $\mu$.

• We may consider random variables over bit strings or over quantum states. This will be clear from the context.

• For two random variables $X$ and $Y$ supported on quantum states, quantum distinguisher circuit $D$ with, quantum auxiliary input $\rho$, and $\mu \in [0, 1]$, we write $X \approx_{D, \rho, \mu} Y$ if
  \[ |\Pr[D(X; \rho) = 1] - \Pr[D(Y; \rho) = 1]| \leq \mu. \]

• Two ensembles of random variables $\mathcal{X} = \{X_i\}_{\lambda \in \mathbb{N}, i \in I_\lambda}, \mathcal{Y} = \{Y_i\}_{\lambda \in \mathbb{N}, i \in I_\lambda}$ over the same set of indices $I = \cup_{\lambda \in \mathbb{N}} I_\lambda$ are said to be computationally indistinguishable, denoted by $\mathcal{X} \approx_{c} \mathcal{Y}$, if for every polynomial-size quantum distinguisher $D = \{D_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}, i \in I_\lambda$,
  \[ X_i \approx_{D_\lambda, \rho_\lambda, \mu(\lambda)} Y_i. \]
The trace distance between two distributions $X,Y$ supported over quantum states, denoted \( \text{TD}(X,Y) \), is a generalization of statistical distance to the quantum setting and represents the maximal distinguishing advantage between two distributions supported over quantum states, by unbounded quantum algorithms. We thus say that ensembles $X = \{X_i\}_{i \in I}$, $Y = \{Y_i\}_{i \in I}$, supported over quantum states, are statistically indistinguishable (and write $X \approx_s Y$), if there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $i \in I$, \[
abla \text{TD}(X_i, Y_i) \leq \mu(\lambda) .
\]

In what follows, we introduce the cryptographic tools used in this work. By default, all algorithms are classical and efficient unless stated otherwise, and security holds against polynomial-size non-uniform quantum adversaries with quantum advice.

### 2.1 Interactive Protocols, Witness Indistinguishability, and Zero Knowledge

We define proof and argument systems that are secure against quantum adversaries. We start with classical protocols and proceed to define quantum protocols. In what follows, we denote by $(P,V)$ a protocol between two parties $P$ and $V$. For common input $x$, we denote by $\text{OUT}_V(P,V)(x)$ the output of $V$ in the protocol. For honest verifiers, this output will be a single bit indicating acceptance or rejection of the proof. Malicious quantum verifiers may have arbitrary quantum output (which is formally captured by the verifier outputting its inner quantum state).

**Definition 2.1 (Classical Proof and Argument Systems for NP).** Let $(P,V)$ be a protocol with an honest PPT prover $P$ and an honest PPT verifier $V$ for a language $L \in \text{NP}$, satisfying:

1. **Perfect Completeness:** For any $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_L(x)$, \[
\Pr[\text{OUT}_V(P(w),V)(x) = 1] = 1 .
\]

2. **Soundness:** The protocol satisfies one of the following.

   - **Computational Soundness:** For any quantum polynomial-size prover $P^* = \{P^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ there exists a negligible function $\mu(\cdot)$ such that for any security parameter $\lambda \in \mathbb{N}$ and any $x \in \{0,1\}^\lambda \setminus L$, \[
\Pr[\text{OUT}_V(P^*_\lambda(\rho_\lambda),V)(x) = 1] \leq \mu(\lambda) .
\]
   A protocol with computational soundness is called an argument.

   - **Statistical Soundness:** There exists a negligible function $\mu(\cdot)$, such that for any (unbounded) prover $P^*$, any security parameter $\lambda \in \mathbb{N}$, and any $x \in \{0,1\}^\lambda \setminus L$, \[
\Pr[\text{OUT}_V(P^*,V)(x) = 1] \leq \mu(\lambda) .
\]
   A protocol with statistical soundness is called a proof.

**Definition 2.2 (Quantum Proof and Argument Systems for QMA).** Let $(P,V)$ be a quantum protocol with an honest QPT prover $P$ and an honest QPT verifier $V$ for a language $L \in \text{QMA}$, satisfying:

1. **Statistical Completeness:** There is a polynomial $k(\cdot)$ and a negligible function $\mu(\cdot)$ s.t. for any $\lambda \in \mathbb{N}, x \in L \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_L(x)$ \footnote{For a language $L$ in QMA, for an instance $x \in L$ in the language, the set $\mathcal{R}_L(x)$ is the (possibly infinite) set of quantum witnesses that make the BQP verification machine accept with some overwhelming probability $1 - \negl(\lambda)$.}, \[
\Pr[\text{OUT}_V(P(w^{\otimes k(\lambda)}),V)(x) = 1] \geq 1 - \mu(\lambda) .
\]

2. **Soundness:** As in Definition 2.1.
2.1.1 Witness Indistinguishability

We rely on classical constant-round (public-coin) proof systems for $NP$ that are witness-indistinguishable; that is, proofs that use different witnesses (for the same statement) are computationally indistinguishable (for quantum attackers).

**Definition 2.3** (WI Proof System for NP). A classical protocol proof system $(P, V)$ for a language $\mathcal{L} \in NP$ (as in Definition 2.1) is witness-indistinguishable if it satisfies:

**Witness Indistinguishability:** For every quantum polynomial-size verifier $V^* = \{V^*_\lambda(\rho_{\lambda})\}_\lambda$

\[
\{\text{OUT}_{V^*_\lambda}(P(u_0), V^*_\lambda(\rho_{\lambda}))(x)\}_{\lambda, x, w_0, w_1} \approx_c \{\text{OUT}_{V^*_\lambda}(P(w_1), V^*_\lambda(\rho_{\lambda}))(x)\}_{\lambda, x, w_0, w_1},
\]

where $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0, 1\}^\lambda$, and $w_0, w_1 \in \mathcal{R}_\mathcal{L}(x)$ are witnesses for $x$.

**Instantiations.** 3-message, public-coin classical proof systems with WI follow from classical zero-knowledge proof systems such as the parallel repetition of the 3-coloring protocol [GMW91], which is in turn based on non-interactive perfectly-binding commitments. For the proof system to be WI against quantum attacks, we need the non-interactive commitments to be computationally hiding against quantum adversaries, which can be instantiated for example from QLWE.

2.1.2 Sigma Protocols

We use the abstraction of Sigma Protocols, which are public-coin three-message proof systems with a special zero knowledge guarantee. We define both classical and quantum Sigma Protocols.

**Definition 2.4** (Classical Sigma Protocol for NP). A classical sigma protocol for $\mathcal{L} \in NP$ is a classical proof system $(\Sigma, P, \Sigma, V)$ (as in Definition 2.1) with 3 messages and the following syntax.

- $(\alpha, \tau) \leftarrow \Sigma.P_1(x, w) : \text{Given an instance } x \in \mathcal{L} \text{ and a witness } w \in \mathcal{R}_\mathcal{L}(x), \text{ the first prover execution outputs a public message } \alpha \text{ for } \Sigma.V \text{ and a private state } \tau$.
- $\beta \leftarrow \Sigma.V(x) : \text{The verifier simply outputs a string of } \text{poly}(|x|) \text{ random bits.}$
- $\gamma \leftarrow \Sigma.P_3(\beta, \tau) : \text{Given the verifier’s string } \beta \text{ and the private state } \tau, \text{ the prover outputs a response } \gamma$.

The protocol satisfies the following.

**Special Zero-Knowledge:** There exists a PPT simulator $\Sigma.S$ such that,

\[
\{(\alpha, \gamma) \mid (\alpha, \tau) \leftarrow \Sigma.P_1(x, w), \gamma \leftarrow \Sigma.P_3(\beta, \tau)\}_{\lambda, x, w, \beta} \approx_c \{(\alpha, \gamma) \mid (\alpha, \gamma) \leftarrow \Sigma.S(x, \beta)\}_{\lambda, x, w, \beta},
\]

where $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0, 1\}^\lambda, w \in \mathcal{R}_\mathcal{L}(x)$ and $\beta \in \{0, 1\}^{\text{poly}(\lambda)}$.

The next claim follows directly from the special zero-knowledge requirement, and will be used throughout.

**Claim 2.1** (First-Message Indistinguishability, [BKP18], Claim 8.1). In every $\Sigma$ protocol:

\[
\{\alpha \mid (\alpha, \tau) \leftarrow \Sigma.P_1(x, w)\}_{\lambda, x, w, \beta} \approx_c \{\alpha \mid (\alpha, \gamma) \leftarrow \Sigma.S(x, 0^{|\beta|})\}_{\lambda, x, w, \beta},
\]

where $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0, 1\}^\lambda, w \in \mathcal{R}_\mathcal{L}(x)$ and $\beta \in \{0, 1\}^{\text{poly}(\lambda)}$.

**Instantiations.** Like witness-indistinguishable proofs, Sigma protocols are known to follow from the parallel repetition of the 3-coloring protocol [GMW91]. For the protocol to have special zero knowledge against quantum attacks, we need the non-interactive commitment $\alpha$ to be computationally hiding against quantum adversaries, which can be instantiated for example from QLWE.
**Definition 2.5 (Quantum Sigma Protocol for QMA).** A quantum sigma protocol for $L \in \text{QMA}$ is a quantum proof system $(\Xi, P, \Xi, V)$ (as in Definition 2.2) with 3 messages and the following syntax.

- $(\alpha, \tau) \leftarrow \Xi.P_1(x, w^{\otimes k(\lambda)})$ : Given an instance $x \in L \cap \{0, 1\}^\lambda$ and $k(\lambda)$ witnesses $w \in \mathcal{R}_L(x)$ (for a polynomial $k(\cdot)$), the first prover execution outputs a public message $\alpha$ for $\Xi.V$ and a private inner state $\tau$.
- $\beta \leftarrow \Xi.V(x)$ : The verifier simply outputs a string of $\text{poly}(|x|)$ random bits.
- $\gamma \leftarrow \Xi.P_3(\beta, \tau)$ : Given the verifier’s string $\beta$ and the private state $\tau$, the prover outputs a response $\gamma$.

The protocol satisfies the following.

**Special Zero-Knowledge:** There exists a QPT simulator $\Xi.\text{Sim}$ such that,

$$\left\{ (\alpha, \gamma) \mid (\alpha, \tau) \leftarrow \Xi.P_1(x, w^{\otimes k(\lambda)}), \gamma \leftarrow \Xi.P_3(\beta, \tau) \right\}_{\lambda,x,w,\beta} \approx_c \left\{ (\alpha, \gamma) \mid (\alpha, \gamma) \leftarrow \Xi.\text{Sim}(x, \beta) \right\}_{\lambda,x,w,\beta},$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0, 1\}^\lambda$, $w \in \mathcal{R}_L(x)$ and $\beta \in \{0, 1\}^{\text{poly}(\lambda)}$.

**Instantiations.** Quantum sigma protocols follow from the parallel repetition of the 3-message quantum zero-knowledge protocols of [BG19] for QMA$^4$.

### 2.1.3 Quantum Zero-Knowledge Protocols

We next define post-quantum zero-knowledge classical protocols and zero-knowledge quantum protocols.

**Definition 2.6 (Post-Quantum Zero-Knowledge Classical Protocol).** Let $(P, V)$ be a classical protocol (argument or proof) for a language $L \in \text{NP}$ as in Definition 2.1. The protocol is quantum zero-knowledge if it satisfies:

**Quantum Zero Knowledge:** There exists a quantum polynomial-time simulator $\text{Sim}$, such that for any quantum polynomial-size verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$,

$$\left\{ \text{OUT}_{V^*_\lambda}(P(w), V^*_\lambda(\rho_\lambda)(x)) \right\}_{\lambda,x,w} \approx_c \left\{ \text{Sim}(x, V^*_\lambda, \rho_\lambda) \right\}_{\lambda,x,w},$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0, 1\}^\lambda$, $w \in \mathcal{R}_L(x)$.

- If $V^*$ is a classical circuit, then the simulator is computable by a classical polynomial-time algorithm.

**Definition 2.7 (Zero-Knowledge Quantum Protocol).** Let $(P, V)$ be a quantum protocol (argument or proof) for a language $L \in \text{QMA}$ as in Definition 2.2, where the prover uses $k(\lambda)$ copies of a witness. The protocol is quantum zero-knowledge if it satisfies:

**Quantum Zero Knowledge:** There exists a quantum polynomial-time simulator $\text{Sim}$, such that for any quantum polynomial-size verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$,

$$\left\{ \text{OUT}_{V^*_\lambda}(P(w^{\otimes k(\lambda)}), V^*_\lambda(\rho_\lambda)(x)) \right\}_{\lambda,x,w} \approx_c \left\{ \text{Sim}(x, V^*_\lambda, \rho_\lambda) \right\}_{\lambda,x,w},$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0, 1\}^\lambda$, $w \in \mathcal{R}_L(x)$.

$^4$The authors in [BG19] use the name “sigma protocols” differently than in this work. Specifically, [BG19] call their 3-message protocols, that are zero-knowledge but have large soundness error, “sigma protocols”. In this work we call the parallel repetition of such protocols (which have amplified soundness but weakened zero knowledge) “sigma protocols”.

13
2.2 Additional Tools

2.2.1 Compute-and-Compare Obfuscation

We define compute-and-compare (CC) circuits and obfuscators for CC circuits.

**Definition 2.8** (Compute-and-Compare Circuit). Let \( f : \{0, 1\}^n \rightarrow \{0, 1\}^\lambda \) be a circuit, and let \( u \in \{0, 1\}^\lambda, z \in \{0, 1\}^* \) be strings. Then \( \text{CC}[f, u, z](x) \) is a circuit that returns \( z \) if \( f(x) = y \), and \( \bot \) otherwise. \( \text{CC}[f, u, z] \) has a canonical description from which \( f, u, \) and \( z \) can be read.

We now define compute-and-compare (CC) obfuscators (with perfect correctness). In what follows \( \text{Obf} \) is a PPT algorithm that takes as input a CC circuit \( \text{CC}[f, u, z] \) and outputs a new circuit \( \tilde{\text{CC}} \).

**Definition 2.9** (CC obfuscator). A PPT algorithm \( \text{Obf} \) is a compute-and-compare obfuscator if it satisfies:

1. **Perfect Correctness:** For any circuit \( f : \{0, 1\}^n \rightarrow \{0, 1\}^\lambda, u \in \{0, 1\}^\lambda \) and \( z \in \{0, 1\}^* \),
   
   \[
   \Pr\left[ \forall x \in \{0, 1\}^n : \tilde{\text{CC}}(x) = \text{CC}[f, u, z](x) \mid \tilde{\text{CC}} \leftarrow \text{Obf}(\text{CC}[f, u, z]) \right] = 1 .
   \]

2. **Simulation:** There exists a PPT simulator \( \text{Sim} \) such that for every two polynomials \( \ell_1(\cdot), \ell_2(\cdot) \),

   \[
   \{ \text{CC} \mid u \leftarrow \{0, 1\}^\lambda, \tilde{\text{CC}} \leftarrow \text{Obf}(\text{CC}[f, u, z]) \}_{\lambda, f, z} \approx_c \{ \text{Sim}(1^{\ell_1(\lambda)}, 1^{\ell_2(\lambda)}, 1^\lambda) \}_{\lambda, f, z} ,
   \]

   where \( \lambda \in \mathbb{N}, f : \{0, 1\}^n \rightarrow \{0, 1\}^\lambda \) is a \( \ell_1(\lambda) \)-size circuit, \( z \in \{0, 1\}^{\ell_2(\lambda)} \).

**Instantiations.** Compute-and-compare obfuscators with almost-perfect correctness are constructed in [GKW17, WZ17] based on QLWE. CC obfuscators with perfect correctness are constructed [GKVW19] by Goyal, Koppula, Vusirikala and Waters, also based on QLWE.

2.2.2 Non-Interactive Commitments

We define non-interactive commitment schemes.

**Definition 2.10** (Non-Interactive Commitment). A non-interactive commitment scheme is given by a PPT algorithm \( \text{Com}(\cdot) \) with the following syntax:

- \( \text{cmt} \leftarrow \text{Com}(1^\lambda, x) : \) A randomized algorithm that takes as input a security parameter \( 1^\lambda \) and input \( x \in \{0, 1\}^* \), and outputs a commitment \( \text{cmt} \).

The commitment algorithm satisfies:

1. **Perfect Binding:** For any \( \lambda_0, \lambda_1 \in \mathbb{N}, x_0, x_1, r_0, r_1 \in \{0, 1\}^* \), \( \text{Com}(1^{\lambda_0}, x_0; r_0) = \text{Com}(1^{\lambda_1}, x_1; r_1) \) implies \( x_0 = x_1 \).

