Post-Quantum Zero Knowledge in Constant Rounds[*]

Nir Bitansky[†]   Omri Shmueli[‡]

November 17, 2019

Abstract

We construct the first constant-round zero-knowledge classical argument for **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 the first constant-round zero-knowledge quantum argument for **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.

Contents

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' techniques do 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 arbitrary polynomial-size non-uniform quantum circuits.

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.[1]

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 $\mathcal{L} \in$ **NP**.*

[1] We note that cryptographic hardness assumptions against algorithms with quantum advice are considered quite often, QLWE included (see for instance [Wat09, BJSW16, BCM$^+$18]).

Based on our techniques, we also construct (under the same assumptions) a fully-simulatable coin-flipping protocol, which combined with the work of Broadbent et al. [BJSW16], 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 $\mathcal{L} \in$ **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 statistically 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 Post-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 α to a set of bits, (2) a verifier challenge β, and (3) a prover response γ, in which it opens the commitments corresponding to the challenge β. 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 describe 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 β 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 α. 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*, which roughly shows that *non-rewinding* simulators that succeed in simulating any prover message with noticeable probability can be amplified into full-fledged simulators (provided certain independence conditions on the initial success probability). This allows proving that classical protocols, like the polynomial sequential repetition of 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 classical *constant-round* protocols, such as the GK protocol described above. In GK-like 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 by having the simulator take 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.[2] 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.

[2] 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.

1.2.2 Our Technique: A No-Cloning Extraction Procedure

Toward describing the technique, we first restrict attention to a more specific problem. Specifically, constructing a constant-round post-quantum zero-knowledge protocol can be reduced to the problem of 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 S and a receiver R. 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 E that given any malicious sender S^*, represented by a polynomial-size quantum circuit, can simulate the view of S^* in the commitment protocol while extracting the committed plaintext message. Specifically, $E(S^*)$ outputs a classical transcript $\widetilde{T}$, a verifier (quantum) state $|\widetilde{\psi}\rangle$, and an extracted plaintext $\widetilde{m}$ 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 R and sender S^*, 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 β 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 $E(V^*)$, with V^* acting as the sender, to obtain a simulated commitment as well as the corresponding challenge β. Now the simulator knows the challenge ahead of time, before producing the prover message α in step (1), and using the (simulated) verifier state $|\widetilde{\psi}\rangle$, 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.)

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 S^* 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 $\mathbf{CC}[f, u, z]$ is given by a function f (represented as a circuit), a target string u in its range, and a message z; it outputs z on every input x such that $f(x) = u$, and rejects all other inputs. A corresponding obfuscator compiles any such program into a program $\widetilde{\mathbf{CC}}$ with the same functionality. In terms of security, provided that the target u has high entropy conditioned on f and z, the obfuscated program is computationally indistinguishable from a simulated dummy program, independent of (f, u, 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 S samples:
   - two random strings u and v,
   - a secret key sk for an FHE scheme,
   - an FHE encryption $\mathsf{ct}_v = \mathsf{FHE.Enc}_{\mathsf{sk}}(v)$ of v,
   - an obfuscation $\widetilde{\mathbf{CC}}$ of $\mathbf{CC}[f, u, z]$, where $z = (m, \mathsf{sk})$ and $f = \mathsf{FHE.Dec}_{\mathsf{sk}}$ is the FHE decryption circuit.

   It then sends $(\mathsf{ct}_v, \widetilde{\mathbf{CC}})$ to the receiver R.

2. The receiver R sends a guess v'.

3. S rewards a successful guess: if $v = v'$, it sends back u (and otherwise $\bot$).

The described commitment protocol comes close to our objective. First, it is binding — the obfuscation $\widetilde{\mathbf{CC}}$ uniquely determines $z = (m, \mathsf{sk})$. 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 $v' = v$ at the second message, but with negligible probability. Indeed, given only the first sender message $(\mathsf{ct}_v, \widetilde{\mathbf{CC}})$, the receiver obtains no information about u. Hence, we can invoke the CCO security and replace the obfuscation $\widetilde{\mathbf{CC}}$ 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 $(\mathsf{ct}_v, \widetilde{\mathbf{CC}})$ hides v. It follows that the third sender message is $\bot$ (rather than the target u) 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 S^*, can simulate the sender's view, while extracting m. It first runs the sender to obtain the first message $(\mathsf{ct}_v, \widetilde{\mathbf{CC}})$. At this point, it can use the sender's circuit S^* to continue the emulation of S^* *homomorphically under the encryption* ct_v. The key point is that, under the encryption, we do have v. We can (homomorphically) feed v to the sender, and obtain an encryption ct_u of u. Now, the simulator feeds ct_u to the obfuscation $\widetilde{\mathbf{CC}}$, and gets back $z = (m, \mathsf{sk})$. (Note that here the compactness of FHE is crucial — the sender S^* could be of arbitrary polynomial size, whereas $\widetilde{\mathbf{CC}}$ and thus also ct_u are of fixed size.)

Having extracted m, it remains to simulate the inner (for now, classical) state ψ 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 ψ 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 $v' = v$, 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 S samples:
   - two random strings u and v,
   - a secret key sk for an FHE scheme,
   - an FHE encryption $\mathsf{ct}_v = \mathsf{FHE.Enc}_{\mathsf{sk}}(v)$ of v,
   - an obfuscation $\widetilde{\mathbf{CC}}$ of $\mathbf{CC}[f, u, z]$, where $z = (m, \mathsf{sk})$ and $f = \mathsf{FHE.Dec}_{\mathsf{sk}}$ is the FHE decryption circuit.

   It then sends $(\mathsf{ct}_v, \widetilde{\mathbf{CC}})$ to the receiver R.
2. The receiver R sends $\mathsf{ct}'_{v'}$, a guess v' encrypted using SFE. (The honest receiver sets v' arbitrarily.)
3. S homomorphically evaluates the function that given input v, returns u (and otherwise $\bot$).

Indistinguishability of the simulated sender view from the real sender view now follows since the SFE encryption $\mathsf{ct}'_{v'}$ hides v'. The SFE circuit privacy guarantees that the homomorphic SFE evaluation does not leak any information about the target u, as long as the receiver does not send an SFE encryption of v.

A Malleability Problem and its Resolution. While we could argue before that a malicious receiver cannot output v in the clear, arguing that it does not output an SFE encryption of v is more tricky. In particular, the receiver might be able to somehow maul the FHE encryption ct_v to get an SFE encryption ct'_v of the value v, without actually "knowing" the value v. Classically, such malleability problems are solved using *extraction*. If we could efficiently extract the value encrypted in the SFE encryption ct', 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). Here 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 postqantum 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 S^*. 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 S^*. A technical point is that QFHE should support the evaluation of a quantum circuit with an initial quantum state — in our case the quantum sender S^* 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 general adversaries who, with noticeable probability, may abort at some stage of the protocol, existing black-box techniques completely fail. In contrast, as we explain next, our non-black-box technique achieves a meaningful guarantee even against aborting senders.

In our protocol, an aborting sender S^* 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 $\widetilde{\mathbf{CC}}$ to get the decryption key sk). Nevertheless, *in executions where there is no abort, the simulation is valid*. That is, non-aborting simulated executions are indistinguishable from non-aborting real executions; furthermore, they occur with the same probability (up to a negligible difference).

Another feature of the protocol is that if aborting executions occur with noticeable probability, they are easy to simulate by rejection sampling. To simulate an aborting execution, the simulator sends an SFE encryption $\mathsf{ct}'_{v'}$ of some arbitrary string v' (e.g. the all-zero string). Crucially, it does so in the clear, rather than under the encryption. This is repeated until an aborting execution is produced. The hiding of SFE ciphertexts guarantee that this will occur with the right probability.

This gives rise to a full simulation strategy: (1) run the non-black-box simulator. If no abort occurs, simulation succeeds. Otherwise, (2) simulate an abort. Some care has to be taken to make sure that the expected running time of the simulator is polynomial. This is done using standard techniques [GK96a] for estimating the probability of abort.

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 we described crucially relies on this; in particular, the CCO $\widetilde{\mathbf{CC}}$ and the FHE ciphertext ct_v must be formed consistently with each other for the simulator to work. Crucially, it suffices that *there exists an explanation* for the messages, and we do not have to efficiently extract it as part of the simulation;[3] indeed, efficient quantum extraction is exactly the problem we are trying to solve.

[3] 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].

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. Specifically, 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.

A Note on Universal Simulators, Composition, and Verifiers with Quantum Advice. The traditional zero knowledge requirement asks that for any malicious verifier V^*, there exists a simulator Sim_{V^*}. However, most classical protocols achieve a stronger guarantee that flips the order of quantifiers — they provide one *universal* simulator Sim, which works for all verifiers. That is, $\mathsf{Sim}(x, V^*)$ is given as input the instance x in the **NP** language and any circuit V^* representing the verifier, and successfully simulates the view of V^* in a real interaction. This stronger guarantee is especially useful for *composition*, namely, when using the zero-knowledge protocol as a subroutine inside an (outer) protocol. This is because it allows to hardwire the inner state s of the adversary in the (outer) protocol into the description of V^*, and modularly apply the simulation guarantee.

We achieve the same universal guarantee as above with respect to (non-uniform) quantum, rather than classical, circuits. However, in the context of composability in the quantum setting, we may ask for a stronger requirement. Specifically, we would like to be able to "hardwire" into V^* a quantum inner state $|\psi\rangle$, that is not necessarily efficiently generatable. Indeed, if we wish to show post-quantum security of an outer protocol that uses the zero-knowledge protocol as a subroutine, the state of the adversary in the outer protocol is quantum, and when we perform an averaging argument (to modularly use the ZK definition guarantee) and fix a specific quantum state for the verifier, that specific state might not be efficiently generatable. Our protocol falls short of achieving this stronger feature of *universality with respect to quantum advice*. In a nutshell, this is because whenever our non-black-box simulator gets stuck due to an abort, we discard the simulation, and simulate from scratch an abort. The initial quantum state $|\psi\rangle$ of the verifier is disturbed due to the first (stuck) simulation, preventing us from lawfully simulating an abort relative to $V^*(|\psi\rangle)$.

The lack of universality with respect to quantum advice means that when using our zero-knowledge protocol inside another protocol, one has to work harder and essentially reprove simulation validity, rather than modularly use the universal simulation guarantee. We do, however, show a somewhat relaxed form of universality with respect quantum advice, which does make composition more modular, but is not as clean as full universality. The problem of (non-relaxed) universality with respect to quantum advice is left open.

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]. 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 **NP** (in a polynomial number of rounds).

Zero Knowledge for QMA. Another line of work aims at constructing quantum (rather than classical) protocols for **QMA** (rather than **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 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).

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]). 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

All algorithms of cryptographic functionalities in this work are implicitly efficient and classical (i.e. require no quantum computation or a quantum communication channel), unless noted otherwise. We rely on the standard notions of classical Turing machines and Boolean circuits:

- We say that a Turing machine (or algorithm) is PPT if it is probabilistic and runs in polynomial time.
- We sometimes think about PPT Turing machines as polynomial-size uniform families of circuits (as 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_λ is of polynomial size $\lambda^{O(1)}$ and has $\lambda^{O(1)}$ input and output bits. We say that the family is uniform if there exists a polynomial-time deterministic Turing machine M that on input 1^λ outputs C_λ.
- For a PPT Turing machine (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 may write $m \in M(x)$ to denote the fact that m is in the support of $M(x;\cdot)$.

2.1 Quantum Computation

We use standard notions from quantum computation.

- We say that a Turing machine (or algorithm) is QPT if it is quantum and runs in polynomial time.

- We sometimes think about QPT Turing machines as polynomial-size uniform families of quantum circuits (as these are equivalent models). We call a polynomial-size quantum circuit familiy $\mathcal{C} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ uniform if there exists a polynomial-time deterministic Turing machine M that on input 1^λ outputs C_λ.

- Classical communication channels in the quantum setting are identical to classical communication channels in the classical setting, except that when a set of qubits is sent through a classical communication channel, then the qubits are automatically measured in the standard basis, and the measured (now classical-state) qubits are then sent through the channel.

- A quantum interactive algorithm (in a 2-party setting) has input divided into two registers and output divided into two registers. For the input qubits, one register is for an input message from the other party, and a second register is for a potential inner state the machine held. For the output, one register is for the message to be sent to the other party, and another register is for a potential inner state for the machine to keep to itself.

Quantum Adversarial Model. As noted at the end of the technical overview, we would like to consider security definitions that not only achieve quantum security, but are also composable and can be used modularly inside other protocols. For this we think by default on security against polynomial-size quantum adversaries with n.u. polynomial-size quantum advice (i.e. an arbitrary quantum mixed state that's not necessarily efficiently generatable).

In some of the security definitions we will explicitly note if the security is stronger (statistical security, against unbounded algorithms) or standard quantum security (against non-uniform quantum circuits without quantum advice), and in any case, security will always hold at least against non-uniform quantum circuits (without arbitrary quantum advice). Accordingly, an adversary will be usually denoted by $\mathsf{A}^* = \{\mathsf{A}^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, where $\{\mathsf{A}^*_\lambda\}_{\lambda\in\mathbb{N}}$ is a polynomial-size non-uniform sequence of quantum circuits, and $\{\rho_\lambda\}_{\lambda\in\mathbb{N}}$ is some polynomial-size sequence of mixed quantum states. All adversaries are implicitly unrestricted in their behaviour (i.e. they are fully malicious and can arbitrarily deviate from protocols).