2. **Computational Hiding:** For any polynomial \( \ell(\cdot) \),

   \[
   \{ \text{Com}(1^\lambda, x_0) \}_{\lambda, x_0, x_1} \approx_c \{ \text{Com}(1^\lambda, x_1) \}_{\lambda, x_0, x_1} ,
   \]

   where \( \lambda \in \mathbb{N}, x_0, x_1 \in \{0, 1\}^{\ell(\lambda)} \).

**Instantiations.** The above non-interactive commitments are known based on various standard assumptions, including QLWE [GHKW17, LS19].
2.2.3 Quantum Fully Homomorphic Encryption

We rely on quantum fully homomorphic encryption, specifically, a scheme where a classical input can be encrypted classically and a quantum input quantumly. The formal definition follows.

**Definition 2.11** (Quantum Fully-Homomorphic Encryption). A quantum fully homomorphic encryption scheme is given by six algorithms \((\text{QHE.Keygen, QHE.Enc, QHE.Dec, QHE.QEnc, QHE.QDec, QHE.Eval})\) with the following syntax:

- \((\text{pk}, \text{sk}) \leftarrow \text{QHE.Keygen}(1^\lambda)\) : A PPT algorithm that given a security parameter \(1^\lambda\), samples a classical public key \(\text{pk}\) and a classical secret key \(\text{sk}\).
- \(\text{ct} \leftarrow \text{QHE.Enc}_{\text{pk}}(x)\) : A PPT algorithm that takes as input a classical string \(x \in \{0, 1\}^*\) and outputs a classical ciphertext \(\text{ct}\).
- \(|\phi\rangle \leftarrow \text{QHE.QEnc}_{\text{pk}}(|\psi\rangle)\) : A QPT algorithm that takes as input a quantum state \(|\psi\rangle\) and outputs a quantum ciphertext \(|\phi\rangle\).
- \(x \leftarrow \text{QHE.Dec}_{\text{sk}}(\text{ct})\) : A PPT algorithm that takes as input a classical ciphertext \(\text{ct}\) and outputs a string \(x\).
- \(|\psi\rangle \leftarrow \text{QHE.QDec}_{\text{sk}}(|\phi\rangle)\) : A QPT algorithm that takes as input a quantum ciphertext \(|\phi\rangle\) and outputs a quantum state \(|\psi\rangle\).
- \(|\phi\rangle \leftarrow \text{QHE.Eval}_{\text{pk}}(C, \text{ct}, |\phi\rangle)\) : A QPT algorithm that takes as input a general quantum circuit \(C\), a classical ciphertext \(\text{ct}\) and a quantum ciphertext \(|\phi\rangle\) and outputs an evaluated quantum ciphertext \(|\phi\rangle\).

The scheme satisfies the following.

- **Quantum Semantic Security:** For every polynomial \(\ell(\cdot)\),
  
  \[
  \begin{align*}
  \{ (ct, |\phi\rangle) \} & \approx_c \{ (\text{pk}, \text{sk}) \leftarrow \text{QHE.Keygen}(1^\lambda), \\
  & \quad \text{ct} \leftarrow \text{QHE.Enc}_{\text{pk}}(x_0), \\
  & \quad |\phi\rangle \leftarrow \text{QHE.QEnc}_{\text{pk}}(|\psi_0\rangle) \} \quad \lambda, x_0, |\psi_0\rangle, x_1, |\psi_1\rangle
  \\
  \{ (ct, |\phi\rangle) \} & \approx_c \{ (\text{pk}, \text{sk}) \leftarrow \text{QHE.Keygen}(1^\lambda), \\
  & \quad \text{ct} \leftarrow \text{QHE.Enc}_{\text{pk}}(x_1), \\
  & \quad |\phi\rangle \leftarrow \text{QHE.QEnc}_{\text{pk}}(|\psi_1\rangle) \} \quad \lambda, x_0, |\psi_0\rangle, x_1, |\psi_1\rangle
  \end{align*}
  
  where \(\lambda \in \mathbb{N}\), \(x_0, x_1 \in \{0, 1\}^{\ell(\lambda)}\) and \(|\psi_0\rangle, |\psi_1\rangle\) are \(\ell(\lambda)\)-qubit states.

- **Compactness:** There exists a polynomial \(\text{poly}(\cdot)\) s.t. for every quantum circuit \(C\) with \(\ell\) output qubits and an encryption of an input for \(C\), the output size of the evaluation algorithm is \(\ell \cdot \text{poly}(\lambda)\), where \(\lambda\) is the security parameter of the scheme.

- **Measurement-Preserving Homomorphism:** For every polynomial \(s(\cdot)\) there exists a negligible function \(\text{negl}(\cdot)\) such that for every \(\lambda \in \mathbb{N}\), size-\(s(\lambda)\) quantum circuit \(C\), input \((x, |\psi\rangle)\) for \(C\) which is comprised of a classical string \(x\) and quantum state \(|\psi\rangle\), subset \(M\) of the output qubits of \(C\), public and secret key pair \((\text{pk}, \text{sk}) \in \text{QHE.Keygen}(1^\lambda)\) and randomness strings \((r_x, r_{|\psi\rangle})\):
  
  \[
  \text{TD}\left(D_0, D_1\right) \leq \text{negl}(\lambda),
  \]
  
  where \(D_0, D_1\) are the distributions which are defined as follows:
\[ D_0 : \text{Compute } |\psi'\rangle \leftarrow C(x, |\psi\rangle), \text{measure the subset of qubits of } |\psi'\rangle \text{ which are in } M \text{ and output the obtained state.} \]

\[ D_1 : \]
\[ \star \text{Encrypt } ct = \text{QHE.Enc}_{pk}(x; r_x), |\phi\rangle = \text{QHE.QEnc}_{pk}(|\psi\rangle; r_{|\psi\rangle}). \]
\[ \star \text{Evaluate } |\tilde{\phi}\rangle \leftarrow \text{QHE.Eval}_{pk}(C, ct, |\phi\rangle). \]
\[ \star \text{Measure the } |M| \text{ packets of qubits that correspond to the output qubits in } M \text{ (by compactness, each packet is exactly of size } \text{poly}(\lambda)). \]
\[ \star \text{Decrypt the measured } |M| \text{ packets with } \text{QHE.Dec}_{sk}(-), \text{ and decrypt the rest of the qubits with } \text{QHE.QDec}_{sk}(-). \text{ Output the obtained state. } \]

**Instantiations.** Mahadev [Mah18a] shows how to build quantum FHE based on super-polynomial QLWE modulus and a circular security assumption with respect to a secret key and an additional trapdoor information. Brakerski [Bra18] subsequently shows how to construct quantum FHE based on polynomial QLWE modulus and a circular security assumption (analogous to the assumptions required for multi-key FHE in the classical setting). The above definition is more specific than the standard definition. The construction of Brakerski satisfies this more general definition. This follows readily from the main Theorem (4.1) in [Bra18].

### 2.2.4 Function-Hiding Secure Function Evaluation

We define two-message function evaluation protocols with statistical circuit privacy and quantum input privacy.

**Definition 2.12 (2-MESSAGE FUNCTION HIDING SFE).** A two-message secure function evaluation protocol (SFE.Gen, SFE.Enc, SFE.Eval, SFE.Dec) has the following syntax:

- \( dk \leftarrow \text{SFE.Gen}(1^\lambda) \): a probabilistic algorithm that takes a security parameter \( 1^\lambda \) and outputs a secret key \( dk \).
- \( ct \leftarrow \text{SFE.Enc}_{dk}(x) \): a probabilistic algorithm that takes a string \( x \in \{0, 1\}^* \), and outputs a ciphertext \( ct \).
- \( \hat{ct} \leftarrow \text{SFE.Eval}(C, ct) \): a probabilistic algorithm that takes a (classical) circuit \( C \) and a ciphertext \( ct \) and outputs an evaluated ciphertext \( \hat{ct} \).
- \( \hat{x} = \text{SFE.Dec}_{dk}(\hat{ct}) \): a deterministic algorithm that takes a ciphertext \( \hat{ct} \) and outputs a string \( \hat{x} \).

The scheme satisfies the following.

- **Perfect Correctness:** For any polynomial \( s(\cdot) \), for any \( \lambda \in \mathbb{N} \), size-\( s(\lambda) \) circuit \( C \) and input \( x \) for \( C \),
  \[
  \Pr \left[ \text{SFE.Dec}_{dk}(\hat{ct}) = C(x) \middle| \begin{array}{l}
  dk \leftarrow \text{SFE.Gen}(1^\lambda), \\
  ct \leftarrow \text{SFE.Enc}_{dk}(x), \\
  \hat{ct} \leftarrow \text{SFE.Eval}(C, ct)
  \end{array} \right] = 1 .
  \]

- **Quantum Input Privacy:** For every polynomial \( \ell(\cdot) \),
  \[
  \{ ct \mid dk \leftarrow \text{SFE.Gen}(1^\lambda), \} \approx_c \{ ct \mid dk \leftarrow \text{SFE.Gen}(1^\lambda), \}
  \{ ct \mid x_0, x_1 \} \approx_c \{ ct \mid x_0, x_1 \},
  \]
  where \( \lambda \in \mathbb{N} \) and \( x_0, x_1 \in \{0, 1\}^\ell(\lambda) \).
• **Statistical Circuit Privacy:** There exist unbounded algorithms, probabilistic \( \text{Sim} \) and deterministic \( \text{Ext} \) such that:
  
  - For every \( x \in \{0, 1\}^* \), \( ct \in \text{SFE.Enc}(x) \), the extractor outputs \( \text{Ext}(ct) = x \).
  - For any polynomial \( s(\cdot) \),
    \[
    \{\text{SFE.Eval}(C, ct^*)\}_{\lambda, C, ct^*} \approx_s \{\text{Sim}(1^\lambda, C(\text{Ext}(1^\lambda, ct^*)))\}_{\lambda, C, ct^*} ,
    \]
    where \( \lambda \in \mathbb{N} \), \( C \) is a \( s(\lambda) \)-size circuit, and \( ct^* \in \{0, 1\}^* \).

The next claim follows directly from the circuit privacy property, and will be used throughout the analysis.

**Claim 2.2** (Evaluations of Agreeing Circuits are Statistically Close). For any polynomial \( s(\cdot) \),
\[
\{\text{SFE.Eval}(C_0, ct^*)\}_{\lambda, C_0, ct^*} \approx_s \{\text{SFE.Eval}(C_1, ct^*)\}_{\lambda, C_0, C_1, ct^*} ,
\]
where \( \lambda \in \mathbb{N} \), \( C_0, C_1 \) are two \( s(\lambda) \)-size functionally-equivalent circuits, and \( ct^* \in \{0, 1\}^* \).

**Instantiations.** Such secure function evaluation schemes are known based on QLWE [OPCPC14, BD18].

### 2.2.5 Quantum Rewinding Lemma

We use Lemma 9 from [Wat09], which constructs a quantum algorithm for amplifying the success probability of quantum sampler circuits under some conditions.

**Lemma 2.1** (Lemma 9, [Wat09]). There is a quantum algorithm \( R \) that gets as input:

- A general quantum circuit \( Q \) with \( n \) input qubits that outputs a classical bit \( b \) and an additional \( m \) output qubits.
- An \( n \)-qubit state \( |\psi\rangle \).
- A number \( t \in \mathbb{N} \).

\( R \) executes in time \( t \cdot \text{poly}(|Q|) \) and outputs a distribution over \( m \)-qubit states \( D_\psi := R(Q, |\psi\rangle, t) \) with the following guarantees.

For an \( n \)-qubit state \( |\psi\rangle \), denote by \( Q_0 \) the conditional distribution of the output distribution \( Q(|\psi\rangle) \), conditioned on \( b = 0 \), and denote by \( p(\psi) \) the probability that \( b = 0 \). If there exist \( p_0, q \in (0, 1) \), \( \varepsilon \in (0, \frac{1}{2}) \) such that:

- Amplification executes for enough time: \( t \geq \frac{\log(1/\varepsilon)}{4p_0(1-p_0)} \).
- There is some minimal probability that \( b = 0 \): For every \( n \)-qubit state \( |\psi\rangle \), \( p_0 \leq p(\psi) \).
- \( p(\psi) \) is input-dependent, up to \( \varepsilon \) distance: For every \( n \)-qubit state \( |\psi\rangle \), \( |p(\psi) - q| < \varepsilon \), and
- \( q \) is closer to \( \frac{1}{2} \): \( p_0(1-p_0) \leq q(1-q) \),

then for every \( n \)-qubit state \( |\psi\rangle \),
\[
\text{TD}(Q_\psi, D_\psi) \leq 4\sqrt{\varepsilon} \frac{\log(1/\varepsilon)}{p_0(1-p_0)} .
\]

The exact wording in the Lemma from [Wat09] differs from the above in two manners. First, the original lemma states that for each circuit \( Q \) there exists an amplified circuit \( Q' \), but actually the proof of the Lemma proves that there is an algorithm \( R \) that on input \( Q \), executes an amplified version of \( Q \) (and thus the circuit implementation of \( R(Q) \) can be thought of as \( Q' \)). Second, the original lemma deals with unitary quantum circuits i.e. \( Q \) contains no measurement gates. By standard quantum circuit purification, it follows that the above formulation is equivalent to the analogous statement that includes only unitary circuits.
3 Constant-Round Zero-Knowledge Arguments for NP

In this section we construct a classical argument system for an arbitrary NP language \( L \), with a constant number of rounds, quantum soundness and quantum zero-knowledge (according to Definition 2.6).

Ingredients and notation:

- A non-interactive commitment scheme \( \text{Com} \).
- A CC obfuscator \( \text{Obf} \).
- A quantum fully homomorphic encryption scheme \( \text{QHE}.\text{Keygen}, \text{QHE}.\text{Enc}, \text{QHE}.\text{QEnc}, \text{QHE}.\text{Dec}, \text{QHE}.\text{QDec}, \text{QHE}.\text{Eval} \).
- A 2-message function-hiding secure function evaluation scheme \( \text{SFE}.\text{Gen}, \text{SFE}.\text{Enc}, \text{SFE}.\text{Eval}, \text{SFE}.\text{Dec} \).
- A 3-message WI proof \( \text{WI}.\text{P}, \text{WI}.\text{V} \) for \( L \in \text{NP} \).
- A 3-message sigma protocol \( \Sigma.\text{P}, \Sigma.\text{V} \) for \( L \in \text{NP} \).

We describe the protocol in Figure 1.

3.1 Quantum Soundness

**Proposition 3.1** (The Protocol is Sound). Let \( V \) be the verifier from Protocol 1. For any quantum polynomial-size prover \( P^* = \{P^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}} \), there exists a negligible function \( \mu(\cdot) \) such that for any security parameter \( \lambda \in \mathbb{N} \) and any \( x \in \{0, 1\}^\lambda \setminus L \),

\[
\Pr[\text{OUT}_V(P^*_\lambda, \rho_\lambda)(x) = 1] \leq \mu(\lambda) .
\]

**Proof.** Let \( P^* = \{P^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}} \) a polynomial-size quantum prover and let \( x = \{x_\lambda\}_{\lambda \in \mathbb{N}} \) be a sequence such that \( \forall \lambda \in \mathbb{N} : x_\lambda \in \{0, 1\}^\lambda \setminus L \). We prove soundness by a hybrid argument. We consider a series of hybrid processes with output over \( \{0, 1\} \), starting from \( \text{OUT}_V(P^*(\rho), V)(x) \) the output distribution of \( V \) in the interaction with \( P^* \). The proof will show that the probability to output 1 is negligible, which proves the soundness of the protocol.

We assume without the loss of generality that the first prover message is deterministic, and that the commitments \( \text{cmt}_1, \text{cmt}_2 \) it sends are both valid commitments and furthermore, there is some SFE secret key \( dk \in \text{SFE}.\text{Gen}(1^\lambda) \) such that \( \text{cmt}_2 \in \text{Com}(1^\lambda, dk) \). First note that if the above property is false, then the whole WI statement of the prover is false (because the first statement in \( P^* \)'s OR statement, that claims \( x \in L \), is always false in the case of a cheating prover).

This assumption is without the loss of generality because we can consider a new prover that chooses the first message (and quantum inner state at the end of this message) as the message that maximizes the probability that \( V \) outputs 1. If this message is such that \( \text{cmt}_1, \text{cmt}_2 \) are not consistent with the prover’s WI statement, then by the soundness of the proof that \( P^* \) gives, with overwhelming probability \( V \) outputs 0 and soundness already holds.

As a final note, observe that because \( \text{cmt}_1, \text{cmt}_2 \) are consistent with the prover’s WI statement, \( \text{cmt}_1 \) is necessarily a commitment to a non-witness \( u \notin \mathcal{R}_L(x) \), and denote by \( r_u \) a string s. t. \( \text{cmt}_1 = \text{Com}(1^\lambda, u; r_u) \).

Define the following hybrid distributions.

- \( \text{Hyb}_0 \) : The output distribution of \( \text{OUT}_V(P^*(\rho), V)(x) \).
• Hyb₁ : This hybrid process is identical to Hyb₀, with the exception that in step 4 (verifier WI), V uses the information \((u, r_u)\) as witness for its WI statement, instead of the witness that shows its transcript is explainable.

• Hyb₂ : This hybrid process is identical to Hyb₁, with the exception that V obtains dk, and when it gets the prover message ctp in step 2c, it performs the following check: If \(t = \text{SFE.Dec}_{dk}(ctp)\) then the process halts and outputs \(\perp\), otherwise the interaction carries on regularly.

• Hyb₃ : This hybrid process is identical to Hyb₂, except that in step 2d when the verifier responds with an SFE evaluation, instead of performing an SFE evaluation of \(\text{CC}[\text{Id}(\cdot), t, s]\), the verifier performs an SFE evaluation of \(C_{\perp}\) a circuit that always outputs \(\perp\).

• Hyb₄ : This hybrid process is identical to Hyb₃, except that the verifier does not perform the check at step 2c like described in Hyb₂.

• Hyb₅ : This hybrid process is identical to Hyb₄, except that in step 2b where V sends its first message, the reward value of the CC program \(\text{CC}\) it uses is \((r, 0)\) instead of \((r, \beta)\).

We now explain why each consecutive pair of the distributions above are statistically indistinguishable (recall that for a pair of distributions over a single bit, they are statistically indistinguishable iff they are computationally indistinguishable). We will then use the last process Hyb₅ to show that soundness follows from the soundness of the sigma protocol \((\Sigma.P, \Sigma.V)\).

• Hyb₀ \(\approx_s\) Hyb₁ : Follows from the witness indistinguishability property of the WI proof that the verifier gives.

• Hyb₁ \(\approx_s\) Hyb₂ : Follows from Claim 3.1, which says that the probability that ctp is an encryption of (the correct) \(t\) with the secret key dk (that is inside \(\text{cmt}_2\)) is negligible, and thus the erasure of such cases can’t be noticed by a distinguisher.

• Hyb₂ \(\approx_s\) Hyb₃ : As a basic explanation, this indistinguishability follows from the combination of the circuit privacy property of the SFE and the soundness of the WI proof that \(P^*\) gives.

As a fuller explanation, assume toward contradiction there’s a distinguisher \(D^*\) that tells the difference between the two distributions, and by an averaging argument, consider the transcript (and inner quantum state of \(P^*\)) generated at the end of step 2c (where \(P^*\) sends ctp), which maximizes \(D^*\’s\) distinguishability advantage - other than the prover’s ciphertext ctp, this transcript fixes \(t, s\), which in turn fix the circuit \(\text{CC}[\text{Id}(\cdot), t, s]\). We now consider three cases, and explain why we get a contradiction in each of them.

1. ctp \(\in\) \(\text{SFE.Enc}_{dk}(t)\): In this case, no matter what will be generated next in the transcript, the output will be \(\perp\) (by the check described in Hyb₂), thus it is impossible to distinguish the outputs of the two processes and we get a contradiction.

2. \(\exists y \in \{0, 1\}^λ \setminus \{t\}\) s.t. ctp \(\in\) \(\text{SFE.Enc}_{dk}(y)\): In this case, \(\perp = \text{CC}[\text{Id}(\cdot), t, s](y)\) and thus we get a contradiction by using the circuit privacy property of the SFE.

3. Else: In that case, either ctp is a ciphertext encrypted with some other SFE key \(dk’\), or it is not a valid ciphertext at all and in any case, it is not a valid ciphertext encrypted with the secret key dk. In that case, the WI statement of the prover is necessarily false, and thus a 1 output happens with at most negligible probability in both cases (by the soundness of the WI proof of \(P^*\)), thus the statistical distance between them is at most negligible, in contradiction.

• Hyb₃ \(\approx_s\) Hyb₄ : Follows from the same reasoning as in the indistinguishability \(\text{Hyb}_1 \approx_s \text{Hyb}_2\).
• $\text{Hyb}_4 \simeq_s \text{Hyb}_5$: Follows from the simulation property (obfuscation security) of the CC obfuscation scheme.

Now, assume toward contradiction that $P^*$ succeeds in making the verifier accept with some noticeable probability $\varepsilon(\lambda)$, that is, the probability for the output 1 in $\text{Hyb}_0$ is noticeable. $\text{Hyb}_0 \simeq_s \text{Hyb}_5$, and thus the probability for the output 1 in $\text{Hyb}_5$ is also noticeable. Finally, we get a contradiction to the soundness of the sigma protocol $(\Sigma.P, \Sigma.V)$, by using the prover sigma protocol messages from steps 3a, 6 as messages to convince a sigma protocol verifier $\Sigma.V$. Since the probability that the verifier $V$ is convinced in $\text{Hyb}_5$ is noticeable, and such verifier is convinced if and only if the sigma protocol verifier is convinced, we get our contradiction.

\begin{claim}
(Producing an SFE Encryption of $t$ with $dk$ is Hard). Let $P^* = \{P^*_\lambda, \rho^*_\lambda\}_{\lambda \in \mathbb{N}}$ be a quantum polynomial-size prover in Protocol 1, sending a deterministic first message $cmt_1$, $cmt_2$ where there exists $dk \in \text{SFE.Gen}(1^\lambda)$ s.t. $cmt_2 \in \text{Com}(1^\lambda, dk)$. Then there exists a negligible function $\mu(\cdot)$ such that the probability that $t = \text{SFE.Dec}_{dk}(ct_p)$ is bounded by $\mu(\lambda)$.
\end{claim}

\begin{proof}
The proof will be based on the security of the QFHE, and on the security of the CC obfuscation. We start with observing that the security of the QFHE implies that for every efficient quantum adversary $A^* = \{A^*_\lambda, \rho^*_\lambda\}_{\lambda \in \mathbb{N}}$, there’s a negligible function $\mu(\cdot)$ s.t. the probability that $A^*$ finds $t$ given $pk, ct \leftarrow \text{QHE.Enc}_{pk}(t)$ for a uniformly random $t \leftarrow \{0, 1\}^\lambda$, is bounded by $\mu(\lambda)$ - we will assume toward contradiction that our claim is false, that is, we assume that $P^*$ sends $ct_p$ s.t. $t = \text{SFE.Dec}_{dk}(ct_p)$ with noticeable probability (for infinitely many security parameters), and get a contradiction with the last claim about the hardness of finding a random encrypted $t$.

Using $P^*$ and the fact that $t = \text{SFE.Dec}_{dk}(ct_p)$ with noticeable probability, we now describe a (non-uniform) algorithm $A^*$ that finds $t$ given $pk, ct \leftarrow \text{QHE.Enc}_{pk}(t)$ for $t \leftarrow \{0, 1\}^\lambda$ and thus breaks the security of the QFHE. As part of the non-uniform advice of $A^*$, it will have the secret SFE key $dk$, which is fixed. Given $pk, ct \leftarrow \text{QHE.Enc}_{pk}(t)$, the algorithm $A^*$ will use the simulator $\text{Sim}^{CC}$ (from the simulation property of the CC obfuscation) and send to $P^*$ the following, as the protocol message sent at step 2b,

$$ pk, ct, \text{Sim}^{CC}(1^{\text{QHE.Dec|1}, 1^{\ell+|\beta|}, 1^{\lambda}}), $$

where $\ell$ is the randomness complexity of the QFHE key generation algorithm $\text{QHE.Keygen}$. $P^*$ will respond with $ct_p$, and $A^*$ uses $dk$ to output $\text{SFE.Dec}_{dk}(ct_p)$.

We now use the simulation property guarantee of the CC obfuscation: Note that the probability that $P^*$ outputs $ct_p$ s.t. $\text{SFE.Dec}_{dk}(ct_p) = t$ in the simulated setting, where $A^*$ sends $\text{Sim}^{CC}(1^{\text{QHE.Dec}_{sk}|1, 1^{sk+|\beta|}, 1^{\lambda}})$ instead of $CC$, is negligibly close to the probability that it outputs $ct_p$ s.t. $\text{SFE.Dec}_{dk}(ct_p) = t$ in the regular setting where it gets $CC$ - this is due to the security of the CC obfuscator. Because we know that in the regular interaction, $P^*$ sends $ct_p$ s.t. $t = \text{SFE.Dec}_{dk}(ct_p)$ with a noticeable probability, this implies that so does $A^*$, in contradiction.
\end{proof}

3.2 Quantum Zero-Knowledge

We construct a quantum polynomial-time universal simulator $\text{Sim}$ that for a quantum verifier $V^*$, an arbitrary quantum auxiliary input $\rho$ and an instance in the language $x \in L$, simulates the output distribution of the verifier after the real interaction, $\text{OUT}_{V^*}(P, V^*(\rho))(x)$. Throughout this section, a malicious verifier $V^*$ is modeled as a family of non-uniform quantum circuits with auxiliary quantum input, consistently with the rest of the paper.

**High-Level Description of Simulation.** Our simulation is composed as follows. We first describe two simulators, $\text{Sim}_a$ and $\text{Sim}_{na}$ that try to simulate different types of transcripts, specifically, $\text{Sim}_a$ will try to
simulate an aborting interaction, and $\text{Sim}_{\text{na}}$ will try to simulate a non-aborting interaction. By "aborting interaction" and "non-aborting interaction" we formally mean the following:

- **An aborting interaction** is one where the verifier $V^*$ either aborts before the end of step 5 (prover WI), or fails to prove its WI statement in step 4.
- **A non-aborting interaction** is one that is not aborting. More precisely, a non-aborting interaction is one where the verifier did not abort before the end of step 5 (prover WI), and also succeeded in proving its WI statement in step 4.

Our next step will be to describe a unified simulator $\text{Sim}_{\text{comb}}$ that randomly chooses $b \leftarrow \{a, \text{na}\}$ and then uses $\text{Sim}_b$ to simulate the interaction. We will prove that on input $(x, V^*, \rho)$, $\text{Sim}_{\text{comb}}$ outputs a quantum state that is computationally indistinguishable from $\text{OUT}_{V^*}(P, V^*(\rho))(x)$, with the following exception: $\text{Sim}_{\text{comb}}$ outputs a quantum state $\tilde{\text{OUT}}$ that indistinguishable from the real verifier output $\text{OUT}_{V^*}(P, V^*(\rho))(x)$ conditioned on $\text{OUT} \neq \text{Fail}$. Furthermore $\text{OUT} \neq \text{Fail}$ with probability negligibly close to $1/2$. In other words, $\text{Sim}_{\text{comb}}$ is going to succeed simulating only with probability (negligibly close to) $1/2$.

We further show that $\text{Sim}_{\text{comb}}$ satisfies the required conditions for applying Watrous’ quantum rewinding lemma so that the success probability can be amplified from $\approx 1/2$ to $\approx 1$.

**The Actual Proof.** We start by describing the above mentioned simulators.

$\text{Sim}_a(x, V^*, \rho)$:

1. **Simulation of Initial Commitments and Verifier Message:**
   - (a) $\text{Sim}_a$ computes $d_k \leftarrow \text{SFE.Gen}(1^\lambda)$ and sends to $V^*$ the commitments $\text{cmt}_1 \leftarrow \text{Com}(1^\lambda, 0|w|)$, $\text{cmt}_2 \leftarrow \text{Com}(1^\lambda, d_k)$.
   - (b) $V^*$ sends $\text{pk}, \text{ct}_{V^*}, \tilde{\text{CC}}$.

2. **Trying to get an Abort:** $\text{Sim}_a$ interacts with $V^*$ as the honest prover $P$ until the end of step 5 of the original protocol, with exactly 2 differences:
   - The message $\alpha$ at step 3a is generated by the sigma protocol simulator $\alpha \leftarrow \Sigma.S(x, 0|\beta|)$, and not by the sigma protocol prover.
   - At step 5, the witness used to prove the WI statement is for the second statement in the OR expression (that the commitments $\text{cmt}_1, \text{cmt}_2$ are valid and consistent), and not the first (that $x \in \mathcal{L}$).

3. **Simulation Verdict:** If at some point $V^*$ aborts or fails in its WI proof, $\text{Sim}_a$ outputs the aborting verifier’s output. Otherwise, $\text{Sim}_a$ outputs Fail.

$\text{Sim}_{\text{na}}(x, V^*, \rho)$:

1. **Simulation of Initial Commitments and Verifier Message:**
   - (a) $\text{Sim}_{\text{na}}$ computes $d_k \leftarrow \text{SFE.Gen}(1^\lambda)$ and sends to $V^*$ the commitments $\text{cmt}_1 \leftarrow \text{Com}(1^\lambda, 0|w|)$, $\text{cmt}_2 \leftarrow \text{Com}(1^\lambda, d_k)$.
   - (b) $V^*$ sends $\text{pk}, \text{ct}_{V^*}, \tilde{\text{CC}}$.

2. **Non-Black-Box Extraction Attempt:**
(a) $\text{Sim}_{\text{na}}$ computes
\[
  r_1 \leftarrow \{0, 1\}^*, \quad \text{ct}_t^{\text{SFE}} = \text{QHE.Eval}_{\text{pk}}(\text{SFE.Enc}_{\text{dk}}(r_1), \text{ct}_V^{\ast}).
\]
$\text{Sim}_{\text{na}}$ also encrypts $\rho^{(1)}$, the inner (quantum) state of the verifier after its first message:
\[
  \text{ct}_{\rho^{(1)}} \leftarrow \text{QHE.QEnc}_{\text{pk}}(\rho^{(1)}).
\]
(b) $\text{Sim}_{\text{na}}$ performs a quantum homomorphic evaluation of the verifier’s response. It computes,
\[
  \left(\text{ct}_s^{\text{SFE}}, \text{ct}_{\rho^{(2)}}\right) \leftarrow \text{QHE.Eval}_{\text{pk}}\left(V^{\ast}, (\text{ct}_t^{\text{SFE}}, \text{ct}_{\rho^{(1)}})\right).
\]
(c) $\text{Sim}_{\text{na}}$ computes $\text{ct}_s \leftarrow \text{QHE.Eval}_{\text{pk}}(\text{SFE.Dec}_{\text{dk}}(\cdot), \text{ct}_s^{\text{SFE}})$, and then computes $(r, \beta') = \text{CC}(\text{ct}_s)$.
(d) $\text{Sim}_{\text{na}}$ checks validity: $(\text{pk}', \text{sk}) = \text{QHE.Keygen}(1^{\lambda}; r)$, if $\text{pk}' \neq \text{pk}$ then it halts simulation and outputs $\text{Fail}$. Otherwise, $\text{Sim}_{\text{na}}$ obtains the inner state of $V^{\ast}$ by decryption: $\rho^{(2)} \leftarrow \text{QHE.QDec}_{\text{sk}}(\text{ct}_{\rho^{(2)}})$. Additionally, $\text{Sim}_{\text{na}}$ simulates the missing transcript (for the verifier to later prove that its messages were explainable): for the prover message at step 2c it inserts $\text{ct}_t = \text{SFE.Enc}_{\text{dk}}(t; r_1)$, and for the verifier message at step 2d it inserts $\text{ct}_s = \text{QHE.Dec}_{\text{sk}}(\text{ct}_s^{\text{SFE}})$.

3. Sigma Protocol Messages Simulation:
   (a) $\text{Sim}_{\text{na}}$ executes the sigma protocol simulator $(\alpha, \gamma) \leftarrow \Sigma.S(x, \beta')$ and sends $\alpha$ to $V^{\ast}$.
   (b) $V^{\ast}$ returns $\beta$.

4. WI Proof by the Malicious Verifier: $\text{Sim}_{\text{na}}$ takes the role of the honest prover $P$ in the WI proof $V^{\ast}$ gives. If $V^{\ast}$ fails to prove the statement, the simulation fails and the output is $\text{Fail}$.