We conclude with notions regarding indistinguishability in the quantum setting.

- A function $f : \mathbb{N} \to [0, 1]$ is,
  - negligible if for every constant $c \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n > N$, $f(n) < n^{-c}$.
  - noticeable if there exists $c \in \mathbb{N}, N \in \mathbb{N}$ s.t. for every $n \geq N$, $f(n) \geq n^{-c}$.

- A quantum random variable is simply a random variable that can have values that are quantum states. That is, a quantum r.v. induces a probability distribution over a (possibly infinite) set of quantum states. Such quantum r.v. can be also thought of as a mixed quantum state, which is simply a distribution over quantum states.

- For two quantum random variables X and Y, quantum distinguisher D with quantum mixed state as auxiliary input ρ, and $\mu \in [0, 1]$, we write $X \approx_{\mathsf{D}(\rho),\mu} Y$ if

$$|\Pr[\mathsf{D}(X; \rho) = 1] - \Pr[\mathsf{D}(Y; \rho) = 1]| \leq \mu.$$

- Two ensembles of quantum random variables $\mathcal{X} = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are said to be computationally indistinguishable, denoted by $\mathcal{X} \approx_c \mathcal{Y}$, if for every polynomial-size non-uniform quantum distinguisher with quantum advice $\mathsf{D} = \{\mathsf{D}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$,

$$X_\lambda \approx_{\mathsf{D}_\lambda(\rho_\lambda),\, \mu(\lambda)} Y_\lambda \ .$$

- The trace distance between two quantum distributions X, Y, denoted $\mathrm{TD}(X, Y)$, is a generalization of statistical distance to the quantum setting and represents the maximal distinguishing advantage between two quantum distributions, by unbounded quantum algorithms. We thus say that $\mathcal{X} = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are statistically indistinguishable (and write $\mathcal{X} \approx_s \mathcal{Y}$), if for every unbounded non-uniform quantum distinguisher $\mathsf{D} = \{\mathsf{D}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$,

$$\mathrm{TD}\left(X_\lambda, Y_\lambda\right) \le \mu(\lambda) \ .$$

In what follows, we introduce the cryptographic tools we will use in this work. As a default, all algorithms are classical and efficient unless noted otherwise, and security (which will be formally defined for each primitive) holds against efficient non-uniform quantum adversaries.

2.2 Classical WI Proofs with Quantum Security

We use classical constant-round proof systems for NP (where both honest prover and verifier are classical efficient algorithms) that are witness-indistinguishable, that is, the prover gets the privacy guarantee that the transcripts it generates for two witnesses of the same instance, are indistinguishable (for quantum attackers).

In what follows, we denote by (P, V) a protocol between two parties P and V. For common input x, we denote by $\mathsf{OUT}_\mathsf{V}\langle \mathsf{P}, \mathsf{V}\rangle(x)$ the output of V in the protocol. For honest verifiers, this output will be a single bit indicating acceptance (or rejection), malicious quantum verifiers may have arbitrary quantum output (which is captured by the verifier outputting its inner quantum state at the end of interaction).

Definition 2.1 (WI Proof System for NP)**.** *A protocol* (P, V) *with an honest PPT prover* P *and an honest PPT verifier* V *for a language* $\mathcal{L} \in$ ***NP*** *is a witness-indistinguishable proof system if it satisfies:*

1. **Perfect Completeness:** *For any* $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0,1\}^\lambda, w \in \mathcal{R}_\mathcal{L}(x)$*,*

$$\Pr[\mathsf{OUT}_\mathsf{V}\langle \mathsf{P}(w), \mathsf{V}\rangle(x) = 1] = 1 \ .$$

2. **Statistical Soundness:** *For any non-uniform unbounded prover* $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_\lambda$*, there exists a negligible function* $\mu(\cdot)$ *such that for any security parameter* $\lambda \in \mathbb{N}$ *and any* $x \in \{0,1\}^\lambda \setminus \mathcal{L}$*,*

$$\Pr\left[\mathsf{OUT}_\mathsf{V}\langle \mathsf{P}^*_\lambda, \mathsf{V}\rangle(x) = 1\right] \le \mu(\lambda) \ .$$

3. **Witness Indistinguishability:** *For every non-uniform quantum polynomial-size verifier* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda, \rho_\lambda\}_\lambda$*, for any two sequences of witnesses* $\{w_\lambda\}_{\lambda\in\mathbb{N}}$*,* $\{v_\lambda\}_{\lambda\in\mathbb{N}}$ *s.t. for every* $\lambda \in \mathbb{N}$*,* w_λ *and* v_λ *are both witnesses for the same* $x_\lambda \in \mathcal{L} \cap \{0,1\}^\lambda$*, we have,*

$$\{\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w_\lambda), \mathsf{V}^*_\lambda(\rho_\lambda)\rangle(x_\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(v_\lambda), \mathsf{V}^*_\lambda(\rho_\lambda)\rangle(x_\lambda)\}_{\lambda\in\mathbb{N}} \ .$$

Instantiations of WI proofs. 3-message 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 attacks (rather only against classical attacks).

2.3 Sigma Protocols for NP

We use the abstraction called *Sigma Protocols*, which are public-coin three-message proof systems that only give a very weak zero-knowledge guarantee.

Definition 2.2. *A sigma protocol for an* **NP** *relation $\mathcal{R}$ is a three-message protocol (α, β, γ) between a PPT prover $\Sigma.\mathsf{P}$ and a public-coin PPT verifier $\Sigma.\mathsf{V}$, with the following properties:*

Completeness: *for any $(x, w) \in \mathcal{R}$, $\langle \Sigma.\mathsf{P}(w), \Sigma.\mathsf{V} \rangle(x) = 1$.*

Statistical Soundness: *for any non-uniform unbounded prover $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_\lambda$, there exists a negligible function $\mu(\cdot)$ such that for any security parameter $\lambda \in \mathbb{N}$ and any $x \in \{0,1\}^\lambda \setminus \mathcal{L}$,*

$$\Pr\left[\langle \mathsf{P}^*, \Sigma.\mathsf{V} \rangle(x) = 1\right] \leq \mu(\lambda) \enspace .$$

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

$$\left\{(\alpha, \gamma) \mid (\alpha, r_\alpha) \leftarrow \Sigma.\mathsf{P}_1(x, w), \gamma = \Sigma.\mathsf{P}_3(x, r_\alpha; \beta)\right\}_{\substack{(x,w)\in\mathcal{R} \\ x,\beta\in\{0,1\}^\lambda}} \approx_c \left\{(\alpha, \gamma) \leftarrow \Sigma.\mathsf{S}(x, \beta)\right\}_{\substack{(x,w)\in\mathcal{R} \\ x,\beta\in\{0,1\}^\lambda}} \enspace ,$$

where r_α is the prover's inner state that it prints for itself along with outputting α.

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

Claim 2.1 (First-Message Indistinguishability)**.** *In every Σ protocol:*

$$\left\{\alpha \leftarrow \Sigma.\mathsf{P}_1(x, w)\right\}_{\substack{(x,w)\in\mathcal{R} \\ |x|=\lambda}} \approx_c \left\{\alpha \leftarrow \Sigma.\mathsf{S}_1(x, 0^\lambda)\right\}_{\substack{(x,w)\in\mathcal{R} \\ |x|=\lambda}} \enspace ,$$

where $\Sigma.\mathsf{P}_1(x, w)$ and $\Sigma.\mathsf{S}_1(x, \beta)$ are the distributions on the first message of the prover and simulator.

Proof Sketch. Note that the first prover message is computed independently of the verifier's message. The claim now follows by the special zero-knowledge guarantee. □

Sigma protocols are known to 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 and computationally hiding commitments.

2.4 Compute-and-Compare Obfuscation

We define compute-and-compare (CC) circuits, and obfuscators for CC circuits. We start by defining the class of *compute-and-compare circuits*.

Definition 2.3 (Compute-and-Compare Circuit)**.** *Let $f : \{0,1\}^n \to \{0,1\}^\lambda$ be a circuit, and let $u \in \{0,1\}^\lambda, z \in \{0,1\}^*$ be strings. Then $\mathbf{CC}[f, u, z](x)$ is a circuit that returns z if $f(x) = y$, and $\perp$ otherwise.*

We now define compute-and-compare (CC) obfuscators (with perfect correctness). In what follows Obf is a PPT algorithm that takes as input a CC circuit $\mathbf{CC}[f, u, z]$ and outputs a new circuit $\widetilde{\mathbf{CC}}$. (We assume that the CC circuit $\mathbf{CC}[f, u, z]$ is given in some canonical description from which f, u, and z can be read.)

Definition 2.4 (CC obfuscator)**.** *A PPT algorithm Obf is a compute-and-compare obfuscator if it satisfies:*

1. **Perfect Correctness:** *For any circuit* $f : \{0,1\}^n \to \{0,1\}^\lambda$, $u \in \{0,1\}^\lambda$ *and* $z \in \{0,1\}^*$,

$$\Pr\left[\forall x \in \{0,1\}^n : \widetilde{\mathbf{CC}}(x) = \mathbf{CC}[f,u,z](x) \;\middle|\; \widetilde{\mathbf{CC}} \leftarrow \mathsf{Obf}(\mathbf{CC}[f,u,z])\right] = 1 \ .$$

2. **Simulation:** *There exists a PPT simulator* Sim *such that for any polynomial-size classical circuit family* $f = \{f_\lambda\}_{\lambda\in\mathbb{N}}$ *and polynomial-length output string* $z = \{z_\lambda\}_{\lambda\in\mathbb{N}}$,

$$\{\widetilde{\mathbf{CC}} \mid u \leftarrow U_\lambda, \widetilde{\mathbf{CC}} \leftarrow \mathsf{Obf}(\mathbf{CC}[f_\lambda, u, z_\lambda])\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{Sim}(1^{|f_\lambda|}, 1^{|z_\lambda|}, 1^\lambda)\}_{\lambda\in\mathbb{N}} \ .$$

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

2.5 Non-Interactive Commitments

We define non-interactive commitment schemes with perfect binding and computational hiding (against quantum attacks).

Definition 2.5 (Non-Interactive Commitment)**.** *A non-interactive commitment scheme is given by a PPT algorithm* $\mathsf{Com}(\cdot)$ *with the following syntax:*

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

The commitment algorithm satisfies:

1. **Perfect Binding:** *For any* $x, x' \in \{0,1\}^*$ *of the same length, if* $\mathsf{cmt} \in \mathsf{Com}(1^\lambda, x)$, $\mathsf{cmt} \in \mathsf{Com}(1^\lambda, x')$, *then* $x = x'$.

2. **Quantum Computational Hiding:** *For any pair of* $\ell(\lambda)$*-length strings* $x_0 = \{x_{0,\lambda}\}_{\lambda\in\mathbb{N}}$, $x_1 = \{x_{1,\lambda}\}_{\lambda\in\mathbb{N}}$, *where* $\ell(\lambda)$ *is a polynomial, we have,*

$$\{\mathsf{Com}(1^\lambda, x_{0,\lambda})\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{Com}(1^\lambda, x_{1,\lambda})\}_{\lambda\in\mathbb{N}} \ .$$

Instantiations. The above non-interactive commitments are known based on various standard assumptions, including QLWE [GHKW17, LS19].

2.6 Quantum Fully Homomorphic Encryption

In our protocol we will be using a quantum fully homomorphic encryption scheme, specifically, a scheme where a classical input can be encrypted classically and a quantum input quantumly. Moreover, we can evaluate an encrypted input that was partially encrypted classically and partially quantumly, and if part of the output is classical, it can be decrypted by a classical decryption algorithm. The formal definition follows.

Definition 2.6 (Quantum Fully-Homomorphic Encryption)**.** *A quantum fully homomorphic encryption scheme is given by six algorithms* ($\mathsf{QHE.Keygen}$, $\mathsf{QHE.Enc}$, $\mathsf{QHE.QEnc}$, $\mathsf{QHE.Dec}$, $\mathsf{QHE.QDec}$, $\mathsf{QHE.Eval}$) *with the following syntax:*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QHE.Keygen}(1^\lambda)$: *A PPT algorithm that given a security parameter, samples a classical public key and a classical secret key.*

- $c \leftarrow \mathsf{QHE.Enc}_{\mathsf{pk}}(b)$: *A PPT algorithm that takes as input a bit b and outputs a classical ciphertext.*

- $|\phi\rangle \leftarrow \mathsf{QHE.QEnc}_{\mathsf{pk}}(|\psi\rangle)$: *A QPT algorithm that takes as input a qubit* $|\psi\rangle$ *and outputs a ciphertext represented in qubits.*

- $b \leftarrow \mathsf{QHE.Dec}_{\mathsf{sk}}(c)$: *A PPT algorithm that takes as input a classical ciphertext* c *and outputs a bit.*

- $|\psi\rangle \leftarrow \mathsf{QHE.QDec}_{\mathsf{sk}}(|\phi\rangle)$: *A QPT algorithm that takes as input a quantum ciphertext* $|\phi\rangle$ *and outputs a qubit.*

- $\hat{\mathbf{c}}, |\hat{\Phi}\rangle \leftarrow \mathsf{QHE.Eval}_{\mathsf{pk}}(C, |\Phi\rangle)$: *A QPT algorithm that takes as input:*

  1. *A general quantum circuit* C *with* ℓ *input qubits and* ℓ' *output qubits, out of which* m *are measured.*
  2. *A quantum ciphertext* $|\Phi\rangle$ *that encrypts an* ℓ*-qubit state, some of the* ℓ *ciphertexts are possibly classical ciphertexts (generated by the classical encryption algorithm) encrypting classical bits.*

  The evaluation algorithm outputs a classical ciphertext $\hat{\mathbf{c}}$ *encrypting* m *bits, plus a quantum ciphertext* $|\hat{\Phi}\rangle$ *encrypting an* $(\ell' - m)$*-qubit quantum state.*

The scheme satisfies the following.