5. Simulation of the Prover’s WI Proof and Information Reveal: $\text{Sim}_{\text{na}}$ gives $V^{\ast}$ a WI proof using the witness that shows $\text{cmt}_1$, $\text{cmt}_2$ are both valid commitments (and that $\text{cmt}_2$ is a commitment to the SFE key $\text{dk}$ used in step 2c). After the proof, $\text{Sim}_{\text{na}}$ sends $\gamma$ to $V^{\ast}$.

6. Simulation Verdict: If $V^{\ast}$ completed interaction without aborting and gave a convincing WI proof, $\text{Sim}_{\text{na}}$ outputs the verifier’s output. Otherwise, $\text{Sim}_{\text{na}}$ outputs $\text{Fail}$.

$\text{Sim}_{\text{comb}}(x, V^{\ast}, \rho):$ Sample $b \leftarrow \{0, 1\}$ and execute $\text{Sim}_b(x, V^{\ast}, \rho)$.

$\text{Sim}(x, V^{\ast}, \rho):$

1. Generate the circuit $\text{Sim}_{\text{comb}, x, V^{\ast}}$, which is the circuit implementation of $\text{Sim}_{\text{comb}}$, with hardwired input $x, V^{\ast}$, that is, the only input to $\text{Sim}_{\text{comb}, x, V^{\ast}}$ is the quantum state $\rho$.
2. Let $R$ be the algorithm from Lemma 2.1. The output of the simulation is $R(\text{Sim}_{\text{comb}, x, V^{\ast}}, \rho, \lambda)$.

Proof of Simulation Validity. We now turn to prove that the simulated output $\text{Sim}(x, V^{\ast}, \rho)$ is computationally indistinguishable from $\text{OUT}_{V^{\ast}}(P, V^{\ast}(\rho))(x)$. This is done in several steps:

1. Simulating aborting interactions: Let $V^{\ast}_a$ be the augmented verifier that is identical to $V^{\ast}$, with the exception that if $V^{\ast}$ does not abort, $V^{\ast}_a$ outputs $\text{Fail}$. Then the output of $\text{Sim}_a$ is indistinguishable from the output of $V^{\ast}_a$ in a real interaction.
2. **Simulating non-aborting interactions:** Let $V_{na}^*$ be the augmented verifier that is identical to $V^*$, with the exception that if $V^*$ aborts, $V_{na}^*$ outputs Fail. Then the output of $Sim_{na}$ is indistinguishable from the output of $V_{na}^*$ in a real interaction.

3. The above two statements imply:
   1. $Sim_{comb} \neq Fail$ with probability negligibly close to $\frac{1}{2}$, for every verifier and auxiliary input $\rho$.
   2. The output of $V^*$ in a real interaction is indistinguishable from the output of $Sim_{comb}$ conditioned on $Sim_{comb} \neq Fail$.

These in turn imply that we can use Watrous’ quantum rewinding lemma in order to amplify $Sim_{comb}$ into a full-fledged simulator $Sim$.

**Proposition 3.2 (Similarity of Aborting Part).** Let $V^* = \{V^*_\alpha, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ a polynomial-size quantum verifier, and let $OUT_{V^*_a}$ be the verifier’s output at the end of protocol such that if $V^*$ does not abort, the output is $Fail$. Then, $$\{OUT_{V^*_a}(P(w), V^*_\lambda(\rho_\lambda))(x)\}_{\lambda, x, w} \approx_c \{Sim_{a}(x, V^*_\lambda, \rho_\lambda)\}_{\lambda, x, w},$$ where $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0, 1\}^\lambda, w \in \mathcal{R}_\mathcal{L}(x)$.

**Proof.** We prove the claim by a hybrid argument, specifically, we consider hybrid distributions, all of which will be computationally indistinguishable.

- **Hyb$_0$** : The output distribution of $Sim_a$.
- **Hyb$_1$** : This hybrid process is identical to Hyb$_0$, with the exception that when the simulator gives a WI proof in the simulation, it uses the witness $w$ in the proof, that proves the first statement in the OR statement ($x \in \mathcal{L}$) rather then the second statement.
- **Hyb$_2$** : This hybrid process is identical to Hyb$_1$, with the exception that $\text{cmt}_1$ is a commitment to $w$ rather than to $0^{|w|}$.
- **Hyb$_3$** : This hybrid process is identical to Hyb$_2$, with the exception that the message $\alpha$ that the simulator sends to $V^*$ is generated by the actual sigma protocol $(\alpha, \tau) \leftarrow \Sigma.P_1(x, w)$, and not by the sigma protocol simulator $\Sigma.S(x, 0^{|\beta|})$. Note that this process is exactly $OUT_{V^*_a}(P(w), V^*(\rho))(x)$.

It is left to reason about the indistinguishability between each two subsequent hybrids.

- **Hyb$_0$ $\approx_c$ Hyb$_1$** : Follows from the witness-indistinguishability property of the WI proof that the simulator gives (as the prover) in step 5 of the protocol.
- **Hyb$_1$ $\approx_c$ Hyb$_2$** : Follows from the hiding property of the commitment $\text{cmt}_1$.
- **Hyb$_2$ $\approx_c$ Hyb$_3$** : Follows from Claim 2.1.

$\square$

**Proposition 3.3 (Similarity of Non-Aborting Part).** Let $V^* = \{V^*_\alpha, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ a polynomial-size quantum verifier, and let $OUT_{V^*_na}$ be the verifier’s output at the end of protocol such that if $V^*$ aborts, the output is $Fail$. Then, $$\{OUT_{V^*_na}(P(w), V^*_\lambda(\rho_\lambda))(x)\}_{\lambda, x, w} \approx_c \{Sim_{na}(x, V^*_\lambda, \rho_\lambda)\}_{\lambda, x, w},$$ where $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0, 1\}^\lambda, w \in \mathcal{R}_\mathcal{L}(x)$. 

23
Proof. We prove the claim by a hybrid argument, specifically, we consider hybrid distributions, all of which will be computationally indistinguishable.

- **Hyb₀**: The output distribution of \( \text{Sim}_{na} \).

- **Hyb₁**: This hybrid process is identical to \( \text{Hyb₀} \), with the exception that when the simulator gives a WI proof in the simulation, it uses the witness \( w \) in the proof, that proves the first statement in the OR statement \( x \in L \) rather then the second statement.

- **Hyb₂**: This hybrid process is identical to \( \text{Hyb₁} \), with the exception that \( \text{cmt₁} \) is a commitment to the witness \( w \) rather than to \( 0^{|w|} \), and \( \text{cmt₂} \) is a commitment to \( 0^{|dk|} \) rather than to the generated SFE key \( dk \leftarrow \text{SFE.Gen}(1^\lambda) \).

- **Hyb₃**: This hybrid process is identical to \( \text{Hyb₂} \), with the exception that if the verifier’s message \( \beta \) from part 3b of the simulation does not match the extracted \( \beta' \) from part 2c of the simulation, the process halts on the spot and outputs Fail₁.

- **Hyb₄**: This hybrid process is identical to \( \text{Hyb₃} \), with the exception that in parts 3a, 5 where the simulator sends \( \alpha \) and \( \gamma \), instead of computing \( \alpha, \gamma \) using \( \Sigma.S \), it computes \( (\alpha, \tau) \leftarrow \Sigma.P₁(x, w) \) and \( \gamma \leftarrow \Sigma.P₃(\beta, \tau) \).

- **Hyb₅**: This hybrid process is identical to \( \text{Hyb₄} \), with the exception that it does not perform the check described in \( \text{Hyb₃} \), that is, even if the extracted challenge and sent challenge are distinct \( \beta' \neq \beta \), the process carries on regularly.

- **Hyb₆**: At this point in our series of hybrid distributions, we do not use the extracted challenge \( \beta' \), and we would like to move to a process that does not perform extraction. The current hybrid will still perform extraction, but will not use the extracted information. This hybrid process is identical to \( \text{Hyb₅} \), with the changes described next. If the first verifier message is not explainable then the process chooses to fail and outputs Fail. If the first verifier message is explainable, note that it fixes a public and secret key pair \( (pk, sk) = \text{QHE.Keygen}(1^\lambda; r) \), and a string \( s \in \{0, 1\}^\lambda \) hidden inside the CC program \( \text{CC} \). In that case, the process acts like \( \text{Hyb₅} \), except that at the end of step 2b of the simulation, the process inefficiently obtains \( sk \) and uses it to decrypt \( \left( \text{ct}_{s}^{\text{SFE}}, \text{ct}_{s}^{(2)} \right) \), instead of using the program \( \tilde{\text{CC}} \) to get \( sk \). The process also inefficiently obtains \( s \) and performs a check: if \( s \neq \text{SFE.Dek}_{dk}(\text{QHE.Dek}_{pk}(\text{ct}_{s}^{\text{SFE}})) \) then the process fails and outputs Fail₁, and otherwise continues the simulation regularly as in the rest of \( \text{Hyb₅} \).

- **Hyb₇**: This process will get rid of extraction altogether and will not perform the homomorphic evaluation of the verifier’s response. This distributions is the output of a process that acts like \( \text{Hyb₆} \), with the exception that if the first verifier message is explainable (in particular, \( ct_V \) is a QFHE encryption of some \( t \in \{0, 1\}^\lambda \)), then as the prover message from step 2c of the protocol, the process sends \( \text{SFE.Enc}_{dk}(t) \). If at step 2d the verifier responds with \( ct \) s.t. \( s = \text{SFE.Dek}_{dk}(ct) \) then the simulation continues regularly as in \( \text{Hyb₆} \), and otherwise the process fails and outputs Fail₁.

- **Hyb₈**: Like the previous two processes, this process is also inefficient. This hybrid process is identical to \( \text{Hyb₇} \), with the exception that it does not perform the check on the verifier’s response \( ct \), and continues regularly either way, even when \( s \neq \text{SFE.Dek}_{dk}(ct) \).

- **Hyb₉**: We now go back to an efficient hybrid process. This hybrid process is identical to \( \text{Hyb₈} \), with the exception that instead of performing the inefficient check on the verifier’s first message from step 1b (and then either halting and outputting Fail₁, or sending \( \text{SFE.Enc}_{dk}(t) \) to \( V^* \)), the
process always sends $\text{SFE.Enc}_{dk}(0^\lambda)$ to $V^\star$, and continues simulation regularly. Observe that this process is exactly $\text{OUT}_{\text{in}}(P(w), V^\star(\rho))(x)$.

We now prove that each pair of consecutive distributions are computationally indistinguishable, and our proof is finished.

- **$\text{Hyb}_0 \approx_c \text{Hyb}_1$** : This indistinguishability follows from the witness-indistinguishability property of the WI proof that the simulator gives in step 5 of the simulation.

- **$\text{Hyb}_1 \approx_c \text{Hyb}_2$** : This indistinguishability follows from the hiding of the commitments $\text{cmt}_1$, $\text{cmt}_2$ that the simulator gives in step 1a of the simulation.

- **$\text{Hyb}_2 \approx_c \text{Hyb}_3$** : The indistinguishability follows from the perfect correctness of both the CC obfuscation and the SFE schemes, along with the soundness of the WI proof. Assume toward contradiction that the two distributions are distinguishable and fix, by an averaging argument, the partial transcript $T'$ that is generated at the end of step 3b of the simulation, which maximizes distinguishability. We consider two cases for $T'$:

  - $T'$ is explainable. In that case it follows from the perfect correctness of the CC obfuscation and the perfect correctness of the SFE evaluation, that the extracted $\beta'$ and the sent $\beta$ are necessarily equal, and the processes are identical (and have statistical distance of 0), in contradiction.

  - $T'$ is not explainable. In that case, recall that $\text{cmt}_1$ is a commitment to a witness and thus the statement in the verifier’s WI proof is necessarily false. By the soundness of the WI proof, $V^\star$ will fail in proving the statement with overwhelming probability, which implies that with the same probability the output in the process $\text{Hyb}_3$ is $\text{Fail}$. Because with *at least* the same probability, the output in $\text{Hyb}_3$ is also $\text{Fail}$, the contradiction follows.

- **$\text{Hyb}_3 \approx_c \text{Hyb}_4$** : This indistinguishability follows from the special zero-knowledge property of the sigma protocol.

- **$\text{Hyb}_4 \approx_c \text{Hyb}_5$** : The statistical indistinguishability follows from the exact same reasoning that explains why distributions $\text{Hyb}_2 \approx_c \text{Hyb}_3$.

- **$\text{Hyb}_5 \approx_c \text{Hyb}_6$** : This indistinguishability will follow from the perfect correctness of the CC obfuscation, the statistical correctness of the QFHE and from the soundness of the WI proof that $V^\star$ gives. Formally, assume toward contradiction that the two distributions are distinguishable and fix, by an averaging argument, the partial transcript $T'$ and inner quantum state $\rho^{(1)}$ of $V^\star$ generated at the end of step 1b of the simulation, that maximize the distinguishability. Denote by $\text{Hyb}_5$, $\text{Hyb}_6$ the distributions that carry on from the point that $T'$, $\rho^{(1)}$ are fixed, according to $\text{Hyb}_5$, $\text{Hyb}_6$, respectively. Consider two cases for $T'$:

  - $T'$ is not explainable. In that case, $\text{Hyb}_6$ outputs $\text{Fail}$ with probability 1, and by the soundness of the WI proof of the verifier, the proof is going to fail with overwhelming probability (in the process $\text{Hyb}_5$) and with at least the same probability the output is going to be $\text{Fail}$, and the two distributions will have at most negligible statistical distance, in contradiction.

  - $T'$ is explainable, which means that the verifier’s first message fixes $t$, $s$, $sk$, and also $r_t$ the QFHE encryption randomness s.t. $ct_{v^\star} = \text{QHE.Enc}_{pk}(t, r_t)$. Consider the quantum circuit $C$ that for input $(t, \rho^{(1)})$, encrypts $ct_t \leftarrow \text{SFE.Enc}_{dk}(t)$, executes $(\hat{ct}, \rho^{(2)}) \leftarrow V^\star(\text{ct}_t, \rho^{(1)})$, decrypts $s' = \text{SFE.Dec}_{dk}(\hat{ct})$ and outputs $s', \hat{ct}, \rho^{(2)}$. Now, observe the following about the distributions $\text{Hyb}_5$, $\text{Hyb}_6$. 

---
* Hyb$_5$ can be described by the following process: Encrypt $ct_{\nu'} = \text{QHE.Enc}_{pk}(t; r_t)$, $ct'_{\rho(1)} \leftarrow \text{QHE.QEnc}_{pk}(\rho^{(1)})$, perform homomorphic evaluation of the circuit $C$, and then decrypt with $sk$ to get $(s', \hat{c}, \rho^{(2)})$.

If $s' \neq s$ then output Fail, otherwise carry on the simulation as in Hyb$_5$. The fact that Hyb$_5$ can be described by this process follows from the fact that by the perfect correctness of the CC obfuscation, if the first verifier message is explainable then CC indeed executes the decryption circuit $\text{QHE.Dec}_{sk}(\cdot)$ s.t. if the result was $s$, it outputs the QFHE key-generation randomness $r$ (which in turn yields $sk$), and if the result wasn’t $s$, CC necessarily yields $\bot$.

* Hyb$_6$ can be described by the following process: the exact same homomorphic evaluation process as described above, except that after getting the output $(s', \hat{c}, \rho^{(2)})$, the check is that $s = \text{SFE.Dec}_{dk}(\hat{c})$, and if the check fails the output is Fail, and if the check succeeds then the process continues simulation regularly in the exact same way as in Hyb$_5$.

The above descriptions of Hyb$_5$, Hyb$_6$ imply that the statistical distance between them is bounded by the probability that the check in one process fails and in the other it succeeds, which in turn bounded by the probability that $s' \neq \text{SFE.Dec}_{dk}(\hat{c})$. The point is, due to the fact that the SFE decryption algorithm is deterministic, it is always the case when evaluating the circuit $C$ (out in the open, not under homomorphic evaluation) we have $s' = \text{SFE.Dec}_{dk}(\hat{c})$.

It follows by the statistical correctness of the QFHE that the probability that the evaluated $s', \hat{c}$ are s.t. $s' \neq \text{SFE.Dec}_{dk}(\hat{c})$, and thus the bound on the statistical distance between Hyb$_5$, Hyb$_6$, in contradiction.