- **Quantum Semantic Security:** *The encryption algorithm maintains quantum semantic security.*

- **Compactness:** *There exists a polynomial* $\mathrm{poly}(\cdot)$ *s.t. for every quantum circuit* C *with* ℓ' *output qubits and an enryption of an input for* C*, the output size of the evaluation algorithm is* $\mathrm{poly}(\lambda, \ell')$*, where* λ *is the security parameter of the scheme.*

- **Classicality-Preserving Quantum Homomorphism:** *Let* $\mathcal{C} = \{C_\lambda\}_{\lambda \in \mathbb{N}}$ *be a polynomial-size quantum circuit (where* C_λ *has* $m(\lambda)$ *measured output qubits*[4]*), let* $|\Psi\rangle = \{|\Psi\rangle_\lambda\}_{\lambda \in \mathbb{N}}$ *be an input state for* C*, let* $(\mathsf{pk}, \mathsf{sk}) = \{(\mathsf{pk}_\lambda, \mathsf{sk}_\lambda)\}_{\lambda \in \mathbb{N}}$ *be a pair of public and secret keys (*$\forall \lambda \in \mathbb{N} : (\mathsf{pk}_\lambda, \mathsf{sk}_\lambda) \in \mathsf{QHE.Keygen}(1^\lambda)$*) and let* $r = \{r_\lambda\}_\lambda$ *be randomness for the encryption algorithm. Then there exists a negligible function* $\mu(\cdot)$ *s.t. for all* $\lambda \in \mathbb{N}$*,*

  $$\mathrm{TD}\left(\rho_{0,\lambda}, \rho_{1,\lambda}\right) \leq \mu(\lambda) \ ,$$

  where ρ_0, ρ_1 *are the quantum distributions which are defined as follows:*

  - $\rho_{0,\lambda}$: *Encrypt each classical bit of* $|\Psi\rangle$ *with* $\mathsf{QHE.Enc}_{\mathsf{pk}}(\cdot)$ *and the rest with* $\mathsf{QHE.QEnc}_{\mathsf{pk}}(\cdot)$ *(the randomness* r *is the total randomness used for all encryptions). Execute* $\mathsf{QHE.Eval}_{\mathsf{pk}}(C, \cdot)$ *on the encryption to get* $\hat{\mathbf{c}}, |\hat{\Phi}\rangle$*, where* $\hat{\mathbf{c}}$ *is a classical ciphertext encrypting* $m(\lambda)$ *bits. Then output* $\mathsf{QHE.Dec}_{\mathsf{sk}}\left(\hat{\mathbf{c}}\right), \mathsf{QHE.QDec}_{\mathsf{sk}}(|\hat{\Phi}\rangle)$*.*
  - $\rho_{1,\lambda}$: *Output* $C\left(|\psi\rangle\right)$*.*

Comments about the definition:

- For the definition of quantum semantic security see [BJ15].

[4] Out of some total number of output qubits $\ell'(\lambda)$.

- A circuit with m measured output qubits is one where the leftmost m output qubits are measured in the standard basis before they are outputted, the rest of the output qubits can either be measured or not. The choice of the classical bits being lefmost is arbitrary and is only for the sake of simplicity. If desired, our model (with the classical output on the left) can be adapted by adding swap gates to circuits with spread-out classical output (just before the end of computation).
- It might look weird at first that we don't let the evaluation algorithm $\mathsf{QHE.Eval}$ explicitly know which of the ciphertexts its about to process are classical. This is not a problem, since the evaluation algorithm treats quantum ciphertexts and classical ciphertexts all the same; they take the same amount of qubits, and the algorithm performs on them exactly the same computations.

Instantiations. Mahadev and Brakerski [Mah18a, Bra18] show how to build quantum FHE from standard cryptographic assumptions; QLWE, plus a circular security assumption. The above definition is more general then how quantum FHE is defined in these works. Specifically, *classicality-preserving* quantum homomorphism with statistical correctness for *every* pair $(\mathsf{pk}, \mathsf{sk})$ of public and secret keys and randomness r for the encryption algorithm, is not explicitly defined there. However, the construction of Brakerski actually does achieve this more general definition. This follows readily from the main theorem (4.1) in [Bra18].

2.7 Function-Hiding Secure Function Evaluation

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

Definition 2.7 (2-Message Function Hiding SFE)**.** *A two-message secure function evaluation protocol* $(\mathsf{SFE.Gen}, \mathsf{SFE.Enc}, \mathsf{SFE.Eval}, \mathsf{SFE.Dec})$ *has the following syntax:*

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

For any polynomial-size family of classical circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ (for every $\lambda \in \mathbb{N}$, $\mathcal{C}_\lambda$ is a set of circuits) the scheme satisfies:

- **Perfect Correctness:** *For any $\lambda \in \mathbb{N}$, $x \in \{0,1\}^*$ and circuit $C \in \mathcal{C}_\lambda$,*

$$\Pr\left[\mathsf{SFE.Dec}_{\mathsf{dk}}(\hat{\mathsf{ct}}) = C(x) \;\middle|\; \begin{array}{l} \mathsf{dk} \leftarrow \mathsf{SFE.Gen}(1^\lambda), \\ \mathsf{ct} \leftarrow \mathsf{SFE.Enc}_{\mathsf{dk}}(x), \\ \hat{\mathsf{ct}} \leftarrow \mathsf{SFE.Eval}(C, \mathsf{ct}) \end{array}\right] = 1 \ .$$

- **Quantum Input Privacy:** *For any polynomial $\ell(\lambda)$ and polynomial-size quantum adversary $\mathsf{A}^* = \{\mathsf{A}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$, there exists a negligible function $\mu(\cdot)$ such that for every two length-$\ell(\lambda)$ messages $\{x_{0,\lambda}\}_{\lambda \in \mathbb{N}}$, $\{x_{1,\lambda}\}_{\lambda \in \mathbb{N}}$, for every $\lambda \in \mathbb{N}$:*

$$\Pr\left[\mathsf{A}^*_\lambda(\mathsf{ct}) = b \;\middle|\; \begin{array}{l} b \leftarrow \{0,1\}, \\ \mathsf{dk} \leftarrow \mathsf{SFE.Gen}(1^\lambda), \\ \mathsf{ct} \leftarrow \mathsf{SFE.Enc}_{\mathsf{dk}}(x_b) \end{array}\right] \leq \frac{1}{2} + \mu(\lambda) \ .$$

- **Statistical Circuit Privacy:** *There exist unbounded algorithms, probabilistic* Sim *and deterministic* Ext*, such that for every* $x \in \{0,1\}^*$*,* $\mathsf{ct} \in \mathsf{SFE.Enc}(x)$*, the extractor outputs* $\mathsf{Ext}(\mathsf{ct}) = x$*, and:*

$$\{\mathsf{SFE.Eval}(C, \mathsf{ct}^*)\}_{\substack{\lambda\in\mathbb{N}, C\in\mathcal{C}_\lambda, \\ \mathsf{ct}^*\in\{0,1\}^{\mathrm{poly}(\lambda)}}} \approx_s \{\mathsf{Sim}(C(\mathsf{Ext}(\mathsf{ct}^*;1^\lambda));1^\lambda)\}_{\substack{\lambda\in\mathbb{N}, C\in\mathcal{C}_\lambda, \\ \mathsf{ct}^*\in\{0,1\}^{\mathrm{poly}(\lambda)}}} \ .$$

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)**.** *Let* $\mathsf{ct}^* = \{\mathsf{ct}^*_\lambda\}_{\lambda\in\mathbb{N}}$ *be a (possibly non-ciphertext)* $\mathrm{poly}(\lambda)$*-length string, and let* $C_0 = \{C_{0,\lambda}\}_{\lambda\in\mathbb{N}}$*,* $C_1 = \{C_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ *be two circuits such that* $\forall\lambda \in \mathbb{N} : C_{0,\lambda}(\mathsf{Ext}(\mathsf{ct}^*_\lambda;1^\lambda)) = C_{1,\lambda}(\mathsf{Ext}(\mathsf{ct}^*_\lambda;1^\lambda))$*. Then*

$$\{\mathsf{SFE.Eval}(C_{0,\lambda}, \mathsf{ct}^*_\lambda)\}_{\lambda\in\mathbb{N}} \approx_s \{\mathsf{SFE.Eval}(C_{1,\lambda}, \mathsf{ct}^*_\lambda)\}_{\lambda\in\mathbb{N}} \ .$$

Such secure function evaluation schemes are known based on post-quantum hardness of QLWE [OPCPC14, BD18].

3 Constant-Round Zero-Knowledge Arguments for NP with Quantum Security

In this section we construct a classical argument system for an arbitrary NP language $\mathcal{L}$, with a constant number of rounds, quantum soundness and quantum zero-knowledge. More formally, we construct a constant-round protocol satisfying the following definition (in the definition below, $\mathsf{OUT}_{\mathsf{V}^*}\langle\mathsf{P}(w), \mathsf{V}^*\rangle(x)$ denotes the interaction transcript between $\mathsf{P}(w)$ and V^*, and V^*'s inner quantum state at the end of this interaction).

Definition 3.1 (Post-Quantum Zero-Knowledge Argument for NP)**.** *A classical protocol* (P, V) *with an honest PPT prover* P *and an honest PPT verifier* V *for a language* $\mathcal{L} \in$ ***NP*** *is a quantum-secure zero-knowledge argument if it satisfies:*

1. **Perfect Completeness:** *For any* $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0,1\}^\lambda, w \in \mathcal{R}_\mathcal{L}(x)$*,*

$$\Pr[\mathsf{OUT}_\mathsf{V}\langle\mathsf{P}(w), \mathsf{V}\rangle(x) = 1] = 1 \ .$$

2. **Quantum Computational Soundness:** *For any non-uniform quantum polynomial-size prover* $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_{\lambda\in\mathbb{N}}$*, there exists a negligible function* $\mu(\cdot)$ *such that for any security parameter* $\lambda \in \mathbb{N}$ *and any* $x \in \{0,1\}^\lambda \setminus \mathcal{L}$*,*

$$\Pr\left[\mathsf{OUT}_\mathsf{V}\langle\mathsf{P}^*_\lambda, \mathsf{V}\rangle(x) = 1\right] \le \mu(\lambda) \ .$$

3. **Quantum Computational Zero-Knowledge:** *There exists a quantum expected polynomial-time simulator* Sim*, such that for any non-uniform quantum polynomial-size verifier* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda\in\mathbb{N}}$*,*

$$\{\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle\mathsf{P}(w), \mathsf{V}^*_\lambda\rangle(x)\}_{\substack{\lambda\in\mathbb{N}, \\ x\in\mathcal{L}\cap\{0,1\}^\lambda, \\ w\in\mathcal{R}_\mathcal{L}(x)}} \approx_c \{\mathsf{Sim}(x, \mathsf{V}^*_\lambda)\}_{\substack{\lambda\in\mathbb{N}, \\ x\in\mathcal{L}\cap\{0,1\}^\lambda, \\ w\in\mathcal{R}_\mathcal{L}(x)}} \ .$$

   - *If the verifier is a classical circuit, then the simulator is computable by a classical expected polynomial-time algorithm.*

As mentioned in the technical overview and preliminaries, we actually prove a definiton a bit stronger then post-quantum security, that is also quantumly-semi-composable. The definition makes it possible to use the protocol as a subprotocol inside an outer protocol, in particular enabling to prove security against quantum adversaries with quantum advice, in many cryptographic settings. Roughly, the stronger definition states that even if the verifier has an arbitrary polynomial-size quantum advice, as long as the simulator has oracle access to samples of pairs of instance (for the zero-knowledge protocol) and quantum advice (for the malicious verifier), protocol can be simulated. On the soundness side, security simply holds against provers with quantum advice. The definition is given below.