* Hyb$_6 \approx_s$ Hyb$_7$: This indistinguishability follows directly from the statistical correctness of the QFHE. Assume toward contradiction that the two distributions are distinguishable and fix, by an averaging argument, the partial transcript $T'$ and inner quantum state $\rho^{(1)}$ of $V^*$ generated at the end of step 1b of the simulation, that maximize the distinguishability. Denote by Hyb$_6$, Hyb$_7$ the distributions that carry on from the point that $T'$, $\rho^{(1)}$ are fixed, according to Hyb$_6$, Hyb$_7$, respectively. Consider two cases for $T'$:

- $T'$ is not explainable. In that case both processes act the same and output Fail, and are indistinguishable.

- $T'$ is explainable, which means that the verifier’s first message fixes $t, s, sk$, and also $r_t$ the QFHE encryption randomness s.t. $ct_{\nu'} = \text{QHE.Enc}_{pk}(t; r_t)$. In that case, recall the circuit $C$ from the above proof of the indistinguishability Hyb$_5 \approx_s$ Hyb$_6$, and observe the following about the distributions Hyb$_6$, Hyb$_7$.

    * The distribution Hyb$_6$ can be described by the following process: Encrypt $ct_{\nu'} = \text{QHE.Enc}_{pk}(t; r_t)$, $ct'_{\rho(1)} \leftarrow \text{QHE.QEnc}_{pk}(\rho^{(1)})$, perform homomorphic evaluation of the circuit $C$, and then decrypt with $sk$ to get $(s', \hat{c}, \rho^{(2)})$. If $s = \text{SFE.Dec}_{dk}(\hat{c})$ then process continues simulation regularly as in Hyb$_6$, and otherwise fails and outputs Fail.

    * The distribution Hyb$_7$ can be described by the following process: Instead of encrypting $t, \rho^{(1)}$ and computing $C$ under homomorphic evaluation (and then decrypting), we simply execute $(\hat{c}, \rho^{(2)}) \leftarrow C(t, \rho^{(1)})$ in the clear. The process continues in the exact same way as described after the homomorphic evaluation in Hyb$_6$; If $s = \text{SFE.Dec}_{dk}(\hat{c})$ then process continues simulation regularly, and otherwise fails and outputs Fail.

The above implies that the only difference between the two processes is the fact that in Hyb$_6$ we execute $C$ under homomorphic evaluation, and in Hyb$_7$ we execute $C$ in the clear. By
Corollary 3.1 (Probabilities to Abort are Negligibly Close Over Different Cases). For a quantum auxiliary input \( \rho \), instance in the language \( x \in \{0,1\}^\lambda \cap L \) and witness \( w \in R_L(x) \), define the following probabilities.

- \( a(x, \rho) \): The probability that in the simulation \( Sim_a(x, V^*, \rho) \), the verifier \( V^* \) aborted before the end of step 3 where the simulator simulates the Prover's WI proof, or failed to prove its WI statement in step 4 (i.e. the simulation of \( Sim_a(x, V^*, \rho) \) was aborting).

- \( b(x, \rho) \): The probability that in the simulation \( Sim_{na}(x, V^*, \rho) \), the verifier \( V^* \) aborted before the end of step 5 where the simulator simulates the Prover's WI proof, or failed to prove its WI statement in step 4 (i.e. the simulation of \( Sim_{na}(x, V^*, \rho) \) was aborting).

- \( c(x, \rho, w) \): The probability that the interaction \( \langle P(w), V^*(\rho) \rangle(x) \) was aborting.

There exists a negligible function \( \negl(\cdot) \) s.t. for every sequences \( \rho = \{\rho_\lambda\}_{\lambda \in \mathbb{N}}, x = \{x_\lambda\}_{\lambda \in \mathbb{N}}, w = \{w_\lambda\}_{\lambda \in \mathbb{N}} \) where,

- \( \forall \lambda \in \mathbb{N} : \rho_\lambda \) is a \( \lambda^c \)-size quantum state (for some constant \( c \in \mathbb{N} \)),

- \( \forall \lambda \in \mathbb{N} : x_\lambda \in \{0,1\}^\lambda \cap L \),

- \( \forall \lambda \in \mathbb{N} : w_\lambda \in R_L(x_\lambda) \),

we have

\[
\forall \lambda \in \mathbb{N} : |a(x_\lambda, \rho_\lambda) - b(x_\lambda, \rho_\lambda)|, |b(x_\lambda, \rho_\lambda) - c(x_\lambda, \rho_\lambda, w_\lambda)|, |c(x_\lambda, \rho_\lambda, w_\lambda) - a(x_\lambda, \rho_\lambda)| \leq \negl(\lambda) .
\]
Proposition 3.4

indistinguishable from the real interaction.

We next prove that conditioned on succeeding, the output distribution of the simulator

where the last inequality is due to Corollary 3.1.

Proof.

Corollary 3.2 (Success Probability of $\text{Sim}_{\text{comb}}$ is Input-Oblivious). For every quantum verifier $V^* = \{V^*_\lambda\}_{\lambda \in \mathbb{N}}$ there exists a negligible function $\text{negl}(\cdot)$ such that for every instance in the language $x = \{x_\lambda\}_{\lambda \in \mathbb{N}}$ and quantum auxiliary input $\rho = \{\rho_\lambda\}_{\lambda \in \mathbb{N}}$ for the verifier, we have

$$\forall \lambda \in \mathbb{N} : \left| \Pr[\text{The simulation } \text{Sim}_{\text{comb}}(x_\lambda, V^*_\lambda, \rho_\lambda) \text{ succeeds}] - \frac{1}{2} \right| \leq \text{negl}(\lambda).$$

Proof.

$$\forall \lambda \in \mathbb{N} : \left| \Pr[\text{The simulation } \text{Sim}_{\text{comb}}(x_\lambda, V^*_\lambda, \rho_\lambda) \text{ succeeds}] - \frac{1}{2} \right|$$

$$= \left| \frac{1}{2} \cdot \Pr[\text{The simulation } \text{Sim}_a(x_\lambda, V^*_\lambda, \rho_\lambda) \text{ succeeds}] + \frac{1}{2} \cdot \Pr[\text{The simulation } \text{Sim}_{na}(x_\lambda, V^*_\lambda, \rho_\lambda) \text{ succeeds}] - \frac{1}{2} \right|$$

$$= \frac{1}{2} \cdot a(x_\lambda, \rho_\lambda) + \frac{1}{2} \cdot (1 - b(x_\lambda, \rho_\lambda)) - \frac{1}{2}$$

$$= \frac{1}{2} \cdot |a(x_\lambda, \rho_\lambda) - b(x_\lambda, \rho_\lambda)| \leq \text{negl}(\lambda),$$

where the last inequality is due to Corollary 3.1.

We next prove that conditioned on succeeding, the output distribution of the simulator $\text{Sim}_{\text{comb}}$ is indistinguishable from the real interaction.

Proposition 3.4 (The Output of a Successful $\text{Sim}_{\text{comb}}$ is Indistinguishable from Real Interaction). Let $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ be a polynomial-size quantum verifier. For $x \in \mathcal{L}$, let $\text{Sim}_{\text{comb}}(x, V^*, \rho)$ denote the conditional distribution of $\text{Sim}_{\text{comb}}(x, V^*, \rho)$, conditioned on the simulation being successful. Then,

$$\{\text{OUT}_{V^*}(P(w), V^*_\lambda(\rho_\lambda))(x)\}_{\lambda \in \mathbb{N}} \approx \{\text{Sim}_{\text{comb}}(x, V^*_\lambda, \rho_\lambda)\}_{\lambda \in \mathbb{N}},$$

$$\text{for } x \in \mathcal{L} \cap \{0, 1\}^\lambda, \text{ and } w \in R_\mathcal{L}(x).$$

Proof. Denote the following conditional distributions.

- $A_{\text{Sim}} = \{A_{\text{Sim}, \lambda}\}_{\lambda \in \mathbb{N}} : A$ conditional distribution of $\text{Sim}_a(x, V^*, \rho)$, conditioned on that the output is not $\text{Fail}$ (might be an empty distribution, if $a(x, \rho) = 0$).

- $S_{\text{Sim}} = \{S_{\text{Sim}, \lambda}\}_{\lambda \in \mathbb{N}} : S$ conditional distribution of $\text{Sim}_{na}(x, V^*, \rho)$, conditioned on that the output is not $\text{Fail}$ (might be an empty distribution, if $b(x, \rho) = 1$).

- $A_{(P, V^*)} = \{A_{(P, V^*), \lambda}\}_{\lambda \in \mathbb{N}} : A$ conditional distribution of $\text{OUT}_{V^*}(P, V^*)$ (from 3.2), conditioned on that the output is not $\text{Fail}$ (might be an empty distribution, if $c(x, \rho, w) = 0$).

- $S_{(P, V^*)} = \{S_{(P, V^*), \lambda}\}_{\lambda \in \mathbb{N}} : S$ conditional distribution of $\text{OUT}_{V^*}(P, V^*)$ (from 3.3), conditioned on that the output is not $\text{Fail}$ (might be an empty distribution, if $c(x, \rho, w) = 1$).
Observe that the distribution \( \widetilde{\text{Sim}}_{\text{comb}}(x, V^*, \rho) \) is the distribution generated by outputting a sample from \( A_{\text{Sim}} \) with probability \( \frac{a(x, \rho)}{1 + a(x, \rho) - b(x, \rho)} \), and a sample from \( S_{\text{Sim}} \) with probability \( \frac{1 - b(x, \rho)}{1 + a(x, \rho) - b(x, \rho)} \). Additionally, observe that the distribution \( \text{OUT}_{V^*}(P(w), V^*(\rho))(x) \) is the distribution generated by outputting a sample from \( A_{(p, V^*)} \) with probability \( c(x, \rho, w) \) and from \( S_{(p, V^*)} \) with probability \( 1 - c(x, \rho, w) \). We will show that the two distributions are computationally indistinguishable by a hybrid argument. Consider the following distributions.

- **Hyb_0**: The distribution \( \widetilde{\text{Sim}}_{\text{comb}}(x, V^*, \rho) \).
- **Hyb_1**: Same as in Hyb_0, with the exception that instead of sampling from \( A_{\text{Sim}} \) with probability \( \frac{a(x, \rho)}{1 + a(x, \rho) - b(x, \rho)} \) (and from \( S_{\text{Sim}} \) with probability \( \frac{1 - b(x, \rho)}{1 + a(x, \rho) - b(x, \rho)} \)), it samples from \( A_{\text{Sim}} \) with probability \( a(x, \rho) \) (and from \( S_{\text{Sim}} \) with probability \( 1 - a(x, \rho) \)).
- **Hyb_2**: Same as in Hyb_1, but the probability \( a(x, \rho) \) is changed to \( c(x, \rho, w) \).
- **Hyb_3**: Same as in Hyb_2, with the exception that with probability \( c(x, \rho, w) \), the process outputs a sample from \( A_{(p, V^*)} \) rather than from \( A_{\text{Sim}} \).
- **Hyb_4**: Same as in Hyb_3, with the exception that with probability \( 1 - c(x, \rho, w) \), the process outputs a sample from \( S_{(p, V^*)} \) rather than from \( S_{\text{Sim}} \). This process is exactly \( \text{OUT}_{V^*}(P(w), V^*(\rho))(x) \).

It is left to explain why each consecutive pair of distributions are computationally indistinguishable, and our proof is finished. For the following, define \( a'(\lambda) := a(x_\lambda, \rho_\lambda) \), \( b'(\lambda) := b(x_\lambda, \rho_\lambda) \), \( c'(\lambda) := c(x_\lambda, \rho_\lambda, w_\lambda) \).

- **Hyb_0 \approx_s Hyb_1**: Due to the fact that \( a'(\lambda) \) and \( b'(\lambda) \) are negligibly close (Corollary 3.1), it follows that \( a'(\lambda) \) and \( \frac{a(x, \rho)}{1 + a(x, \rho) - b(x, \rho)} \) are also negligibly close, and thus follows the statistical indistinguishability.
- **Hyb_1 \approx_s Hyb_2**: The probabilities \( a'(\lambda) \) and \( c'(\lambda) \) are negligibly close due to Corollary 3.1, and the statistical indistinguishability follows.
- **Hyb_2 \approx_c Hyb_3**: Assume toward contradiction that the indistinguishability does not hold, this means there is a distinguisher \( D^* \), an infinite subset \( Q \subseteq \mathbb{N} \) and a polynomial \( p : \mathbb{N} \to \mathbb{N} \), s.t. for all \( \lambda \in Q \), \( D^* \) distinguishes with advantage at least \( 1/p(\lambda) \) between \( \text{Hyb}_{2, \lambda} \) and \( \text{Hyb}_{3, \lambda} \). We consider two cases for the function \( c' \), and show that in both of them the contradiction follows from 3.2.

  - **Case 1**: for every polynomial function \( q : \mathbb{N} \to \mathbb{N} \), there are only finitely-many \( \lambda \in Q \) s.t. \( 1 - c'(\lambda) > 1/q(\lambda) \). This means that there is a negligible function \( \mu \) s.t. \( \forall \lambda \in \mathbb{N} : 1 - c'(\lambda) \leq \mu(\lambda) \). In that case, the contradiction follows directly from Proposition 3.2, because for indices \( \lambda \in Q \), a sample from \( \text{Sim}_a(x_\lambda, V^*_\lambda, \rho_\lambda) \) is statistically indistinguishable from a sample from \( \text{Hyb}_{2, \lambda} \), and a sample from \( \text{OUT}_{V^*}(P(w_\lambda), V^*_\lambda(\rho_\lambda))(x_\lambda) \) is statistically indistinguishable from a sample from \( \text{Hyb}_{3, \lambda} \).

  - **Case 2**: there is a polynomial function \( q' : \mathbb{N} \to \mathbb{N} \), s.t. there are infinitely-many \( \lambda \in Q \) s.t. \( 1 - c'(\lambda) > 1/q'(\lambda) \), we denote this infinite set of indices by \( Q' \). For these indices we can violate the indistinguishability from 3.2. More specifically, for \( \lambda \in Q' \) we can sample in polynomial time (say, \( q'(\lambda)^2 \)) and using polynomial-size quantum advice, from a distribution that is statistically indistinguishable from \( S_{\text{Sim}, \lambda} \), and reduce distinguishing between \( \text{Sim}_a(x_\lambda, V^*_\lambda, \rho_\lambda) \) and \( \text{OUT}_{V^*}(P(w_\lambda), V^*_\lambda(\rho_\lambda))(x_\lambda) \), to distinguishing between \( \text{Hyb}_{2, \lambda} \) and \( \text{Hyb}_{3, \lambda} \) in the following way.

When getting a sample from either \( \text{Sim}_a(x_\lambda, V^*_\lambda, \rho_\lambda) \) or \( \text{OUT}_{V^*}(P(w_\lambda), V^*_\lambda(\rho_\lambda))(x_\lambda) \), if the sample’s value was \( \text{Fail} \), approximately sample (as mentioned above, in time \( q(\lambda)^2 \) from
$S_{\text{Sim},\lambda}$. This can be done, for example, by using a polynomial amount of copies (i.e. $q'(\lambda)^2$) of the quantum advice $\rho$ of the verifier. This output of the reduction (whether it was $\text{Fail}$ that was swapped to a sample that is close to $S_{\text{Sim},\lambda}$, or whether it was a non-$\text{Fail}$ and was not swapped) is sent to the distinguisher $D^*$. Due to the fact that for the cases we got $\text{Fail}$, the generated sample is statistically indistinguishable from $S_{\text{Sim},\lambda}$, it follows that when we get a sample from $\text{Sim}_a(x, \rho)$ then the output sample of our reduction is statistically close to $\text{Hyb}_{3,\lambda}$, and when we get a sample from $\text{OUT}_{V^*_a}(P(w), \rho)$ then the output sample of our reduction is statistically close to $\text{Hyb}_{3,\lambda}$, and we get a contradiction.

- **Hyb$_3$ $\approx_c$ Hyb$_4$**: Assume toward contradiction that the indistinguishability does not hold, this means there is a distinguisher $D^*$, an infinite subset $Q \subseteq \mathbb{N}$ and a polynomial $p : \mathbb{N} \to \mathbb{N}$, s.t. for all $\lambda \in Q$, $D^*$ distinguishes with advantage at least $1/p(\lambda)$ between $\text{Hyb}_{3,\lambda}$ and $\text{Hyb}_{4,\lambda}$. We consider two cases for the function $c'$, and show that in both of them the contradiction follows from 3.3.