Definition 3.2 (Instance-Advice Distribution)**.** *Let $\rho = \{\rho_\lambda\}_{\lambda\in\mathbb{N}}$ be a sequence of polynomial-size quantum mixed states. We say that ρ is an instance-advice distribution (for the language $\mathcal{L} \in$ **NP**) if for every $\lambda \in \mathbb{N}$ there exists a witness $w_\lambda \in \mathcal{R}_\mathcal{L}(\lambda)$*[5] *s.t. the first λ qubits of ρ_λ always contain an instance $x \in \mathcal{L}, |x| = \lambda$ s.t. $w_\lambda \in \mathcal{R}_\mathcal{L}(x)$.*

Definition 3.3 (Quantum-Semi-Composable (QSC) Zero-Knowledge Argument for NP)**.** *A classical protocol (P, V) with an honest PPT prover P and an honest PPT verifier V for a language $\mathcal{L} \in$ **NP** is a quantum-semi-composable (QSC) zero-knowledge argument if it satisfies:*

1. **Perfect Completeness:** *As in the above definition.*

2. **Quantum (Advice) Computational Soundness:** *For any non-uniform quantum polynomial-size prover $\mathsf{P}^* = \{\mathsf{P}^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ (with advice), there exists a negligible function $\mu(\cdot)$ such that for any security parameter $\lambda \in \mathbb{N}$ and any $x \in \{0,1\}^\lambda \setminus \mathcal{L}$,*

$$\Pr\left[\mathsf{OUT}_\mathsf{V}\langle \mathsf{P}^*_\lambda(\rho_\lambda), \mathsf{V}\rangle(x) = 1\right] \leq \mu(\lambda) \ .$$

3. **Quantum-Semi-Composable Computational Zero-Knowledge:** *There exists a quantum expected polynomial-time simulator Sim, such that for any polynomial-size quantum verifier $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda\in\mathbb{N}}$ and polynomial-size instance-advice distribution $\rho = \{\rho_\lambda\}_{\lambda\in\mathbb{N}}$,*

$$\{x, \mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w_\lambda), \mathsf{V}^*_\lambda(\rho^{(x)}_\lambda)\rangle(x) \mid (x, \rho^{(x)}_\lambda) \leftarrow \rho_\lambda\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{Sim}(\mathsf{V}^*_\lambda, O_{\rho_\lambda})\}_{\lambda\in\mathbb{N}} \ ,$$

where,

- *For a mixed state σ, O_σ is an inputless oracle that upon activation returns a sample σ.*

The above definition is clearly stronger then the standard one, as for a verifier without quantum advice, the simulator always "have oracle access to the verifier's advice". As for why this definition is somewhat composable inside quantum-secure protocols, in section 4 we use the semi-composable definition as part of the security proof of the quantum-secure (fully-simulatable) coin-flipping protocol, and it is made clear there how to use the definition. As a side note, it is worth noting that in our construction, proving that the protocol is QSC ZK rather than standard post-quantum ZK is practically the same.

Ingredients and notation:

- A non-interactive commitment scheme Com.
- A CC obfuscator Obf.
- A quantum fully homomorphic encryption scheme ($\mathsf{QHE.Keygen}$, $\mathsf{QHE.Enc}$, $\mathsf{QHE.QEnc}$, $\mathsf{QHE.Dec}$, $\mathsf{QHE.QDec}$, $\mathsf{QHE.Eval}$).

[5] $\mathcal{R}_\mathcal{L}(\lambda)$ is the set of all witnesses of λ-sized instances, that is, $w \in \mathcal{R}_\mathcal{L}(\lambda)$ iff there exists $x \in \mathcal{L} \cap \{0,1\}^\lambda$ s.t. $w \in \mathcal{R}_\mathcal{L}(x)$.

- A 2-message function-hiding secure function evaluation scheme $(\mathsf{SFE.Gen}, \mathsf{SFE.Enc}, \mathsf{SFE.Eval}, \mathsf{SFE.Dec})$.

- A public-coin 3-message WI proof $(\mathsf{WI.P}, \mathsf{WI.V})$.

- A 3-message sigma protocol $(\Sigma.\mathsf{P}, \Sigma.\mathsf{V})$ for the language $\mathcal{L}$ with quantum security.

Handling Aborts and Alikes. For the simplicity of describing the protocol and to avoid distractions from unimportant technical issues, we set a convention to handle publicly checkable misbehaviors by the parties in the protocol.

For a security parameter λ, for each message in the protocol, it is known (publicly) based on λ, what is the length of each message (or upper and lower bounds on that length). If a party sends a message in an incorrect length, the receiving party fixes it locally and trivially; if the message is too long, it cuts the message in a suitable place, and if it's too short then pads with zeros.

In the below protocol, whenever a party either aborts or fails to prove its statement in its WI proof (which is publicly verifiable), the other party ends communication, and if the party that ends communication is the verifier, it also rejects.

We describe the protocol in Figure 1.

3.1 Quantum Soundness

We will reduce the soundness of the sigma protocol to the soundness of the ZK protocol.

Lemma 3.1. *Protocol 1 has quantum soundness.*

Proof. Assume towards contradiction there exists a quantum prover $\mathsf{P}^* = \{\mathsf{P}^*_\lambda, \rho'_\lambda\}_{\lambda \in \mathbb{N}}$ that (for infinitely many $\lambda \in \mathbb{N}$) breaks soundness with noticeable probability. We construct a successfully cheating quantum prover $\Sigma.\mathsf{P}^* = \{\Sigma.\mathsf{P}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ for the sigma protocol. $\Sigma.\mathsf{P}^*$ will be a quantum circuit that uses P^* to interact with $\Sigma.\mathsf{V}$ and to break soundness of the sigma protocol, by simulating to P^* the honest verifier's V responses.

Using a simple averaging argument, we can assume w.l.o.g. that $\mathsf{P}^* = \{\mathsf{P}^*_\lambda, \rho'_\lambda\}_{\lambda \in \mathbb{N}}$ always sends a fixed first message (which is $\mathsf{cmt}_1, \mathsf{cmt}_2$), as we can consider the message and inner quantum state that maximizes the malicious prover's probability to cheat (this is enabled because this is the first protocol message and does not depend on a message from the verifier). We further add that because P^* succeeds in convincing V to accept instances that are not in $\mathcal{L}$ with a noticeable probability, and by the soundness of the WI proof that the prover gives in step 5 of the protocol, it is in particular guaranteed that the commitment cmt_1 that P^* sends in the first message is valid, that is, there are fixed string and randomness $z, r_z \in \{0,1\}^*$ s.t. $\mathsf{cmt}_1 = \mathsf{Com}(1^\lambda, z; r_z)$. As part of the non-uniform quantum advice ρ of $\Sigma.\mathsf{P}^*$, it will have the quantum advice of P^*, plus the information z, r_z.

The operation of $\Sigma.\mathsf{P}^*$ is described below. All operations of P^* below are implicitly executed by $\Sigma.\mathsf{P}^*$ (who also keeps track of the quantum inner state of P^* and executes P^* with that state) and the messages that P^* "sends" are obtained by $\Sigma.\mathsf{P}^*$.

$\Sigma.\mathsf{P}^*(\rho)$**:**

1. P^* sends $\mathsf{cmt}_1, \mathsf{cmt}_2$.

2. $\Sigma.\mathsf{P}^*$ takes the role of V during step 2 and interacts with P^*, with two changes:

   - $\Sigma.\mathsf{P}^*$ does not compute β, and in step 2a it just uses $0^{|\beta|}$.
   - In step 2c it does not actually send an SFE evaluation of $\mathbf{CC}[\mathsf{Id}(\cdot), t, s]$, but instead performs an SFE evaluation of $C_\perp$, a circuit that always outputs $\perp$.

Protocol 1

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

P**'s private input:** A polynomial-size classical witness $w \in \mathcal{R}_\mathcal{L}(x)$ showing that $x \in \mathcal{L}$.

1. **Initial Commitment by the Prover:** P computes $\mathsf{dk} \leftarrow \mathsf{SFE.Gen}(1^\lambda)$ and sends non-interactive commitments to the witness w and the SFE secret key dk: $\mathsf{cmt}_1 \leftarrow \mathsf{Com}(1^\lambda, w)$, $\mathsf{cmt}_2 \leftarrow \mathsf{Com}(1^\lambda, \mathsf{dk})$.

2. **Extractable Commitment to a Challenge by the Verifier:** V computes a challenge $\beta \leftarrow \Sigma.\mathsf{V}$, and proceeds to the following.

   (a) V computes $s \leftarrow U_\lambda, t \leftarrow U_\lambda, (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QHE.Keygen}(1^\lambda)$ and sends

   $$\mathsf{pk},\ \ \mathsf{ct}_\mathsf{V} \leftarrow \mathsf{QHE.Enc}_\mathsf{pk}(t),\ \ \widetilde{\mathbf{CC}} \leftarrow \mathsf{Obf}\Big(\mathbf{CC}\big[\mathsf{QHE.Dec}_\mathsf{sk}(\cdot), s, (\mathsf{sk}, \beta)\big]\Big)\ .$$

   (b) P sends $\mathsf{ct}_\mathsf{P} \leftarrow \mathsf{SFE.Enc}_\mathsf{dk}(0^\lambda)$, where dk is the SFE key that the prover sent inside cmt_2.

   (c) V sends $\hat{\mathsf{ct}} \leftarrow \mathsf{SFE.Eval}\Big(\mathbf{CC}\big[\mathsf{Id}(\cdot), t, s\big],\ \mathsf{ct}_\mathsf{P}\Big)$, where $\mathsf{Id}(\cdot)$ is the identity function.

3. **Sigma Protocol Messages Exchange:**

   (a) P computes $\alpha \leftarrow \Sigma.\mathsf{P}(x, w)$ and sends α.

   (b) V sends the public-coin challenge β itself.

4. **WI Proof by the Verifier:** V interacts with P through $(\mathsf{WI.P}_\mathsf{V}, \mathsf{WI.V}_\mathsf{V})$ to give P a WI proof of the following statement:

   - The transcript of the verifier so far is explainable.
   - **Or,** there is a non-witness $u \notin \mathcal{R}_\mathcal{L}(x)$ and randomness $r \in \{0,1\}^*$ s.t. $\mathsf{cmt}_1 = \mathsf{Com}(1^\lambda, u; r)$.

5. **WI Proof by the Prover:** P interacts with V through $(\mathsf{WI.P}_\mathsf{P}, \mathsf{WI.V}_\mathsf{P})$ and uses the witness w in order to prove to V the following:

   - cmt_1, cmt_2 are both valid commitments and furthermore cmt_2 is a commitment to the SFE key from the verifier's extractable commitment. That is, there exist $z, \mathsf{dk}', y, r_1, r_2, r_{\mathsf{SFE.Gen}}, r_{\mathsf{SFE.Enc}} \in \{0,1\}^*$ s.t. $\mathsf{cmt}_1 = \mathsf{Com}(1^\lambda, z; r_1)$, $\mathsf{cmt}_2 = \mathsf{Com}(1^\lambda, \mathsf{dk}'; r_2)$, $\mathsf{dk}' = \mathsf{SFE.Gen}(1^\lambda; r_{\mathsf{SFE.Gen}})$, $\mathsf{ct}_\mathsf{P} = \mathsf{SFE.Enc}_{\mathsf{dk}'}(y; r_{\mathsf{SFE.Enc}})$.
   - **Or,** $x \in \mathcal{L}$.

6. **Information Reveal by the Prover:** P sends $\gamma = \Sigma.\mathsf{P}(x, w; \alpha, \beta)$.

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

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

At the end of this step, $\Sigma.\mathsf{P}^*$ starts communicating with the sigma protocol verifier $\Sigma.\mathsf{V}$.

3. P^* sends α, $\Sigma.\mathsf{P}^*$ delivers the message to $\Sigma.\mathsf{V}$.

4. $\Sigma.\mathsf{V}$ answers with β, $\Sigma.\mathsf{P}^*$ delivers the message to P^*.

5. $\Sigma.\mathsf{P}^*$ takes the role of V in step 4 of the protocol and communicates with P^*, and uses the information (z, r_z) as witness for the WI statement (z is necessarily a non-witness for x, because $x \notin \mathcal{L}$).

6. $\Sigma.\mathsf{P}^*$ takes the role of V and interacts with P^* while P^* gives a WI proof.

7. P^* sends γ, $\Sigma.\mathsf{P}^*$ delivers it to $\Sigma.\mathsf{V}$.

In order to show that the soundness reduction works, it will be enough to show that the transcripts that are generated by the reduction procedure, conditioned that P^* succeeds to prove its WI statement (in step 5), are computationally indistinguishable from transcripts generated by the real interaction between P^* and V (under the same condition that P^* succeeds to prove its WI statement), this is captured by Claim 3.1.