  - **Case 1**: for every polynomial function $q : \mathbb{N} \to \mathbb{N}$, there are only finitely-many $\lambda \in Q$ s.t. $c'(\lambda) > 1/q(\lambda)$. This means that there is a negligible function $\mu$ s.t. $\forall \lambda \in \mathbb{N} : c'(\lambda) \leq \mu(\lambda)$. In that case, the contradiction follows directly from Proposition 3.3, because for indices $\lambda \in Q$, a sample from $\text{Sim}_a(x, \rho)$ is statistically indistinguishable from a sample from $\text{Hyb}_{3,\lambda}$, and a sample from $\text{OUT}_{V^*_a}(P(w), \rho)(x)$ is statistically indistinguishable from a sample from $\text{Hyb}_{4,\lambda}$.

  - **Case 2**: there is a polynomial function $q' : \mathbb{N} \to \mathbb{N}$ s.t. $c'(\lambda) > 1/q'(\lambda)$, we denote this infinite set of indices by $Q'$. For these indices we can violate the indistinguishability from 3.3. More specifically, for $\lambda \in Q'$ we can sample, in polynomial time (say, $q'(\lambda)^2$) and using polynomial-size quantum advice, from a distribution that is statistically indistinguishable from $A_{\text{Sim},\lambda}$, and reduce distinguishing between $\text{Sim}_a(x, \rho)$ and $\text{OUT}_{V^*_a}(P(w), \rho)(x)$, to distinguishing between $\text{Hyb}_{3,\lambda}$ and $\text{Hyb}_{4,\lambda}$ in the following way.

  When getting a sample from either $\text{Sim}_a(x, \rho)$ or $\text{OUT}_{V^*_a}(P(w), \rho)(x)$, if the sample’s value was $\text{Fail}$, approximately sample (as mentioned above, in time $q(\lambda)^2$) from $A_{\text{Sim},\lambda}$. This can be done, for example, by using a polynomial amount of copies (i.e. $q'(\lambda)^2$) of the quantum advice $\rho$ of the verifier. This output of the reduction (whether it was $\text{Fail}$ that was swapped to a sample that is close to $A_{\text{Sim},\lambda}$, or whether it was a non-$\text{Fail}$ and was not swapped) is sent to the distinguisher $D^*$. Due to the fact that for the cases we got $\text{Fail}$, the generated sample is statistically indistinguishable from $A_{\text{Sim},\lambda}$, it follows that when we get a sample from $\text{Sim}_a(x, \rho)$ then the output sample of our reduction is statistically close to $\text{Hyb}_{3,\lambda}$, and when we get a sample from $\text{OUT}_{V^*_a}(P(w), \rho)(x)$ then the output sample of our reduction is statistically close to $\text{Hyb}_{4,\lambda}$, and we get a contradiction.

\[\square\]

We conclude with proving that the output of the simulation $\text{Sim}(x, \rho)$ is indeed computationally indistinguishable from the output of the real interaction $\text{OUT}_V(P, \rho)(x)$.

**Proposition 3.5** (Simulation Output is Indistinguishable from Interaction). For any quantum polynomial-size verifier $V^* = \{V^*_\lambda, \rho\}_{\lambda \in \mathbb{N}}$,

\[
\{\text{OUT}_{V^*}(P(w), V^*_\lambda(\rho))(x)\} \approx_c \{\text{Sim}(x, V^*_\lambda, \rho)\}, \quad \lambda \in \mathbb{N},
\]

when $x \in L \cap \{0, 1\}^\lambda$, $w \in R(x)$.

**Proof.** Let $V^* = \{V^*_\lambda\}_{\lambda \in \mathbb{N}}$ be a quantum polynomial-size verifier. According to Corollary 3.2, there is a negligible function $\text{negl}(\cdot)$ s.t. for every instance in the language $x = \{x^\lambda\}_{\lambda \in \mathbb{N}}$ and auxiliary input

30
quantum state $\rho = \{\rho_\lambda\}_{\lambda \in \mathbb{N}}$ for the verifier, we have

$$\forall \lambda \in \mathbb{N} : \Pr[\text{The simulation } \text{Sim}_{\text{comb}}(x_\lambda, V^*_\lambda, \rho_\lambda) \text{ succeeds}] - \frac{1}{2} \leq \text{negl}(\lambda) .$$

Consider the quantum circuit $\text{Sim}_{\text{comb},x,V^*}$, which is the circuit implementation of $\text{Sim}_{\text{comb}}$ with hard-wired inputs $x$ and $V^*$, that gets as input only the quantum state $\rho$. As mentioned above, the success probability of $\text{Sim}_{\text{comb},x,V^*}$ is negligibly close to $\frac{1}{2}$, for any quantum state $\rho$. If we denote the success probability for input $\rho$ by $p(\rho)$ and denote $\varepsilon := \text{negl}(\lambda) + 2^{-\lambda \frac{3}{4}}$, $p_0 := \frac{1}{2}$ and $q := \frac{1}{2}$, we can see that the 4 conditions for the Quantum Rewinding Lemma 2.1 are satisfied:

- $\lambda \geq \frac{\log(1/\varepsilon)}{4p_0(1-p_0)}$.
- For every state $\rho$, $p_0 \leq p(\rho)$.
- For every state $\rho$, $|p(\rho) - q| < \varepsilon$.
- $p_0(1 - p_0) \leq q(1 - q)$.

This implies that $\text{R}(\text{Sim}_{\text{comb},x,V^*}, \lambda)$ has trace distance bounded by $4\sqrt{\varepsilon} \frac{\log(1/\varepsilon)}{p_0(1-p_0)}$ from the success-conditioned output distribution of $\text{Sim}_{\text{comb}}(x, V^*, \rho)$. Since $\varepsilon$ is a negligible function of $\lambda$, so is $4\sqrt{\varepsilon} \frac{\log(1/\varepsilon)}{p_0(1-p_0)}$.

Finally, recall that $\text{Sim}(x, V^*, \rho) = \text{R}(\text{Sim}_{\text{comb},x,V^*}, \rho, \lambda)$, and that proposition 3.4 says that the success-conditioned distribution of $\text{Sim}_{\text{comb}}(x, V^*, \rho)$ is computationally indistinguishable from $\text{OUT}_{V^*}(\text{P}(w), V^*(\rho))(x)$, and our proof is concluded.  

\[ \square \]

**Remark 3.1 (Classical Universal Simulator for Classical Verifiers).** As a side note, we observe that the protocol preserves the trait of classical ZK, that is, classical verifiers learn nothing from the protocol (formally, for classical verifiers there is a classical simulator). A classical simulator showing this will simply execute $\text{Sim}_{\text{comb}}(x, V^*)$ repeatedly some polynomial number of times (either until it succeeds in one of the tries, or fails in all and then the output is $\text{Fail}$), specifically, $\lambda$ tries will do. Since the probability for $\text{Sim}_{\text{comb}}$ to succeed is $\approx \frac{1}{2}$, the probability to successfully sample from the success-conditioned distribution is overwhelming, and thus the output of the simulator is indistinguishable from the output of the verifier in the real interaction.

## 4 Quantumly-Extractable Classical Commitments

In this section we show how to use any constant-round post-quantum zero-knowledge argument for NP (and standard cryptographic assumptions) in order to construct a constant-round, quantumly-extractable classical commitment scheme. We start with the definition, and proceed to the construction.

**Definition 4.1 (Quantumly-Extractable Commitment).** A quantumly-extractable commitment scheme consists of three interactive PPT algorithms ($\text{Sen}$, $\text{Rec}$, $\text{VDcom}$) with the following syntax.

- $\text{Sen}(1^\lambda, m)$ : The sender algorithm gets as input the public security parameter $1^\lambda$ and the secret message $m$ to commit to.
- $\text{Rec}(1^\lambda)$ : The receiver algorithm gets only the public security parameter $1^\lambda$.
- The algorithms $\text{Sen}$, $\text{Rec}$ interact and generate transcript $T$.
- $\text{VDcom}(T, m, r)$ : For a transcript, message and randomness, the decommitment verification algorithm outputs a bit.
The scheme satisfies the following conditions.

- **Perfect Binding:** Let \( m_0, m_1, r_0, r_1 \in \{0, 1\}^* \), and let \( T \) be a transcript. If \( \text{VDcom}(T, m_0, r_0) = \text{VDcom}(T, m_1, r_1) = 1 \), then \( m_0 = m_1 \). Accordingly, for a transcript \( T \) denote by \( m_T \) the (unique) string such that if there exist \( r \) s.t. \( \text{VDcom}(T, m, r) = 1 \), then \( m_T := m \), and \( m_T := \perp \) otherwise.

- **Computational Hiding:** For every polynomial-size quantum receiver \( \text{Rec}^* = \{\text{Rec}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}} \) and polynomial \( \ell(\cdot) \),

\[
\{\text{OUT}_{\text{Rec}^*_\lambda}(\text{Sen}(m_0), \text{Rec}^*_\lambda(\rho_\lambda)(1^\lambda))\}_{\lambda, m_0, m_1} \approx_c \{\text{OUT}_{\text{Rec}^*_\lambda}(\text{Sen}(m_1), \text{Rec}^*_\lambda(\rho_\lambda)(1^\lambda))\}_{\lambda, m_0, m_1},
\]

where \( \lambda \in \mathbb{N}, m_0, m_1 \in \{0, 1\}^{\ell(\lambda)} \).

- **Extractability:** There exists a quantum polynomial-time algorithm \( \text{Ext} \) s.t. for every polynomial-size quantum sender \( \text{Sen}^* = \{\text{Sen}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}} \) outputs a quantum state \( \sigma_{\text{Ext}} \) and message \( m_{\text{Ext}} \), with the following guarantee.

\[
\{ (\sigma, m_T) \mid (T, \sigma, m_T) \leftarrow (\text{Sen}^*_\lambda(\rho_\lambda), \text{Rec})(1^\lambda) \}_{\lambda \in \mathbb{N}} \approx_c \{ (\sigma_{\text{Ext}}, m_{\text{Ext}}) \mid (\sigma_{\text{Ext}}, m_{\text{Ext}}) \leftarrow \text{Ext}(1^\lambda, \text{Sen}^*_\lambda, \rho_\lambda) \}_{\lambda \in \mathbb{N}},
\]

where \( \sigma \) is the inner state of \( \text{Sen}^* \) after executing the interaction with \( \text{Rec} \).

**Remark 4.1.** In the standard definition of extraction and more broadly, of simulation, the simulator does not output the interaction transcript (in classical-interaction protocols). It is noted however that it can be assumed without the loss of generality that the simulator also outputs the simulated transcript whenever needed. This is because, given a classical (or quantum) interactive circuit, it can be compiled in polynomial time (in the circuit size) to a circuit with identical functionality, that records the interaction transcript into its private inner state. Since the simulator simulates the inner state of the adversary at the end of interaction it in particular simulates the transcript.

We describe the protocol between \( \text{Sen} \) and \( \text{Rec} \) in Figure 2.

**Ingredients and notation:**

- A non-interactive commitment scheme \( \text{Com} \).
- A 2-message function-hiding secure function evaluation scheme \( (\text{SFE.Gen, SFE.Enc, SFE.Eval, SFE.Dec}) \).
- A constant-round post-quantum zero-knowledge argument system \( (\text{P}_{\text{NP}}, \text{V}_{\text{NP}}) \) for \( \text{NP} \).

**Decommitment Verification.** On input \( (T, m, r) \) the decommitment verification algorithm \( \text{VDcom} \) deduces the security parameter \( \lambda \) (the security parameter is public and can be assumed to be part of the transcript). It then checks two things:

- The argument that \( \text{Sen} \) gave at the last step of the transcript \( T \) is convincing (this is possible as the argument is publicly verifiable).
- The commitment \( \text{cmt}_{\text{Sen}} \) from step 1 in the transcript \( T \) indeed decommits to \( m, r \) (i.e. \( \text{Com}(1^\lambda, m; r) = \text{cmt}_{\text{Sen}} \)).
The output is 1 iff the check succeeds.

**Binding and Hiding.** The perfect binding property of the scheme follows readily from the perfect binding of the non-interactive commitment scheme Com. We next show hiding.

**Proposition 4.1** (The Commitment Scheme is Computationally Hiding). For every polynomial-size quantum receiver \( \text{Rec}^\star = \{\text{Rec}^\star_\lambda, \rho^\star_\lambda\}_{\lambda \in \mathbb{N}} \) and polynomial \( \ell(\cdot) \),

\[
\{\text{OUT}_{\text{Rec}^\star}(\langle \text{Sen}(m_0), \text{Rec}^\star_\lambda(\rho^\star_\lambda) \rangle(1^\lambda))\}_{\lambda,m_0,m_1} \approx_c \{\text{OUT}_{\text{Rec}^\star_\lambda}(\langle \text{Sen}(m_1), \text{Rec}^\star_\lambda(\rho^\star_\lambda) \rangle(1^\lambda))\}_{\lambda,m_0,m_1},
\]

where \( \lambda \in \mathbb{N}, m_0, m_1 \in \{0,1\}^{\ell(\lambda)} \).

**Proof.** We prove the claim by a hybrid argument. Define the following hybrid distributions on transcripts.

- \( \text{Hyb}_0 \): This is the output distribution \( \text{VIEW}_{\text{Rec}^\star}(\langle \text{Sen}(m_0), \text{Rec}^\star(\rho) \rangle) \).

- \( \text{Hyb}_1 \): The output distribution of a process that acts like \( \text{Hyb}_0 \), with the exception that in step 5, instead of \( \text{Sen} \) communicating with \( \text{Rec}^\star \) to give a ZK argument, we take the ZK simulator \( \text{Sim} \) of the argument system \( (P_{\text{NP}}, V_{\text{NP}}) \) and use it to simulate the argument by \( \text{Sen} \), by executing \( \text{Sim}(T', \text{Rec}^\star, \rho') \), where \( T' \) (resp. \( \rho' \)) is the transcript (resp. inner quantum state of \( \text{Rec}^\star \)) generated at the end of step 4 of the interaction.

- \( \text{Hyb}_2 \): The output distribution of a process that acts like \( \text{Hyb}_1 \), with the exception that in step 4b, instead of actually performing an SFE evaluation of \( C_{1 \rightarrow m_0} \), the process performs an SFE evaluation of the circuit \( C_{1 \rightarrow m_1} \) that always outputs \( \bot \).

- \( \text{Hyb}_3 \): The output distribution of a process that acts like \( \text{Hyb}_2 \), with the exception that in step 1, instead of committing to \( m_0 \), the sender commits to \( m_1 \).

- \( \text{Hyb}_4 \): The output distribution of a process that acts like \( \text{Hyb}_3 \), with the exception that in step 4b, the process performs an SFE evaluation of the circuit \( C_{1 \rightarrow m_1} \) and not of the circuit \( C_{1 \rightarrow m_0} \).

- \( \text{Hyb}_5 \): The output distribution of a process that acts like \( \text{Hyb}_4 \), with the exception that in step 5, instead of using the ZK simulator for the sender’s argument, the process uses the ZK argument regularly, that is, the sender proves that the transcript so far is consistent. Observe that this is exactly the output distribution \( \text{VIEW}_{\text{Rec}^\star}(\langle \text{Sen}(m_1), \text{Rec}^\star(\rho) \rangle) \).

We now explain why each consecutive pair of distributions are computationally indistinguishable, and our proof is finished.

- \( \text{Hyb}_0 \approx_c \text{Hyb}_1 \): Follows from the post-quantum zero-knowledge property of the protocol \( (P_{\text{NP}}, V_{\text{NP}}) \).

- \( \text{Hyb}_1 \approx_s \text{Hyb}_2 \): Assume toward contradiction that the two distributions are distinguishable, and fix, by an averaging argument, the partial transcript \( T' \) (and inner state of \( \text{Rec}^\star \)) that is generated at the end of step 2 of the protocol and maximizes the distinguishing advantage between the two distributions.

We consider two cases for the commitment \( \text{cmt}_{\text{Rec}} \) in the transcript \( T' \): The simpler case is if \( \text{cmt}_{\text{Rec}} \) is not a commitment to 0 (i.e. there is no \( r_0 \in \{0,1\}^s \) s.t. \( \text{cmt}_{\text{Rec}} = \text{Com}(1^\lambda, 0; r_0) \)), in that case, by the soundness of the argument that \( \text{Rec}^\star \) gives in step 3, with overwhelming probability \( \text{Sen} \) is going to reject the proof and end communication, and only with a negligible probability the process continues to a point where the two processes \( \text{Hyb}_1, \text{Hyb}_2 \) differ, in contradiction.

In the second case \( \text{cmt}_{\text{Rec}} \) is a valid commitment to 0. In that case, the contradiction follows from (an implication of) the circuit privacy property of the SFE encryption, specifically, it follows from
Claim 2.2. From the perfect binding of the non-interactive commitment scheme Com, there is no string \( r_1 \) s.t. \( \text{cmt}_\text{Rec} = \text{Com}(1^\lambda, 1; r_1) \), which in turn implies that the circuit \( C_{1 \rightarrow m_0} \) is identical in functionality to the circuit \( C_{\perp} \) that outputs \( \perp \) on any input. By Claim 2.2 it follows that the responses from \( \text{Sen} \) in step 4b are statistically indistinguishable, and thus also the distributions \( \text{Hyb}_1 \) and \( \text{Hyb}_2 \), again in contradiction.

- \( \text{Hyb}_2 \approx_c \text{Hyb}_3 \): Follows from the hiding of the commitment \( \text{cmt}_\text{Sen} \) that \( \text{Sen} \) gives in step 1, that is, the hiding property of the commitment scheme \( \text{Com} \).

- \( \text{Hyb}_3 \approx_s \text{Hyb}_4 \): This indistinguishability follows from the exact same reasoning as in the explanation for the indistinguishability \( \text{Hyb}_1 \approx_s \text{Hyb}_2 \), by swapping \( m_0 \) with \( m_1 \) in the explanation.

- \( \text{Hyb}_4 \approx_c \text{Hyb}_5 \): Like the indistinguishability \( \text{Hyb}_0 \approx_c \text{Hyb}_1 \), this indistinguishability follows again from the zero-knowledge property of the argument system \((P_{NP}, V_{NP})\).

4.1 Extractability

We show a quantum polynomial-time extractor \( \text{Ext} \) s.t. for any polynomial-size quantum sender \( \text{Sen}^* = \{\text{Sen}_{\lambda}, \rho_{\lambda}\}_{\lambda \in \mathbb{N}} \) extracts the sender’s committed message (if it exists) and also simulates its quantum state at the end of the protocol.

\[
\text{Ext}(1^\lambda, \text{Sen}^*, \rho) : \\
1. \text{Simulation of Commitments:} \quad \text{Sen}^* \text{ outputs } \text{cmt}_{\text{Sen}}. \quad \text{Ext then sends to } \text{Sen}^* \text{ a commitment to } 1: \quad \text{cmt}_{\text{Ext}} = \text{Com}(1^\lambda, 1; r_1), \text{ where } r_1 \in \{0, 1\}^{\text{poly}(\lambda, 1)} \text{ is the random string used as the randomness of the commitment algorithm.}

2. \text{Simulation of ZK Argument by Rec:} \quad \text{Ext uses the zero-knowledge simulator } \text{Sim} \text{ of the argument system } (P_{NP}, V_{NP}). \quad \text{Ext executes } \text{Sim}(\text{cmt}_{\text{Ext}}, \text{Sen}^*, \rho^{(1)}) \text{ to simulate the argument that Rec gives to } \text{Sen}^* \text{ at step 3 of the protocol (} \rho^{(1)} \text{ is the inner state of } \text{Sen}^* \text{ after step 1 of the extraction). At the end of the zero-knowledge simulation, we have a simulated argument transcript and a quantum state } \rho' \text{ for } \text{Sen}^* \text{ to carry on to the next step of extraction.}

3. \text{Extraction of Message from } \text{Sen}^*:

- \( \text{Ext computes } d_k \leftarrow \text{SFE.Gen}(1^\lambda) \text{ and sends } \text{ct}_{\text{Ext}} \leftarrow \text{SFE.Enc}_{d_k}(r_1). \)
- \( \text{Sen}^* \text{ outputs a response } \text{ct}. \)

\( \text{Ext then decrypts the evaluated ciphertext to get a message } m'. \)

4. \text{ZK Argument by } \text{Sen}^*: \quad \text{Ext takes the role of the honest receiver Rec in the ZK argument } \text{Sen}^* \text{ gives.}

5. \text{Extraction Procedure Output:} \quad \text{The output } (\sigma_{\text{Ext}}, m_{\text{Ext}}) \text{ of the extraction procedure is as follows.}

- \( \text{The simulated inner state } \sigma_{\text{Ext}} \text{ for the sender is set to be the inner state of } \text{Sen}^* \text{ at the time of halting of the procedure.} \)
- \( \text{If the argument that } \text{Sen}^* \text{ gave in step 4 of the procedure is convincing then } m_{\text{Ext}} = m', \text{ otherwise } m_{\text{Ext}} = \perp. \)
It remains to explain why the extraction process yields an output that is computationally indistinguishable from a tuple \((T, \sigma, m_T)\) generated by the real interaction between \(\text{Sen}^*(\rho)\) and \(\text{Rec}\), and also that the extracted message \(m_{\text{Ext}}\) is indeed the message that \(T_{\text{Ext}}\) can be decommitted to.

**Proposition 4.2.** Let \(\text{Sen}^* = \{\text{Sen}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}\) be a polynomial-size quantum sender, then,
\[
\{ (\sigma, m_T) | (T, \sigma, m_T) \leftarrow \langle \text{Sen}^*_\lambda(\rho_\lambda), \text{Rec} \rangle (1^\lambda) \}_{\lambda \in \mathbb{N}} \approx_c \{ \text{Ext}(1^\lambda, \text{Sen}^*_\lambda, \rho_\lambda) \}_{\lambda \in \mathbb{N}}.
\]

**Proof.** We prove the claim by a hybrid argument. Define the following hybrid processes:

- **Hyb_0:** This distribution is the output distribution \((\sigma, m_T)\) of the real interaction \(\langle \text{Sen}^*(\rho), \text{Rec} \rangle\).

- **Hyb_1:** The output distribution of a process that acts like \(\text{Hyb}_0\), with the exception that in step 3, instead of \(\text{Rec}\) communicating with \(\text{Sen}^*\) to give a ZK argument, we take the ZK simulator \(\text{Sim}\) of the argument system \((\text{P}_{\text{NP}}, V_{\text{NP}})\) and use it to simulate the argument by \(\text{Rec}\) by executing \(\text{Sim}(T', \text{Sen}^*, \rho')\), where \(T'\) (resp. \(\rho'\)) is the transcript (resp. inner quantum state of \(\text{Rec}^*\)) generated at the end of step 2 of the interaction.

- **Hyb_2:** The output distribution of a process that acts like \(\text{Hyb}_1\), with the exception that when \(\text{Rec}\) sends \(\text{cmt}_{\text{Rec}}\), it commits to 1 instead of 0.

- **Hyb_3:** The output distribution of a process that acts like \(\text{Hyb}_2\), with the exception that in step 4a, \(\text{Rec}\) sends an SFE encryption \(\text{cmt}_{\text{Rec}}\) of the randomness \(r_1\) that it used in step 2 when it committed for 1. Note that this output distribution is identical to the extraction’s output \(\text{Ext}(1^\lambda, \text{Sen}^*, \rho)\), with the only change being that \(m_T\) is generated as in \(\langle \text{Sen}^*(\rho), \text{Rec} \rangle\).

- **Hyb_4:** The output distribution of a process that acts like \(\text{Hyb}_3\), with the exception that the output message \(m_T\) is generated differently, specifically, \(m_T = m_{\text{Ext}}\) that is generated as in step 5 of the extraction procedure. Note that this process is exactly the output distribution \((\sigma_{\text{Ext}}, m_{\text{Ext}}) \leftarrow \text{Ext}(1^\lambda, \text{Sen}^*, \rho)\).

We now explain why each pair of consecutive distributions are computationally indistinguishable, and our proof is finished.

- **Hyb_0 \approx_c Hyb_1:** Assume toward contradiction that the two distributions are distinguishable, and fix, by an averaging argument, the partial transcript \(T'\) and inner quantum state \(\sigma'\) of \(\text{Sen}^*\) generated at the end of step 1 of the simulation, that maximizes the distinguishing advantage of the two distributions. Inside such transcript \(T'\) we consider the sender commitment \(\text{cmt}_{\text{Sen}}\), and the (unique, by the perfect binding of the commitment scheme \(\text{Com}\)) message \(m_{T'}\) that is inside this commitment (if the commitment cannot be opened to any message, \(m_{T'} := \perp\)).

From our assumption that \(\text{Hyb}_0, \text{Hyb}_1\) are distinguishable, follows the existence of a distinguisher that breaks the zero-knowledge property of \((\text{P}_{\text{NP}}, V_{\text{NP}})\). Specifically, the distinguisher uses as non-uniform advice the partial transcript \(T'\) and the message \(m_{T'}\), gets either a real interaction transcript or a simulation of the argument that \(\text{Rec}\) gives in step 3 of the protocol, then executes the rest of the commitment protocol, and uses the knowledge \(m_{T'}\) at the end of protocol execution to output \(m_T\). It follows that such distinguisher breaks the zero knowledge property of \((\text{P}_{\text{NP}}, V_{\text{NP}})\), in contradiction.

In the following explanations for the indistinguishabilities we will use the same averaging argument and non-uniform advice that includes the message \(m_{T'}\), and refer to it simply as the "averaging argument with non-uniform advice message".
• \( \text{Hyb}_1 \approx_c \text{Hyb}_2 \): Follows from the same averaging argument and non-uniform advice message reasoning from the proof of \( \text{Hyb}_0 \approx_c \text{Hyb}_1 \), along with the hiding of the commitment scheme Com.

• \( \text{Hyb}_2 \approx_c \text{Hyb}_3 \): Follows from the same averaging argument and non-uniform advice message reasoning from the proof of \( \text{Hyb}_0 \approx_c \text{Hyb}_1 \), along with the input privacy (encryption security) property of the SFE encryption.

• \( \text{Hyb}_3 \approx_s \text{Hyb}_4 \): Recall that both processes \( \text{Hyb}_3, \text{Hyb}_4 \) generate the output state \( \sigma_{\text{Ext}} \) as in the extraction procedure, but differ only in the way they generate the output message. Assume toward contradiction that \( \text{Hyb}_3, \text{Hyb}_4 \) are distinguishable and fix, by an averaging argument, the partial transcript \( T' \) (and inner state \( \sigma' \) of \( \Sigma^* \)) generated at the end of step 3 of the extraction.

Consider two cases for the partial transcript \( T' \).

– \( T' \) is consistent. In that case, by the (perfect) correctness of the SFE evaluation it follows that the generated messages \( m_T \) (from \( \text{Hyb}_3 \)) and \( m_{\text{Ext}} \) (from \( \text{Hyb}_4 \)) are identical. The rest of the protocol execution, which includes only the argument by \( \Sigma^* \), is also identical between the two distributions. The distinguisher between \( \text{Hyb}_3, \text{Hyb}_4 \) (we assumed toward contradiction exists) cannot distinguish between these two distributions as they are identical, in contradiction.

– \( T' \) is not consistent. In that case, by the soundness of the argument system \( (P_{\text{NP}}, V_{\text{NP}}) \), the argument by \( \Sigma^* \) fails with overwhelming probability, and with the same probability the values of both \( m_T \) (from \( \text{Hyb}_3 \)) and \( m_{\text{Ext}} \) (from \( \text{Hyb}_4 \)) are \( \perp \). It follows that the statistical distance between the distributions is negligible, in contradiction.

\[ \square \]

5 Constant-Round Zero-Knowledge Quantum Arguments for QMA

In this section we explain how the tools from previous sections imply a constant-round zero-knowledge quantum argument for QMA, that is, according to Definition 2.7 where honest parties are polynomial-time and quantum (prover is efficient given a quantum witness) and communication is quantum.

The construction uses constant-round (post-quantum) zero-knowledge arguments for NP, quantumly-extractable commitments and a quantum sigma protocol for QMA. We now proceed to the construction and proof.

**Ingredients and notation:**

- A constant-round quantumly-extractable commitment scheme \((\text{Sen}, \text{Rec})\).

- A constant-round post-quantum zero-knowledge argument system \((P_{\text{NP}}, V_{\text{NP}})\) for NP.

- A quantum sigma protocol for QMA \((\Xi, P, \Xi, V)\).

We describe the protocol in Figure 3.

\[ ^{\text{In a previous version of this work we used the QMA zero-knowledge (with large soundness error) protocol of [BJSW16] instead of sigma protocols. Using sigma protocols yields a simplified protocol.}} \]
5.1 Computational Soundness

We prove that Protocol Protocol 3 has quantum computational soundness.

**Proposition 5.1.** For any quantum polynomial-size prover \( P^* = \{P^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}} \), there exists a negligible function \( \mu(\cdot) \) such that for any security parameter \( \lambda \in \mathbb{N} \) and any \( x \in \{0,1\}^\lambda \setminus \mathcal{L} \),

\[
\Pr \left[ \text{OUT}_V(P^*_\lambda, V)(x) = 1 \right] \leq \mu(\lambda).
\]

**Proof.** Let \( P^* = \{P^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}} \) a polynomial-size quantum prover and let \( x = \{x_\lambda\}_{\lambda \in \mathbb{N}} \) be a sequence such that \( \forall \lambda \in \mathbb{N} : x_\lambda \in \{0,1\}^\lambda \setminus \mathcal{L} \). We prove soundness by a hybrid argument. We consider a series of hybrid processes with output over \{0,1\}, starting from \( \text{OUT}_V(P^*(\rho), V)(x) \) the output distribution of \( V \) in the interaction with \( P^* \).

- \( \text{Hyb}_0 \) : The output distribution of \( \text{OUT}_V(P^*_\lambda, V)(x_\lambda) \).

- \( \text{Hyb}_1 \) : Identical to the process \( \text{Hyb}_0 \), with the exception that in step 3b when the verifier gives a ZK argument, the process instead uses the ZK simulator Sim of the argument system \( (P_{\mathbf{NP}}, V_{\mathbf{NP}}) \).

To simulate the prover’s view, the process executes \( \text{Sim}((T_{\text{Sen}}, \beta), \rho^*, \rho') \), where \( \rho' \) is the inner quantum state of \( P^* \) at the end of step 3a where the verifier sends \( \beta \).

- \( \text{Hyb}_2 \) : Identical to the process \( \text{Hyb}_1 \), with the exception that in step 1 when the verifier commits to \( \beta \), the process instead commits to \( 0|\beta| \).

We next explain why each consecutive pair of distributions are indistinguishable.

- \( \text{Hyb}_0 \approx_c \text{Hyb}_1 \) : Follows from the quantum zero knowledge property of the protocol \( (P_{\mathbf{NP}}, V_{\mathbf{NP}}) \).

- \( \text{Hyb}_1 \approx_c \text{Hyb}_2 \) : Follows from the computational hiding of the commitment scheme \( (\text{Sen}, \text{Rec}) \).

Now, assume toward contradiction that \( P^* \) succeeds in making the verifier accept with some noticeable probability \( \varepsilon(\lambda) \), that is, the probability for the output 1 in \( \text{Hyb}_0 \) is noticeable. \( \text{Hyb}_0 \approx_c \text{Hyb}_2 \), and thus the probability for the output 1 in \( \text{Hyb}_2 \) is also noticeable. Finally, we get a contradiction to the soundness of the sigma protocol \((\Xi, P, \Xi, V)\), by using the prover sigma protocol messages from steps 2, 4 as messages to convince a quantum sigma protocol verifier \( \Xi, V \). Since the probability that the verifier \( V \) is convinced in \( \text{Hyb}_2 \) is noticeable, and such verifier is convinced if and only if the sigma protocol verifier is convinced, we get our contradiction.

\( \square \)

5.2 Computational Zero Knowledge

We prove that Protocol Protocol 3 is quantum computational zero knowledge.

We describe a universal simulator Sim for the protocol. We denote by \( V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}} \) a polynomial-size quantum verifier. The simulator takes as input an instance in the language \( x \in \{0,1\}^\lambda \cap \mathcal{L} \), a verifier circuit \( V^*_\lambda \) and quantum auxiliary input \( \rho_\lambda \) for \( V^*_\lambda \). Subscripts are dropped when are clear from the context.

\text{Sim}(x, V^*, \rho):

1. **Extraction of Message from Verifier:** Sim executes the extractor \( \text{Ext} \) of the extractable commitment scheme \((\text{Sen}, \text{Rec})\). Sim computes a simulation of the commitment interaction transcript, inner state at the end of interaction and extracted message \((T_{\text{Ext}}, \sigma_{\text{Ext}}, \beta_{\text{Ext}}) \leftarrow \text{Ext}(1^\lambda, V^*, \rho)\) and uses the simulated state \( \sigma_{\text{Ext}} \) as inner state for \( V^* \) in order to continue the protocol simulation\(^6\).