Note that if we prove this indistinguishability, our soundness proof is finished. To see this, observe that if the claim is true then in particular the probability that at the end of the real interaction we have $1 = \Sigma.\mathsf{V}(\alpha, \beta, \gamma)$ is negligibly close to the probability that in the reduction interaction we have $1 = \Sigma.\mathsf{V}(\alpha, \beta, \gamma)$ (note that V accepts only if the WI proof by the prover was accepted). Because we assume (towards contradiction) that the probability for $1 = \Sigma.\mathsf{V}(\alpha, \beta, \gamma)$ in the real interaction is noticeable, the probability for this event will also be noticeable in the reduction setting, which implies that $\Sigma.\mathsf{P}^*$ breaks the soundness of the sigma protocol, in contradiction. Thus all that remains is to prove Claim 3.1. □

Claim 3.1 (Soundness Reduction Transcripts Indistinguishability)**.** *Let* $\{\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*_\lambda(\rho'_\lambda), \mathsf{V}\rangle_\perp(x_\lambda)\}_{\lambda\in\mathbb{N}}$ *denote the distribution over transcripts and inner quantum state of* P^* *generated by interaction between the malicious prover and the honest verifier of the ZK protocol, s.t. if* P^* *fails to prove its WI statement, the process output is* $\perp$*. Let* $\{\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*_\lambda(\rho'_\lambda), \Sigma.\mathsf{P}^*_\lambda(\rho_\lambda)\rangle_\perp(x_\lambda)\}_{\lambda\in\mathbb{N}}$ *denote the distribution over transcripts and inner quantum state of* P^* *generated by the interaction in the soundness reduction described above, s.t. if* P^* *fails to prove its WI statement, the process output is* $\perp$*. Then,*

$$\{\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*_\lambda(\rho'_\lambda), \mathsf{V}\rangle_\perp(x_\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*_\lambda(\rho'_\lambda), \Sigma.\mathsf{P}^*_\lambda(\rho_\lambda)\rangle_\perp(x_\lambda)\}_{\lambda\in\mathbb{N}} \ .$$

Proof. Define the following hybrid distributions on transcripts.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle^{(1)}_\perp$: A process that acts like $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_\perp$, except that in step 4, V uses the information (z, r_z) as a witness for its WI statement, instead of the witness that shows its transcript is explainable.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle^{(2)}_\perp$: Recall that we assume w.l.o.g. that P^* sends valid commitments $\mathsf{cmt}_1, \mathsf{cmt}_2$ in its first message, in particular there's an SFE secret key $\mathsf{dk} \in \mathsf{SFE.Gen}(1^\lambda)$ s.t. $\mathsf{cmt}_2 \in \mathsf{Com}(1^\lambda, \mathsf{dk})$. Now we can describe the current process: In this process, V has also the information dk, and the process acts like $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle^{(1)}_\perp$, except that when V gets the prover message $\mathsf{ct}_{\mathsf{P}^*}$ in step 2b, it performs the following check: If $t = \mathsf{SFE.Dec}_{\mathsf{dk}}(\mathsf{ct}_{\mathsf{P}^*})$ then the process halts and outputs $\perp$, otherwise the interaction carries on regularly.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle^{(3)}_\perp$: A process that acts like $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle^{(2)}_\perp$, except that in step 2c, instead of performing an SFE evaluation of $\mathbf{CC}[\mathsf{Id}(\cdot), t, s]$, the verifier performs an SFE evaluation of $C_\perp$.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle^{(4)}_\perp$: Similar to $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle^{(3)}_\perp$ except that the verifier does not perform the check at step 2b like described in $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle^{(2)}_\perp$.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(5)}$: In this process, V acts as in $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(4)}$, but with one change; in step 2a it uses $0^{|\beta|}$ instead of the β it computed with $\Sigma.\mathsf{V}$. Observe that this distribution is exactly $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \Sigma.\mathsf{P}^*\rangle_{\perp}$.

We now explain why $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}$ is computationally indistinguishable from $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(1)}$, and why each consecutive pair of distributions are computationally indistinguishable. By the transitivity of computational indistinguisability, this finishes our proof.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp} \approx_c \mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(1)}$: Follows from the witness indistinguishability property of the WI proof that the verifier gives.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(1)} \approx_s \mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(2)}$: This indistinguishability is in fact also a statistical one. This follows from Claim 3.2, which basically says that the probability that $\mathsf{ct}_{\mathsf{P}^*}$ is an encryption of t with the secret key dk is negligible, and thus the erasure of such cases can't be noticed by a distinguisher.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(2)} \approx_s \mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(3)}$: As a basic explanation, this statistical indistinguishability follows from the combination of the circuit privacy property of the SFE (specifically, Claim 2.2) and the soundness of the WI proof that the prover gives.

  As a fuller explanation, assume towards 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 2b, which maximizes D^*'s distinguisability adventage - this transcript fixes in particular the prover's ciphertext $\mathsf{ct}_{\mathsf{P}^*}$, and t, s, which in turn fix the circuit $\mathbf{CC}\left[\mathsf{Id}(\cdot), t, s\right]$. We now consider three cases, and explain why we get a contradiction in each of them.

  1. $\mathsf{ct}_{\mathsf{P}^*} \in \mathsf{SFE.Enc}_{\mathsf{dk}}(t)$: In this case, no matter what will be generated next in the transcript, the output will be $\perp$ (by the check described in the description of $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(2)}$), thus it is impossible to distinguish the outputs of the two processes and we get a contradiction.
  2. $\exists y \in \{0,1\}^{\lambda} \setminus \{t\}$ s.t. $\mathsf{ct}_{\mathsf{P}^*} \in \mathsf{SFE.Enc}_{\mathsf{dk}}(y)$: In this case, $\perp = \mathbf{CC}\left[\mathsf{Id}(\cdot), t, s\right](y)$ and thus we get a contradiction by using the circuit privacy property of the SFE (Claim 2.2).
  3. Else: In that case, either $\mathsf{ct}_{\mathsf{P}^*}$ is a ciphertext encrypted with some other SFE key dk', or it's not a ciphertext at all. In these cases, the WI statement of the prover is necessarily false, and thus a non-$\perp$ 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 to the assumption that D^* distinguishes with noticeable probability.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(3)} \approx_s \mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(4)}$: Follows from the same reasoning as in the indistinguishability $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(1)} \approx_s \mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(2)}$.

- $\mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(4)} \approx_c \mathsf{VIEW}_{\mathsf{P}^*}\langle \mathsf{P}^*, \mathsf{V}\rangle_{\perp}^{(5)}$: Follows from the simulation property (obfuscation security) of the CC obfuscation scheme.

□

Claim 3.2 (Producing an SFE Encryption of t with dk is Hard)**.** *There's a negligible function $\mu(\cdot)$ s.t. the probability that in step 2b, P^* sends $\mathsf{ct}_{\mathsf{P}^*}$ s.t. $t = \mathsf{SFE.Dec}_{\mathsf{dk}}(\mathsf{ct}_{\mathsf{P}^*})$ (where dk is the SFE key inside cmt_2), is bounded by $\mu(\lambda)$.*

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

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

$$\mathsf{pk},\ \mathsf{ct},\ \mathsf{Sim}^{\mathsf{CC}}(1^{|\mathsf{QHE.Dec}_{0^{|\mathsf{sk}|}}|}, 1^{|\mathsf{sk}|+|\beta|}, 1^\lambda)\ ,$$

where $\mathsf{QHE.Dec}_{0^{|\mathsf{sk}|}}$ is the (classical) decryption circuit of the QHE, where the secret key input is hard-wired to be $0^{|\mathsf{sk}|}$. P^* will respond with $\mathsf{ct}_{\mathsf{P}^*}$, and A^* uses dk to output $\mathsf{SFE.Dec}_{\mathsf{dk}}(\mathsf{ct}_{\mathsf{P}^*})$.

We now use the simulation property guarantee of the CC obfuscation: Note that the probability that P^* outputs $\mathsf{ct}_{\mathsf{P}^*}$ s.t. $\mathsf{SFE.Dec}_{\mathsf{dk}}(\mathsf{ct}_{\mathsf{P}^*}) = t$ in the simulated setting, where A^* sends $\mathsf{Sim}^{\mathsf{CC}}(1^{|\mathsf{QHE.Dec}_{0^{|\mathsf{sk}|}}|}, 1^{|\mathsf{sk}|+|\beta|}, 1^\lambda)$ instead of $\widetilde{\mathbf{CC}}$, is negligibly close to the probability that it outputs $\mathsf{ct}_{\mathsf{P}^*}$ s.t. $\mathsf{SFE.Dec}_{\mathsf{dk}}(\mathsf{ct}_{\mathsf{P}^*}) = t$ in the regular setting where it gets $\widetilde{\mathbf{CC}}$ - this is due to the security of the CC obfuscator. Because we know that in the regular interaction, P^* sends $\mathsf{ct}_{\mathsf{P}^*}$ s.t. $t = \mathsf{SFE.Dec}_{\mathsf{dk}}(\mathsf{ct}_{\mathsf{P}^*})$ with a noticeable probability (for infinitely many security parameters), this implies that so does A^*, in contradiction. □

3.2 Quantum Zero-Knowledge

We construct a simulator Sim that for every quantum verifier V^* and an instance-advice distribution ρ, simulates an output distribution that is computationally indistinguishable from the output distribution generated by choosing a random instance and advice $(x, \rho^{(x)})$ from ρ, and then executing the interaction between the honest prover and $\mathsf{V}^*(\rho^{(x)})$. Sim uses as input the verifier's circuit and an access to copies of ρ, and is described below. Every time the verifier circuit V^* is executed, it is implicitly the simulator Sim executing it, who also keeps track of the quantum inner state of V^* during the simulated interaction.

Handling Aborts in Simulation. In the below simulation, the simulator can halt simulation and choose to "simulate an abort" - this means it discards all information kept so far, and then executs $\mathsf{SimAbort}(\mathsf{V}^*, O_\rho)$. This happens when the verifier either aborts, or fails to prove the statement in its WI proof (for messages of incorrect length, the simulator acts like the prover and trivially corrects the messages).

$\mathsf{Sim}(\mathsf{V}^*, O_\rho)$:

1. **Simulation of Initial Commitments:** Sim samples $(x, \rho^{(x)}) \leftarrow O_\rho$, sets $\rho^{(x)}$ to be the inner state of V^* (and sets x to be at the beginning of the simulator's output transcript). Sim then computes $\mathsf{dk} \leftarrow \mathsf{SFE.Gen}(1^\lambda)$ and sends to V^* the commitments $\mathsf{cmt}_1 \leftarrow \mathsf{Com}(1^\lambda, 0^{|w|})$, $\mathsf{cmt}_2 \leftarrow \mathsf{Com}(1^\lambda, \mathsf{dk})$.

2. **Extraction Attempt:**

   (a) V^* sends $\mathsf{pk}, \mathsf{ct}_{\mathsf{V}^*}, \widetilde{\mathbf{CC}}$, and also outputs an inner state $|\psi\rangle$.

   (b) Sim performs a non-black-box step and uses the circuit V^*:

i. Sim computes $\mathsf{ct}_{\mathsf{V}^*}^{\mathrm{SFE}} \leftarrow \mathsf{QHE.Eval}_{\mathsf{pk}}(\mathsf{SFE.Enc}_{\mathsf{dk}}(\cdot),\ \mathsf{ct}_{\mathsf{V}^*})$, where dk is the SFE key that is inside cmt_2. Sim also computes $\mathsf{ct}_{|\psi\rangle} \leftarrow \mathsf{QHE.QEnc}_{\mathsf{pk}}(|\psi\rangle)$.
Observe that at this point, if $\mathsf{ct}_{\mathsf{V}^*}$ is indeed a QHE encryption of some t (using the public key pk), then the joint ciphertext $(\mathsf{ct}_{\mathsf{V}^*}^{\mathrm{SFE}}, \mathsf{ct}_{|\psi\rangle})$ encrypts $(\mathsf{SFE.Enc}_{\mathsf{dk}}(t), |\psi\rangle)$.

ii. Sim performs a quantum homomorphic evaluation of the verifier's response. It computes,

$$\left(\mathsf{ct}_s^{\mathrm{SFE}}, \mathsf{ct}_{|\phi\rangle}\right) \leftarrow \mathsf{QHE.Eval}_{\mathsf{pk}}\left(\mathsf{V}^*, \left(\mathsf{ct}_{\mathsf{V}^*}^{\mathrm{SFE}}, \mathsf{ct}_{|\psi\rangle}\right)\right) \quad .$$

Observe that if, in addition to sending an encryption for t last step, the verifier's circuit indeed performed an SFE evaluation of the circuit $\mathbf{CC}[\mathsf{Id}(\cdot), t, s]$, then the joint ciphertext $(\mathsf{ct}_s^{\mathrm{SFE}}, \mathsf{ct}_{|\phi\rangle})$ encrypts $(\mathsf{SFE.Enc}_{\mathsf{dk}}(s), |\phi\rangle)$, where $|\phi\rangle$ is the inner quantum state of V^* after the SFE evaluation.

iii. Sim computes $\mathsf{ct}_s \leftarrow \mathsf{QHE.Eval}_{\mathsf{pk}}\left(\mathsf{SFE.Dec}_{\mathsf{dk}}(\cdot), \mathsf{ct}_s^{\mathrm{SFE}}\right)$, and then computes $(\mathsf{sk}, \beta') = \widetilde{\mathbf{CC}}(\mathsf{ct}_s)$.Finally, Sim obtains the inner state of V^* by decryption: $|\phi\rangle \leftarrow \mathsf{QHE.Dec}_{\mathsf{sk}}(\mathsf{ct}_{|\phi\rangle})$.

3. **Sigma Protocol Messages Simulation:**

   (a) Sim executes $(\alpha, \gamma) \leftarrow \Sigma.\mathsf{S}(x, \beta')$ (the simulator from the special zero-knowledge property of the sigma protocol) and sends α to V^*.

   (b) V^* returns β.

4. **WI Proof by the Malicious Verifier:** Sim takes the role of the honest prover P in the WI proof V^* gives. If V^* fails to prove the statement, Sim executes $\mathsf{SimAbort}$.

5. **Simulation of the Prover's WI Proof and Information Reveal:** Sim gives V^* a WI proof using the witness that shows $\mathsf{cmt}_1, \mathsf{cmt}_2$ are both valid commitments (and that cmt_2 is a commitment to the SFE key dk used in step 2b). After the proof, Sim sends γ to V^*.

   Sim outputs x, the interaction transcript and the inner quantum state of V as the simulation's output.

$\mathsf{SimAbort}(\mathsf{V}^*, O_\rho)$:

1. **Estimation of Probability to Abort in Simulated Transcript:** Iterate and execute the following until either there are λ^2 aborts, or 2^λ iterations have passed: Sample $\rho \leftarrow O_\rho$ and execute $\mathsf{Sim}(\mathsf{V}^*, \rho)$ until the end of the WI proof inside step 5 of the simulation (i.e. do not perform the sending of γ), and if at some point the verifier either aborted or failed in its WI proof, count it as an abort.

   If you exitted the loop because you ran for 2^λ iterations without sufficient aborts, halt and output `Fail`. Otherwise, let N denote the number of iterations (N is a random variable), and set $a'(\rho) = \frac{\lambda^2}{N}$.

2. **Attempt to Get a Valid Abort Transcript:** Iterate until either you get an abort, or $\lambda \cdot \left\lceil \frac{1}{a'(\rho)} \right\rceil$ iterations have passed:

   - Sample $(x, \rho^{(x)}) \leftarrow O_\rho$ and set $\rho^{(x)}$ to be the inner state of V^*, and x be at the beginning of the output transcript.
   - Interact with V^* as the honest prover P until the end of step 5 of the original protocol, with exactly 3 differences:
     - The commitment cmt_1 is a commitment to $0^{|w|}$ rather than to w.
     - The message α at step 3a is generated by the first-message simulator $\alpha \leftarrow \Sigma.\mathsf{S}_1(x, 0^\lambda)$ (from claim 2.1), and not by the sigma protocol prover.

- At step 5, the witness used is for the first statement in the OR expression (that the commitments $\mathsf{cmt}_1, \mathsf{cmt}_2$ are valid and consistent), and not the second (that $x \in \mathcal{L}$).

If at some point during the interaction V^* either aborts or fails in its WI proof, count it as an abort, exit the loop and output V^*'s inner state (along with the instance x at the beginning).

If you exitted the loop because you ran for $\lambda \cdot \left\lceil \frac{1}{a'(\rho)} \right\rceil$ iterations without an abort, halt and output `Fail`.

We now turn to formally proving that the simulator runs in expected polynomial time and that the transcripts it generates are computationally indistinguishable (by quantum adversaries) from the transcripts generated by the malicious verifier's real interaction with P. In some parts of the proof, specifically, the proof that the simulator indeed runs in expected polynomial time (Claim 3.3) and also the proof that the simulator fails to simulate only with a negligible probability (Claim 3.6), we do an analysis similar to [GK96a], adapted to our setting.

Claim 3.3 (Simulator Runs in Expected Polynomial Time)**.** *There's a fixed polynomial $q(\cdot)$ such that for every verifier $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda \in \mathbb{N}}$, and arbitrary distribution $\rho = \{\rho_\lambda\}_{\lambda \in \mathbb{N}}$, the expected running time of the simulation $\mathsf{Sim}(\mathsf{V}^*_\lambda, O_{\rho_\lambda})$ is $q(|\mathsf{V}^*_\lambda|)$.*

Proof. Denote the following probabilities:

- $a(\rho)$: the probability that $\mathsf{V}^*(\rho^{(x)})$ (after sampling $(x, \rho^{(x)}) \leftarrow \rho$) aborts or fails in its WI proof (for simplicity, we refer to this scenario simply as "abort") in the **standard simulation**, that is, a single iteration of the process described in step 1 of the abort simulation.
- $b(\rho)$: (will be used in later claims) the probability that $\mathsf{V}^*(\rho^{(x)})$ aborts or fails in its WI proof (also here, we refer to this scenario as "abort") in the **abort simulation**, that is, a single iteration of the process described in step 2 of the abort simulation.
- $c(\rho)$: (will be used in later claims) the probability that $\mathsf{V}^*(\rho^{(x)})$ aborts or fails in its WI proof (also here, we refer to this scenario as "abort") in the **real interaction with the honest prover**.

Denote by $T(\mathsf{V}^*, \rho)$ the running time of the simulator on input V^*_λ with state ρ_λ, and by $T'(\mathsf{V}^*, \rho)$ the running time of the subroutine $\mathsf{SimAbort}$ on the same input, thus,

$$\mathbb{E}\left[T(\mathsf{V}^*, \rho)\right] = (1 - a(\rho)) \cdot \mathrm{poly}(\lambda) + a(\rho) \cdot \mathbb{E}[T'(\mathsf{V}^*, \rho)] \ ,$$

for some polynomial $\mathrm{poly}(\cdot)$ that denotes the running time of the simulator when it does not get aborts from V^*_λ (this polynomial depends on the circuit size of V^*_λ, which is also a polynomial in λ).

Recall that the first step of $\mathsf{SimAbort}$ is to get λ^2 aborts in the standard simulation (or score 2^λ iterations), thus the expected number of iterations in this step is $\frac{\lambda^2}{a(\rho)}$, and expected running time is $\frac{\lambda^2}{a(\rho)} \cdot \mathrm{poly}(\lambda)$ (note that if our control path has made it to $\mathsf{SimAbort}$, then $a(\rho) > 0$).

Observe that the expected running time of the second part of $\mathsf{SimAbort}$ is bounded by

$$\Pr\left[a'(\rho) \in \left[\frac{1}{2} \cdot a(\rho),\ \frac{3}{2} \cdot a(\rho)\right]\right] \cdot \frac{3\lambda}{a(\rho)} \cdot \mathrm{poly}(\lambda) + \Pr\left[a'(\rho) \notin \left[\frac{1}{2} \cdot a(\rho),\ \frac{3}{2} \cdot a(\rho)\right]\right] \cdot 2^{O(\lambda)} \ ,$$

thus it will be enough to show that the probability $\Pr\left[a'(\rho) \notin \left[\frac{1}{2} \cdot a(\rho),\ \frac{3}{2} \cdot a(\rho)\right]\right]$ is sufficiently small[6]. We have,

$$\Pr\left[a'(\rho) \notin \left[\frac{1}{2} \cdot a(\rho),\ \frac{3}{2} \cdot a(\rho)\right]\right] = \Pr\left[a'(\rho) < \frac{1}{2} \cdot a(\rho)\right] + \Pr\left[a'(\rho) > \frac{3}{2} \cdot a(\rho)\right] \ ,$$

[6] For the current claim, it will in fact be enough to only show that $\Pr\left[a'(\rho) < \frac{1}{2} \cdot a(\rho)\right]$ is sufficiently small, but we'll use the full case later so it will be beneficial to prove it now.

and we use Chernoff bounds:

$$\Pr\left[a'(\rho) < \frac{1}{2}\cdot a(\rho)\right] \underset{(a'(\rho)=\frac{\lambda^2}{N})}{=} \Pr\left[\frac{2\lambda^2}{a(\rho)} < N\right] \leq \Pr\left[X \leq \lambda^2\right] \underset{(*)}{\leq} e^{-\frac{\lambda^2}{4}} \quad ,$$

where $(*)$ follows from using the Chernoff tail bound $\Pr\left[X \leq (1-\delta)\mu\right] \leq e^{-\mu\cdot\delta^2\cdot\frac{1}{2}}$, with X being the number of aborts in the experiment of running the simulation $\frac{2\lambda^2}{a(\rho)}$ times, the expectation of X is $2\lambda^2$, and δ is $\frac{1}{2}$.

By similar reasoning it can be easily verified that,

$$\Pr\left[a'(\rho) > \frac{3}{2}\cdot a(\rho)\right] = \Pr\left[N < \frac{\frac{2}{3}\cdot\lambda^2}{a(\rho)}\right] \leq \Pr\left[X \geq (1+\frac{1}{2})\frac{2}{3}\cdot\lambda^2\right] \leq e^{-\frac{\lambda^2}{15}} \quad ,$$

using the Chernoff tail bound $\Pr\left[X \geq (1+\delta)\mu\right] \leq e^{-\frac{\delta^2}{2+\delta}\mu}$. □

We now start the proof of the simulation's validity which will be broken down into four parts:

1. We prove that the view in successful interactions (where the verifier does not abort or fail to prove its WI statement) is indistinguishable from the view in successful simulations, where there was no need to simulate an abort with $\mathsf{SimAbort}$.

2. We prove that the view in aborting interactions (where the verifier in fact does abort or fail to prove its WI statement) is indistinguishable from the view in abort simulations by $\mathsf{SimAbort}$.

3. We prove that the simulator's probability to fail in simulating the view (i.e. to output `Fail`) is negligible.

4. Finally, using the above three claims, we prove that the (entire) simulated view is computationally indistinguishable from the real view.

Claim 3.4 (Computational Indistinguishability of Non-Aborting Part)**.** *Let* $\mathsf{Sim}_\perp$ *be a simulation algorithm that acts like* Sim*, but if* V^* *aborts or fails in its WI proof, then it just outputs* $\perp$ *(instead of executing* $\mathsf{SimAbort}$*). Let* $\mathsf{OUT}_{\mathsf{V}^*}\langle\mathsf{P},\mathsf{V}^*\rangle_S$ *be the process of interaction between the honest prover and* V^**, s.t. if* V^* *aborts or fails in the WI proof, the process outputs* $\perp$*. Then, for every verifier* V^* *and instance-advice distribution* ρ*,*

$$\{\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle\mathsf{P}(w_\lambda),\mathsf{V}^*_\lambda(\rho^{(x)})\rangle_S(x)|(x,\rho^{(x)})\leftarrow\rho_\lambda\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{Sim}_\perp(\mathsf{V}^*_\lambda,\rho_\lambda)\}_{\lambda\in\mathbb{N}} \ .$$

Proof. First, we would like to note that the process $\mathsf{Sim}_\perp$ does not need an oracle access O_ρ but only one sample ρ, because it does not need to simulate aborts (which is the only part that needs the oracle access).

We prove the claim by a hybrid argument, specifically, we consider intermediate (hybrid) simulation processes, each consecutive pair can be shown to be computationally indistinguishable.

- $\mathsf{Sim}_\perp^{(1)}$: For an instance-advice distribution ρ, this process gets the witness $w \in \mathcal{R}_\mathcal{L}(x)$ and acts exactly like $\mathsf{Sim}_\perp$, with the exception that when it gives a WI proof in step 5 of the simulation, it uses the witness w for the second statement that says $x \in \mathcal{L}$.

- $\mathsf{Sim}_\perp^{(2)}$: This process acts like $\mathsf{Sim}_\perp^{(1)}$, with the exception that cmt_1 is a commitment to w rather than to $0^{|w|}$.

- $\mathsf{Sim}_\perp^{(3)}$: This process acts like $\mathsf{Sim}_\perp^{(2)}$, with the exception that if the verifier's message β from part 3b of the simulation does not match the extracted β' from part 2(b)iii of the simulation, the process halts and outputs $\perp$.

- $\mathsf{Sim}_\perp^{(4)}$: Acts like $\mathsf{Sim}_\perp^{(3)}$, with the exception that on parts 3a, 5, instead of computing α, γ through $\Sigma.\mathsf{S}$, we compute $\alpha \leftarrow \Sigma.\mathsf{P}(x, w)$ and $\gamma = \Sigma.\mathsf{P}(x, w; \alpha, \beta)$.

- $\mathsf{Sim}_\perp^{(5)}$: Acts like $\mathsf{Sim}_\perp^{(4)}$, with the exception that it does not perform the check described in $\mathsf{Sim}_\perp^{(3)}$.

- $\mathsf{Sim}_\perp^{(6)}$: Acts like $\mathsf{Sim}_\perp^{(5)}$, with the exception that cmt_2 is a commitment to $0^{|\mathsf{dk}|}$ instead of to dk.

- $\mathsf{Sim}_\perp^{(7)}$: Acts like $\mathsf{Sim}_\perp^{(6)}$, with the exception that instead of performing the extraction procedure described in step 2b of the simulation, if the verifier's message at step 2a was explainable and in particular ct_V is indeed a QHE encryption of some $t \in \{0,1\}^\lambda$ (which can be checked inefficiently), then at step 2b, the simulator sends $\mathsf{SFE}.\mathsf{Enc}_{\mathsf{dk}}(t)$, and otherwise acts like $\mathsf{Sim}_\perp^{(6)}$. The interaction continues regularly without extraction.

- $\mathsf{Sim}_\perp^{(8)}$: Acts like $\mathsf{Sim}_\perp^{(7)}$, with the exception that instead of performing the inefficient check on the verifier's message from step 2a (and then sending $\mathsf{SFE}.\mathsf{Enc}_{\mathsf{dk}}(t)$), the simulator just sends $\mathsf{SFE}.\mathsf{Enc}_{\mathsf{dk}}(0^\lambda)$ at step 2b.

- $\mathsf{Sim}_\perp^{(9)}$: Acts like $\mathsf{Sim}_\perp^{(8)}$, with the exception that cmt_2 is again a commitment to dk, rather than to $0^{|\mathsf{dk}|}$. Observe that this process is exactly $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}, \mathsf{V}^* \rangle_S$.

We now turn to explaining why each pair of consecutive distributions are computationally indistinguishable (and also why $\mathsf{Sim}_\perp \approx_c \mathsf{Sim}_\perp^{(1)}$). We note that by the transitivity of computational indistinguishability, this finishes our proof.

- $\mathsf{Sim}_\perp \approx_c \mathsf{Sim}_\perp^{(1)}$: Follows from the witness-indistinguishability property of the WI proof that the simulator gives in step 5.