---

\(^6\)By the standard definition, the extractor \( \text{Ext} \) simulates only the state and extracted message \((\sigma_{\text{Ext}}, m_{\text{Ext}})\), but recall we can assume without the loss of generality that it also simulates the commitment transcript \( T_{\text{Ext}} \) (see Remark 4.1), and the triplet is indistinguishable from \((T, \sigma, m_T) \leftarrow \langle \text{Sen}^*(\rho), \text{Rec} \rangle(1^\lambda)\).
Proposition 5.2. For any polynomial-size quantum verifier \( V^* = \{V^*_\lambda, \rho^*_\lambda\}_{\lambda \in \mathbb{N}} \),

\[
\{ \text{OUT}_{V^*_\lambda} (P(\omega^{\otimes k(\lambda)}), V^*_\lambda(\rho^*_\lambda))(x) \}_{\lambda : x, w} \approx_c \{ \text{Sim}(x, V^*_\lambda, \rho^*_\lambda) \}_{\lambda : x, w},
\]

where \( \lambda \in \mathbb{N} \), \( x \in \mathcal{L} \cap \{0, 1\}^\lambda \), \( w \in \mathcal{R}_\mathcal{L}(x) \).

Proof. We prove the claim by a hybrid argument, specifically, we consider hybrid distributions, all of which will be computationally indistinguishable.

- **Hyb\(_0\)**: The output distribution of the simulator \( \text{Sim}(x, V^*, \rho) \).
- **Hyb\(_1\)**: Identical to the process Hyb\(_0\), except that we erase some extreme cases from the output distribution, by making a check. Specifically, in step 3 when the verifier sends \( \beta \) and a ZK argument, if \( \beta_{\text{Ext}} \neq \beta \) and also the argument by \( V^* \) was convincing, the output of the process is \( \bot \).
- **Hyb\(_2\)**: Identical to the process Hyb\(_1\), except that in steps 2, 4 where the simulator sends \( \alpha_{\text{Sim}} \) and \( \gamma_{\text{Sim}} \), the process instead uses the real sigma protocol prover to generate the messages, \((\alpha, \tau) \leftarrow \Xi. P_1(x, \omega^{\otimes k(\lambda)}), \gamma \leftarrow \Xi. P_3(\beta_{\text{Ext}}, \tau)\).
- **Hyb\(_3\)**: Identical to the process Hyb\(_2\), except that when computing the last sigma protocol message \( \gamma \leftarrow \Xi. P_3(\beta_{\text{Ext}}, \tau) \), the process uses the \( \beta \) that \( V^* \) sent instead of the extracted \( \beta_{\text{Ext}} \), that is, \( \gamma \leftarrow \Xi. P_3(\beta, \tau) \).
- **Hyb\(_4\)**: Identical to the process Hyb\(_3\), except that the check described in Hyb\(_1\) is not performed, that is, even if the extracted challenge \( \beta_{\text{Ext}} \) and the challenge \( \beta \) sent by \( V^* \) are distinct and the ZK argument by \( V^* \) succeeds, the process carries on to the last step 4 and does not outputs \( \bot \).
- **Hyb\(_5\)**: At this point in our series of hybrid distributions we do not use the extracted challenge \( \beta_{\text{Ext}} \), and we would like to move to a final process that does not use extraction at all. This process is identical to Hyb\(_4\), with the exception that in step 1 of the simulation, where the simulator executes Ext to simulate the transcript and inner state of \( V^* \), the process simply executes the real interaction between \( V^* \) and \( \text{Rec} \), \((T, \sigma) \leftarrow \langle V^*(\rho), \text{Rec} \rangle(1^\lambda) \). Observe that Hyb\(_5\) is exactly the real interaction output \( \text{OUT}_{V^*} (P(\omega^{\otimes k}), V^*(\rho))(x) \).

Before proving that each consecutive pair of hybrids is indistinguishable,

We prove why each consecutive pair of distributions are computationally indistinguishable, and our proof is finished.

- **Hyb\(_0 \approx_c\) Hyb\(_1\)**: To show the indistinguishability we need to prove that the probabilistic event that exists in Hyb\(_0\) but is erased in Hyb\(_1\) happens with a negligible probability. This is exactly the statement proven in Claim 5.1.
• $\text{Hyb}_1 \approx \text{Hyb}_2$: This indistinguishability follows from the special zero knowledge property of the quantum sigma protocol.

• $\text{Hyb}_2 \equiv \text{Hyb}_3$: Due to the fact that in both hybrid processes, whenever $\beta_{\text{Ext}} \neq \beta$ the process halts and outputs ⊥, it is always the case that the first prover sigma protocol message $\gamma$ is computed with respect to the sent $\beta$.

• $\text{Hyb}_3 \approx_s \text{Hyb}_4$: The reasoning for this indistinguishability is identical to the reasoning for the indistinguishability $\text{Hyb}_0 \approx_s \text{Hyb}_1$, and follows from Claim 5.1.

• $\text{Hyb}_4 \approx \text{Hyb}_5$: This indistinguishability follows from the extractability property (in Definition 4.1) of the commitment scheme $(\text{Sen}, \text{Rec})$.

\begin{proof}
Let $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ a polynomial-size quantum verifier. Consider the process of interaction between $V^*(\rho)$ and $P$ in the original protocol, with one change: when $V^*$ gives an extractable commitment, instead of executing the interaction $(T, \sigma) \leftarrow \langle V^*(\rho), \text{Rec} \rangle(1^\lambda)$, the process executes the extractor $(T_{\text{Ext}}, \sigma_{\text{Ext}}, \beta_{\text{Ext}}) \leftarrow \text{Ext}(1^\lambda, V^*, \rho)$. Then, there is some negligible function $\text{negl}$ such that,

$$\Pr[(\beta \neq \beta_{\text{Ext}}) \land (V^* \text{ gives a convincing argument})] \leq \text{negl}(\lambda).$$

\end{proof}

Claim 5.1 (Extracted Information is Correct Under an Argument). Let $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ a polynomial-size quantum verifier. Consider the process of interaction between $V^*(\rho)$ and $P$ in the original protocol, with one change: when $V^*$ gives an extractable commitment, instead of executing the interaction $(T, \sigma) \leftarrow \langle V^*(\rho), \text{Rec} \rangle(1^\lambda)$, the process executes the extractor $(T_{\text{Ext}}, \sigma_{\text{Ext}}, \beta_{\text{Ext}}) \leftarrow \text{Ext}(1^\lambda, V^*, \rho)$. Then, there is some negligible function $\text{negl}$ such that,

$$\Pr[(\beta \neq \beta_{\text{Ext}}) \land (V^* \text{ gives a convincing argument})] \leq \text{negl}(\lambda).$$

Proof. Let $T_{\text{Sen}}$ be the transcript generated at the end of the extractable commitment protocol, in the original interaction between $V^*$ and $P$. By the perfect binding of the commitment scheme $(\text{Sen}, \text{Rec})$ it follows that if the statement from $V^*$’s ZK argument is correct, that is, there is some $r \in \{0, 1\}$ s.t. $1 = VD\text{com}(T_{\text{Sen}}, \beta, r)$, then $\beta$ is necessarily the committed message, in symbols (denoted in the binding property in Definition 4.1) $\beta = m_{T_{\text{Sen}}}$. It follows from the soundness of the argument that $V^*$ gives, that only with a negligible probability $\text{negl}'(\lambda)$ it happens that both, $\beta \neq m_{T_{\text{Sen}}}$, and $V^*$ gives a convincing argument.

Recall that by the extractability property of the commitment scheme (Extractability property in Definition 4.1), the following two distributions are indistinguishable,

$$\{(T, \sigma, m_T) \mid (T, \sigma, m_T) \leftarrow \langle \text{Sen}^*_\lambda(\rho_\lambda), \text{Rec} \rangle(1^\lambda)\}_{\lambda \in \mathbb{N}} \approx_e \{(T_{\text{Ext}}, \sigma_{\text{Ext}}, m_{\text{Ext}}) \mid (\sigma_{\text{Ext}}, m_{\text{Ext}}) \leftarrow \text{Ext}(1^\lambda, \text{Sen}^*_\lambda, \rho_\lambda)\}_{\lambda \in \mathbb{N}}.$$

This means that when considering the process described in this claim’s statement, where extraction takes place (instead of executing the commitment procedure), only with some negligible probability $\text{negl}(\lambda)$ it can happen that both, $\beta \neq \beta_{\text{Ext}}$, and $V^*$ gives a convincing argument, this is because if this probability wasn’t negligible we would be able to break the extractability property of the commitment scheme $(\text{Sen}, \text{Rec})$.

Acknowledgments

We thank Zvika Brakerski for insightful discussions about Quantum Fully-Homomorphic Encryption. We also thank Venkata Koppula for advice regarding the state of the art of Compute-and-Compare Obfuscation.
References

[AP] Prabhanjan Ananth and Rolando La Placa. Personal communication.


Protocol 1

**Common Input:** An instance $x \in L \cap \{0, 1\}^\lambda$, for security parameter $\lambda \in \mathbb{N}$.

**P’s private input:** A classical witness $w \in \mathcal{R}_L(x)$ for $x$.

1. **Prover Commitment:** $P$ sends non-interactive commitments to the witness $w$ and to a string of zeros in the length of an SFE secret key $dk$: $\text{cmt}_1 \leftarrow \text{Com}(\lambda, w)$, $\text{cmt}_2 \leftarrow \text{Com}(\lambda, 0^{dk})$.

2. **Extractable Commitment to Verifier Challenge:**
   
   (a) $V$ computes a challenge $\beta \leftarrow \Sigma.V$.
   
   (b) $V$ computes $s \leftarrow \{0, 1\}^\lambda$, $t \leftarrow \{0, 1\}^\lambda$, $(pk, sk) = \text{QHE.Keygen}(\lambda; r)$ where $r$ is the sampled randomness for the QFHE key generation algorithm. $V$ sends $pk$, $ct_V \leftarrow \text{QHE.Enc}_{pk}(t)$, $C C \leftarrow \text{Obf}\left(CC[\text{QHE.Dec}_{sk}(-), s, (r, \beta)]\right)$.
   
   (c) $P$ computes $dk \leftarrow \text{SFE.Gen}(\lambda)$ and sends $ct_P \leftarrow \text{SFE.Enc}_{dk}(0^\lambda)$.
   
   (d) $V$ sends $\hat{ct} \leftarrow \text{SFE.Eval}\left(CC[\text{Id}(-), t, s], ct_P\right)$, where $\text{Id}(-)$ is the identity function.

3. **Sigma Protocol Execution:**
   
   (a) $P$ computes $(\alpha, \tau) \leftarrow \Sigma.P_1(x, w)$ and sends $\alpha$.
   
   (b) $V$ sends the challenge $\beta$.

4. **WI Proof by the Verifier:** $V$ gives a WI proof of the following statement:
   
   - The transcript of the verifier so far is explainable.
   - Or, $\text{cmt}_1$ is a commitment to a non-witness $u \notin \mathcal{R}_L(x)$.

   The witness that $V$ uses for the proof is its randomness, that proves that the transcript is explainable.

5. **WI Proof by the Prover:** $P$ gives a WI proof of the following statement:
   
   - $x \in L$.
   - Or, $\text{cmt}_1$, $\text{cmt}_2$ are both valid commitments and furthermore, $ct_P$ is a valid SFE encryption and is encrypted with a key $dk$ which is the content of the commitment $\text{cmt}_2$.

   The witness that $P$ uses for the proof is $w$, that proves $x \in L$.

6. **Sigma Protocol Completion:** $P$ sends $\gamma = \Sigma.P_3(\beta, \tau)$.

7. **Acceptance:** $V$ accepts if $\Sigma.V(\alpha, \beta, \gamma) = 1$.

8. **Reactions to Aborts:** During the protocol, if either party sends a message of an incorrect form or provides a non-convincing WI proof, the other party terminates the interaction.

Figure 1: A classical constant-round zero-knowledge argument for $L \in \text{NP}$ with quantum security.
Protocol 2

**Common Input:** A security parameter \( \lambda \in \mathbb{N} \).

**Private Input of Sen:** A message \( m \in \{0, 1\}^* \) to commit to.

1. **Commitment by Sen:** Sen sends a commitment to \( m \),
   \[ \text{cmt}_{\text{Sen}} \leftarrow \text{Com}(1^\lambda, m) \].

2. **Commitment by Rec:** Rec sends a commitment to 0,
   \[ \text{cmt}_{\text{Rec}} \leftarrow \text{Com}(1^\lambda, 0) \].

3. **ZK Argument by Rec:** Rec interacts with Sen through \( (P_{\text{NP}}, V_{\text{NP}}) \) to give a ZK argument that \( \text{cmt}_{\text{Rec}} \) is indeed a commitment to 0, that is, there exists randomness \( r_0 \in \{0, 1\}^{\text{poly}(\lambda, 1)} \),
   a string\(^a\) s.t.
   \[ \text{cmt}_{\text{Rec}} = \text{Com}(1^\lambda, 0; r_0) \].

4. **Sen Challenges Rec:** The parties interact so that Sen can offer to send \( m \) if Rec managed to trick Sen in the ZK argument.
   (a) Rec computes \( \text{dk} \leftarrow \text{SFE.Gen}(1^\lambda) \) and sends \( \text{ct}_{\text{Rec}} \leftarrow \text{SFE.Enc}_{\text{dk}}(0^{\text{poly}(\lambda, 1)}) \).
   (b) Sen sends \( \hat{\text{ct}} \leftarrow \text{SFE.Eval}\left(\text{C}_{1 \rightarrow m}, \text{ct}_{\text{Rec}}\right) \), where \( \text{C}_{1 \rightarrow m} \) is the (canonical) circuit that for input \( r_1 \in \{0, 1\}^{\text{poly}(\lambda, 1)} \), s.t.
   \[ \text{cmt}_{\text{Rec}} = \text{Com}(1^\lambda, 1; r_1) \], outputs \( m \), and for any other input outputs \( \bot \).

5. **ZK Argument by Sen:** Sen interacts with Rec through \( (P_{\text{NP}}, V_{\text{NP}}) \) to give a ZK argument for the statement that its transcript until the end of step 4b is consistent, that is, there exists a message and randomness for the honest sender algorithm Sen that generates the transcript.

\[^a\text{Let poly}(\lambda, \ell)\text{ denote the polynomial that represents the amount of randomness the commitment algorithm Com(·) needs for security parameter } \lambda \text{ and message length } \ell.\]

---

Figure 2: A Quantumly-Extractable Classical Commitment Scheme.
**Protocol 3**

**Common Input:** An instance $x \in \mathcal{L} \cap \{0, 1\}^\lambda$, for security parameter $\lambda \in \mathbb{N}$.

**P’s private input:** Polynomially many identical witnesses for $x$: $w^{\otimes k(\lambda)}$ s.t. $w \in \mathcal{R}_x$.

1. **Verifier Extractable Commitment to Challenge:** $V$ computes $\beta \leftarrow \Xi$, $V$ and commits to it using the extractable commitment $(\text{Sen}, \text{Rec})$. $V$ executes $\text{Sen}(1^\lambda, \beta)$ and $P$ executes $\text{Rec}(1^\lambda)$, and commitment transcript $T_{\text{sen}}$ is generated.

2. **Prover Commitment:** $P$ computes $(\alpha, \tau) \leftarrow \Xi, P_1(x, w^{\otimes k(\lambda)})$ and sends $\alpha$ to $V$.

3. **Verifier Challenge and ZK Argument:**
   (a) $V$ sends $\beta$.
   (b) $V$ proves in ZK (using the argument system $(P_{NP}, V_{NP})$) that the sent $\beta$ is the value inside the extractable commitment, that is, $\exists r \in \{0, 1\}^*$ such that $1 = V\text{Dcom}(T_{\text{sen}}, \beta, r)$. If the argument was not convincing $P$ terminates communication.

4. **Sigma Protocol Completion:** If the proof by $V$ was convincing then $P$ computes $\gamma \leftarrow \Xi, P_3(\beta, \tau)$ and sends $\gamma$.

5. **Acceptance:** The verifier accepts iff $1 = \Xi, V(\alpha, \beta, \gamma)$.

Figure 3: A quantum constant-round zero-knowledge argument for $\mathcal{L} \in \text{QMA}$. 

46