- $\mathsf{Sim}_\perp^{(1)} \approx_c \mathsf{Sim}_\perp^{(2)}$: Follows from the hiding property of the non-interactive commitment cmt_1.

- $\mathsf{Sim}_\perp^{(2)} \approx_s \mathsf{Sim}_\perp^{(3)}$: These two distributions are in fact statistically indistinguishable. Note that the only difference between the distributions is the transcripts from $\mathsf{Sim}_\perp^{(2)}$ where $\beta \neq \beta'$, but were not erased by a failed WI proof. This means that either the verifier's transcript was not explainable (but still made it through the WI proof), or there was an error in the quantum homormorphic evaluation (because correctness of both SFE and CC obfuscation is perfect). Both scenarios happen only with a negligible probability, and thus the erasure of such cases creates only a negligible statistical distance between the distributions.

- $\mathsf{Sim}_\perp^{(3)} \approx_c \mathsf{Sim}_\perp^{(4)}$: Follows from the special zero-knowledge property of the sigma protocol.

- $\mathsf{Sim}_\perp^{(4)} \approx_s \mathsf{Sim}_\perp^{(5)}$: Also here there's more generally, a statistical indistinguishability, by the same reasoning that explains why distributions $\mathsf{Sim}_\perp^{(2)}$ and $\mathsf{Sim}_\perp^{(3)}$ are statistically indistinguishable.

- $\mathsf{Sim}_\perp^{(5)} \approx_c \mathsf{Sim}_\perp^{(6)}$: Follows from the hiding property of the non-interactive commitment cmt_2.

- $\mathsf{Sim}_\perp^{(6)} \approx_s \mathsf{Sim}_\perp^{(7)}$: Follows from the perfect correctness of the SFE and CC obfuscation, and the fact that the QHE evaluation is statistically correct *for every* pair $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{QHE}.\mathsf{Keygen}$.

- $\mathsf{Sim}_{\perp}^{(7)} \approx_c \mathsf{Sim}_{\perp}^{(8)}$: This indistinguishability follows from the input privacy property of the SFE encryption.
- $\mathsf{Sim}_{\perp}^{(8)} \approx_c \mathsf{Sim}_{\perp}^{(9)}$: Here we use the hiding property of the non-interactive commitment cmt_2.

□

Note that the above claim in particular implies that the probailities not to abort in both processes (or alternatively, probabilities to abort) are negligibly close.

Corollary 3.1 ($a(\rho)$ and $c(\rho)$ are Negligibly Close)**.** *Let $\rho = \{\rho_\lambda\}_{\lambda\in\mathbb{N}}$ be any sequence of instance-advice distributions, then there's a negligible function $\mu(\cdot)$ s.t. $\forall\lambda \in \mathbb{N} : |a(\rho_\lambda) - c(\rho_\lambda)| \leq \mu(\lambda)$.*

Claim 3.5 (Computational Indistinguishability of Aborting Part)**.** *Let $\mathsf{SimAbort}_{\perp}$ be an abort-simulation algorithm that acts exactly like executing only part 2 of $\mathsf{SimAbort}$, but running only one iteration of it, and if there's no abort (i.e. the verifier gets to the end of the prover's WI proof without any aborts, and also succeeded in its own WI proof), it outputs $\perp$. Let $\mathsf{OUT}_{\mathsf{V}^*}\langle\mathsf{P},\mathsf{V}^*\rangle_A$ be the process of interaction between the honest prover and V^*, s.t. if V^* finishes the interaction successfully until the end of the prover's WI proof, the process outputs $\perp$. Then, for every verifier V^* and instance-advice distribution ρ,*

$$\{\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle\mathsf{P}(w_\lambda), \mathsf{V}^*_\lambda(\rho^{(x)})\rangle_A(x) | (x, \rho^{(x)}) \leftarrow \rho_\lambda\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{SimAbort}_{\perp}(\mathsf{V}^*_\lambda, \rho_\lambda)\}_{\lambda\in\mathbb{N}} \ .$$

Proof. The proof will take a very similar route as the proof of Claim 3.4. We consider hybrid distributions, all of which will be computationally indistinguishable.

- $\mathsf{SimAbort}_{\perp}^{(1)}$: This process gets the witness $w \in \mathcal{R}_{\mathcal{L}}(x)$ (all next processes to come also get w) and acts exactly like $\mathsf{SimAbort}_{\perp}$, with the exception that when it gives a WI proof in step 5 of the protocol, it uses the witness w for the second statement that says $x \in \mathcal{L}$.
- $\mathsf{SimAbort}_{\perp}^{(2)}$: Acts exactly like $\mathsf{SimAbort}_{\perp}^{(1)}$, with the exception that cmt_1 is a commitment to w rather than to $0^{|w|}$.
- $\mathsf{SimAbort}_{\perp}^{(3)}$: Acts exactly like $\mathsf{SimAbort}_{\perp}^{(2)}$, with the exception that the message α at step 3a is generated by the sigma protocol $\alpha \leftarrow \Sigma.\mathsf{P}(x, w)$, and not by the first-message simulator $\alpha \leftarrow \Sigma.\mathsf{S}_1(x, 0^\lambda)$ (from claim 2.1). Note that this process is exactly $\mathsf{OUT}_{\mathsf{V}^*}\langle\mathsf{P},\mathsf{V}^*\rangle_A$.

It's left to reason about the indistinguishabilities between each pair.

- $\mathsf{SimAbort}_{\perp} \approx_c \mathsf{SimAbort}_{\perp}^{(1)}$: Follows from the witness-indistinguishability property of the WI proof that the simulator gives (as the prover) in step 5 of the protocol.
- $\mathsf{SimAbort}_{\perp}^{(1)} \approx_c \mathsf{SimAbort}_{\perp}^{(2)}$: Follows from the hiding property of the commitment cmt_1.
- $\mathsf{SimAbort}_{\perp}^{(2)} \approx_c \mathsf{SimAbort}_{\perp}^{(3)}$: Follows from Claim 2.1.

□

Note that the above claim in particular implies that the probailities to abort in both processes (or alternatively, probabilities not to abort) are negligibly close.

Corollary 3.2 ($b(\rho)$ and $c(\rho)$ are Negligibly Close)**.** *Let $\rho = \{\rho_\lambda\}_{\lambda\in\mathbb{N}}$ be any sequence of instance-advice distributions, then there's a negligible function $\mu(\cdot)$ s.t. $\forall\lambda \in \mathbb{N} : |b(\rho_\lambda) - c(\rho_\lambda)| \leq \mu(\lambda)$.*

We thus conclude that $a(\cdot), b(\cdot), c(\cdot)$ are all negligibly close to each other.

Claim 3.6 (The Simulation Almost Always Succeeds)**.** *Let* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda\in\mathbb{N}}$ *be a quantum verifier and let* $\rho = \{\rho_\lambda\}_{\lambda\in\mathbb{N}}$ *be an instance-advice distribution, then there's a negligible function* μ *s.t. for every* $\lambda \in \mathbb{N}$*:*

$$\Pr\left[\mathsf{Sim}(\mathsf{V}^*_\lambda, O_{\rho_\lambda}) = \mathtt{Fail}\right] \le \mu(\lambda) \enspace .$$

Proof. We have,

$$\Pr\left[\mathsf{Sim}(\mathsf{V}^*_\lambda, O_{\rho_\lambda}) = \mathtt{Fail}\right] =$$

$$a(\rho_\lambda) \cdot \left(\Pr\left[\mathsf{SimAbort}\text{ Failed in its first part}\right] + \Pr\left[\mathsf{SimAbort}\text{ Failed in its second part}\right]\right) \enspace , \qquad (1)$$

and we start with showing that the first probability is exponentially small.

$$\Pr\left[\mathsf{SimAbort}\text{ Failed in its first part}\right] = \sum_{i=0}^{\lambda^2-1} \Pr\left[\text{There were exactly } 2^\lambda - i \text{ fails out of } 2^\lambda \text{ tries}\right] \enspace . \qquad (2)$$

Now, recall that the probability to abort in a try (i.e. to succeed in a try) inside the first step of $\mathsf{SimAbort}$ is $a(\rho)$, and consider two cases.

- $a(\rho) \le 1/2$: Thus the sum from 2 is bounded by

  $$\lambda^2 \cdot \Pr\left[\text{There were } 2^\lambda \text{ fails}\right] = \lambda^2 \cdot (1 - a(x))^{2^\lambda} \enspace ,$$

  and by considering the multiplicative factor $a(\rho)$ (from 1) we can split to two cases (e.g. $a(\rho) \le 2^{-\lambda/2}$ and $a(\rho) > 2^{-\lambda/2}$) and show that (in both cases) when the above probability is multiplied by $a(\rho)$, it is bounded by $2^{-\Omega(\lambda)}$.

- $a(\rho) > 1/2$: thus the sum from 2 is bounded by

  $$\lambda^2 \cdot \Pr\left[\text{There were exactly } 2^\lambda - \lambda^2 \text{ fails}\right] = \lambda^2 \cdot \binom{2^\lambda}{2^\lambda - \lambda^2} \cdot (1 - a(\rho))^{2^\lambda - \lambda^2} \cdot a(\rho)^{\lambda^2} \le$$

  $$\lambda^2 \cdot 2^{\lambda^3} \cdot (1 - a(\rho))^{2^\lambda - \lambda^2} < \lambda^2 \cdot 2^{\lambda^3} \cdot 2^{-2^\lambda + \lambda^2} \le 2^{-\Omega(2^\lambda)} \enspace .$$

It is left to show that the probability to fail in the second step is also negligible, and we calculate.

$$\Pr\left[\mathsf{SimAbort}\text{ Failed in its second part}\right] \le \Pr\left[\text{There were } \frac{\lambda}{a'(\rho)} \text{ failed trials}\right] \le$$

$$\Pr\left[\text{There were } \frac{\lambda}{a'(\rho)} \text{ failed trials} \,\middle|\, a'(\rho) \in \left[\frac{1}{2}\cdot a(\rho), \frac{3}{2}\cdot a(\rho)\right]\right] + \Pr\left[a'(\rho) \notin \left[\frac{1}{2}\cdot a(\rho), \frac{3}{2}\cdot a(\rho)\right]\right] \enspace .$$

As a part of the proof of claim 3.3, we saw that $\Pr\left[a'(\rho) \notin \left[\frac{1}{2}\cdot a(\rho), \frac{3}{2}\cdot a(\rho)\right]\right] \le e^{-\Omega(\lambda^2)}$, so we only need to bound the first probability.

$$\Pr\left[\text{There were } \frac{\lambda}{a'(\rho)} \text{ failed trials} \,\middle|\, a'(\rho) \in \left[\frac{1}{2}\cdot a(\rho), \frac{3}{2}\cdot a(\rho)\right]\right] \le$$

$$\Pr\left[\text{There were } \frac{2\lambda}{3a(\rho)} \text{ failed trials}\right] = (1 - b(\rho))^{\frac{2\lambda}{3a(\rho)}} \le e^{-\frac{b(\rho)}{a(\rho)}\cdot\frac{2}{3}\cdot\lambda} \enspace .$$

Assume towards contradiction that there is some noticeable probability $\varepsilon(\lambda) = \lambda^{-O(1)}$ s.t. for infinitely many $\lambda \in \mathbb{N}$, we have,

$$\varepsilon(\lambda) \le a(\rho_\lambda) \cdot \Pr\left[\text{There were } \frac{\lambda}{a'(\rho_\lambda)} \text{ failed trials} \,\middle|\, a'(\rho_\lambda) \in \left[\frac{1}{2}\cdot a(\rho_\lambda), \frac{3}{2}\cdot a(\rho_\lambda)\right]\right] \enspace .$$

Thus,

$$\varepsilon(\lambda) \leq a(\rho_\lambda) \cdot e^{-\frac{b(\rho_\lambda)}{a(\rho_\lambda)} \cdot \frac{2}{3} \cdot \lambda} \; .$$

Now, we saw that $a(\rho)$ and $b(\rho)$ are negligibly close, and the last equation implies that $a(\rho)$ is a noticeable function (that is, $a(\rho_\lambda)$ is a noticeable function of $\lambda \in \mathbb{N}$). These two properties together imply that (for large values of λ) we have $\frac{b(\rho_\lambda)}{a(\rho_\lambda)} \in [\frac{1}{2}, 2]$, in contradiction to the inequality $\varepsilon(\lambda) \leq e^{-\frac{b(x)}{a(x)} \cdot \frac{2}{3} \cdot \lambda}$. □

We conclude with showing that the simulation is successful.

Lemma 3.2. *Let* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda \in \mathbb{N}}$ *be a quantum verifier and let* $\rho = \{\rho_\lambda\}_{\lambda \in \mathbb{N}}$ *be an instance-advice distribution, then,*

$$\{x, \mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w_\lambda), \mathsf{V}^*_\lambda(\rho^{(x)}_\lambda)\rangle(x) \mid (x, \rho^{(x)}_\lambda) \leftarrow \rho_\lambda\}_{\lambda \in \mathbb{N}} \approx_c \{\mathsf{Sim}(\mathsf{V}^*_\lambda, O_{\rho_\lambda})\}_{\lambda \in \mathbb{N}} \; .$$

Proof. Denote the following distributions.

- $S_{\mathsf{Sim}} = \{S_{\mathsf{Sim},\lambda}\}_{\lambda \in \mathbb{N}}$: A conditional distribution of $\mathsf{Sim}_\perp(\mathsf{V}^*, \rho)$, conditioned on that the output is not $\perp$ (might be an empty distribution, if $a(\rho) = 1$).

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

Now, consider the distribution that with probability $1 - a(\rho)$ samples from S_{Sim}, and with probability $a(\rho)$ samples from A_{Sim} - denote this distribution by $\mathsf{Sim}_{\text{No-Fail}}(\mathsf{V}^*, \rho)$ (we do not claim nothing about the efficiency of such process).

Note that by Claim 3.6, $\mathsf{Sim}_{\text{No-Fail}}(\mathsf{V}^*, \rho)$ is statistically indistinguishable from the output distribution of $\mathsf{Sim}(\mathsf{V}^*, O_\rho)$. This follows from three facts: (1) the (original) simulator fails only with negligible probability, (2) the verifier succeeds in the main simulation (and SimAbort isn't executed) with probability $1 - a(\rho)$, and then the output distribution is exactly S_{Sim}, (3) the verifier aborts or fails in its WI proof and we arrive to SimAbort with probability $a(\rho)$, and then if the simulation does not fail, the output distribution is exactly A_{Sim}. It thus remains to show that $\mathsf{Sim}_{\text{No-Fail}}$ is computationally indistinguishable from the distribution of the real interaction.

We define two more conditional distributions.

- $S_{\langle \mathsf{P},\mathsf{V}^*\rangle} = \{S_{\langle \mathsf{P},\mathsf{V}^*\rangle,\lambda}\}_{\lambda \in \mathbb{N}}$: A conditional distribution of $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}, \mathsf{V}^*\rangle_S$, conditioned on that the output is not $\perp$ (might be an empty distribution, if $c(\rho) = 1$).

- $A_{\langle \mathsf{P},\mathsf{V}^*\rangle} = \{A_{\langle \mathsf{P},\mathsf{V}^*\rangle,\lambda}\}_{\lambda \in \mathbb{N}}$: A conditional distribution of $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}, \mathsf{V}^*\rangle_A$, conditioned on that the output is not $\perp$ (might be an empty distribution, if $c(\rho) = 0$).

Similarly to the proofs of claims 3.4, 3.5, we define hybrid distributions, and show in steps why each consecutive pair of distributions are computationally indistinguishable.

- $\mathsf{Sim}^{(1)}_{\text{No-Fail}}$: Like $\mathsf{Sim}_{\text{No-Fail}}$, except that with probability $1 - a(\rho)$ it samples from $S_{\langle \mathsf{P},\mathsf{V}^*\rangle}$ instead from S_{Sim}.

- $\mathsf{Sim}^{(2)}_{\text{No-Fail}}$: Like $\mathsf{Sim}^{(1)}_{\text{No-Fail}}$, except that with probability $a(\rho)$ it samples from $A_{\langle \mathsf{P},\mathsf{V}^*\rangle}$ instead from A_{Sim}.

- $\mathsf{Sim}^{(3)}_{\text{No-Fail}}$: Like $\mathsf{Sim}^{(2)}_{\text{No-Fail}}$, except that it samples from $A_{\langle \mathsf{P},\mathsf{V}^*\rangle}$ with probability $c(\rho)$ instead of probability $a(\rho)$ (and thus also samples from $S_{\langle \mathsf{P},\mathsf{V}^*\rangle}$ with probability $1 - c(\rho)$ instead of $1 - a(\rho)$). Observe that this distribution is exactly $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}, \mathsf{V}^*(\rho^{(x)})\rangle(x)$, $(x, \rho^{(x)}) \leftarrow \rho$.

It is left to explain why each consecutive pair of distributions are computationally indistinguishable, and then by the transitivity of computational indistinguishability, our proof is finished.

- $\mathsf{Sim}_{\text{No-Fail}} \approx_c \mathsf{Sim}^{(1)}_{\text{No-Fail}}$: Assume towards contradiction there's an efficient quantum distinguisher D^* and a noticeable probability $\epsilon(\lambda)$ s.t. for an infinite subsequence $Q \subseteq \mathbb{N}$, for every $\lambda \in Q$, we have,
$$\left|\Pr[\mathsf{D}^*(\mathsf{Sim}_{\text{No-Fail}}(\mathsf{V}^*_\lambda, \rho_\lambda)) = 1] - \Pr[\mathsf{D}^*(\mathsf{Sim}^{(1)}_{\text{No-Fail}}(\mathsf{V}^*_\lambda, \rho_\lambda)) = 1]\right| \geq \epsilon(\lambda).$$
Consider two cases.
  - There's an infinite subsequence $Q' \subseteq Q$ and a negligible function μ s.t. $a(\rho_\lambda) \leq \mu(\lambda)$ for all $\lambda \in Q'$.
    In that case, the contradiction follows directly from Claim 3.4.
  - Otherwise, there's an infinite subsequence $Q' \subseteq Q$ where $a(\rho_\lambda)$ is noticeable for all indices inside Q'. This implies that for these indices we can sample from A_{Sim} in polynomial time and overwhelming probability of success in sampling. The last fact implies in turn that we can still use Claim 3.4 to get a contradiction in the following manner: We reduce distinguishability of non-aborting parts to distinguishability between $\mathsf{Sim}_{\text{No-Fail}}$ and $\mathsf{Sim}^{(1)}_{\text{No-Fail}}$, by getting a sample from either $\mathsf{Sim}_\perp$ or $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}, \mathsf{V}^* \rangle_S$, and if we got $\perp$, then we sample (with overwhelming probability of success) from A_{Sim}. If the sample was from $\mathsf{Sim}_\perp$, then the output sample of the reduction is from a distribution that's statistically indistinguishable from $\mathsf{Sim}_{\text{No-Fail}}$, and if the sample came from $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}, \mathsf{V}^* \rangle_S$, then the reduction output is sample from a distribution which is statistically indistinguishable from $\mathsf{Sim}^{(1)}_{\text{No-Fail}}$.

- $\mathsf{Sim}^{(1)}_{\text{No-Fail}} \approx_c \mathsf{Sim}^{(2)}_{\text{No-Fail}}$: Assume towards contradiction there's an efficient quantum distinguisher D^* and a noticeable probability $\epsilon(\lambda)$ s.t. for an infinite subsequence $Q \subseteq \mathbb{N}$, for every $\lambda \in Q$, we have,
$$\left|\Pr[\mathsf{D}^*(\mathsf{Sim}^{(1)}_{\text{No-Fail}}(\mathsf{V}^*_\lambda, \rho_\lambda)) = 1] - \Pr[\mathsf{D}^*(\mathsf{Sim}^{(2)}_{\text{No-Fail}}(\mathsf{V}^*_\lambda, \rho_\lambda)) = 1]\right| \geq \epsilon(\lambda).$$
Consider two cases.
  - There's an infinite subsequence $Q' \subseteq Q$ and a negligible function μ s.t. $1 - c(\rho_\lambda) \leq \mu(\lambda)$ for all $\lambda \in Q'$.
    In that case, the contradiction follows directly from Claim 3.5.
  - Otherwise, there's an infinite subsequence $Q' \subseteq Q$ where $1-c(\rho_\lambda)$ is noticeable for all indices inside Q'. This implies that for these indices we can sample from $S_{\langle \mathsf{P}, \mathsf{V}^* \rangle}$ in polynomial time and overwhelming probability of success in sampling. The last fact implies in turn that we can still use Claim 3.5 to get a contradiction in the following manner: We reduce distinguishability of aborting parts to distinguishability between $\mathsf{Sim}^{(1)}_{\text{No-Fail}}$ and $\mathsf{Sim}^{(2)}_{\text{No-Fail}}$, by getting a sample from either $\mathsf{SimAbort}_\perp$ or $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}, \mathsf{V}^* \rangle_A$, and if we got $\perp$, then we sample (with overwhelming probability of success) from $S_{\langle \mathsf{P}, \mathsf{V}^* \rangle}$. If the sample was from $\mathsf{SimAbort}_\perp$, then the output sample of the reduction is from a distribution that's statistically indistinguishable from $\mathsf{Sim}^{(1)}_{\text{No-Fail}}$, and if the sample came from $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}, \mathsf{V}^* \rangle_A$, then the reduction output is sample from a distribution which is statistically indistinguishable from $\mathsf{Sim}^{(2)}_{\text{No-Fail}}$.

- $\mathsf{Sim}^{(2)}_{\text{No-Fail}} \approx_s \mathsf{Sim}^{(3)}_{\text{No-Fail}}$: These two distributions are also statistically indistinguishable, which follows from the fact that $a(\rho)$ and $c(\rho)$ are negligibly close (Corollary 3.1).

□

4 Constant-Round Zero-Knowledge Quantum Arguments for QMA

In this section, we show how to use our constant-round ZK argument in order to construct constant-round ZK quantum arguments for QMA, that is, honest parties are efficient and quantum (prover is efficient given a quantum witness) and quantum, and communication is also quantum. In [BJSW16], Broadbent, Ji, Song and Watrous construct a constant-round zero-knowledge quantum protocol for QMA with constant soundness. They then perform a polynomial sequential repetition of that protocol to get a proof system (with negligible soundness) with polynomially many rounds.

The authors also note that a parallel repetition of their (core, constant-round) protocol can be used to obtain a constant-round zero-knowledge quantum protocol for QMA, given two tools:

- A fully-simulatable constant-round coin-flipping protocol for polynomially-many bits.
- A constant-round zero-knowledge protocol for NP.

In section 3 we constructed a constant-round argument for NP and in this section (using our zero-knowledge protocol) we construct the required coin-flipping protocol.

Next, we formally define a zero-knowledge quantum argument system for QMA, and then state our result regarding QMA.

Definition 4.1 (Zero-Knowledge Quantum Argument for QMA)**.** *A quantum protocol* (P, V) *with an honest QPT prover* P *and an honest QPT verifier* V *for a language* $\mathcal{L} \in$ ***QMA*** *is a quantum-secure zero-knowledge argument if it satisfies:*

1. **Statistical Completeness:** *There is some negligible function* $\mu(\cdot)$ *s.t. for any* $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0,1\}^\lambda, w \in \mathcal{R}_\mathcal{L}(x)$[7]*,*

$$\Pr[\mathsf{OUT}_\mathsf{V}\langle \mathsf{P}(w), \mathsf{V}\rangle(x) = 1] \geq 1 - \mu(\lambda) \enspace .$$

2. **Quantum Computational Soundness:** *For any non-uniform quantum polynomial-size prover* $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_{\lambda \in \mathbb{N}}$*, there exists a negligible function* $\mu(\cdot)$ *such that for any security parameter* $\lambda \in \mathbb{N}$ *and any* $x \in \{0,1\}^\lambda \setminus \mathcal{L}$*,*

$$\Pr\left[\mathsf{OUT}_\mathsf{V}\langle \mathsf{P}^*_\lambda, \mathsf{V}\rangle(x) = 1\right] \leq \mu(\lambda) \enspace .$$

3. **Quantum Computational Zero-Knowledge:** *There exists a quantum expected polynomial-time simulator* Sim*, such that for any non-uniform quantum polynomial-size verifier* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda \in \mathbb{N}}$*,*

$$\{\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda\rangle(x)\}_{\substack{\lambda\in\mathbb{N},\\ x\in\mathcal{L}\cap\{0,1\}^\lambda,\\ w\in\mathcal{R}_\mathcal{L}(x)}} \approx_c \{\mathsf{Sim}(x, \mathsf{V}^*_\lambda)\}_{\substack{\lambda\in\mathbb{N},\\ x\in\mathcal{L}\cap\{0,1\}^\lambda,\\ w\in\mathcal{R}_\mathcal{L}(x)}} \enspace .$$

We note that like in the case of NP, the stronger (quantumly-semi-composable) version of the zero-knowledge definition is proven by practically the same proof, but for the sake of simplicity (and because we don't actually use the ZK argument for QMA in this work, only construct it), we stay with the simpler definition.

We now turn to the definition of constant-round fully-simulatable coin-flipping protocol, where the simulation is in the same quantum-semi-composable sense as the zero-knowledge guarantee for the argument from section 3.

[7] For a language $\mathcal{L}$ in QMA, for an instance $x \in \mathcal{L}$ in the language, the set $\mathcal{R}_\mathcal{L}(x)$ is the (possibly infinite) set of quantum witnesses that make the BQP verification machine accept with some probability negligibly close to 1.

Definition 4.2 (2-Party Coin-Flipping Protocol with QSC Full Simulation of Party A)**.** *A 2-party coin-flipping protocol (with full simulation of party* A*) is given by two classical algorithms* (A, B) *with joint input* $1^\lambda, 1^{k(\lambda)}$*, and joint output* $r \in \{0,1\}^k \cup \{\perp\}$ *(which can be efficiently computed from the protocol transcript), with the following randomness guarantee:*

- **Computational Randomness Guarantee for** A**:** *Let* $\mathsf{B}^* = \{\mathsf{B}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ *be a polynomial-size quantum adversary. Let* p_λ *be the probability that the protocol ends with a non-*$\perp$ *output when* A *interacts with* B^*_λ*, and let* (p, U_k) *be the distribution that with probability* $1 - p$ *outputs* $\perp$ *and with probability* p *outputs a sample from* U_k*. Then, the protocol output* r *is computationally indistinguishable from* $\{(p_\lambda, U_{k(\lambda)})\}_{\lambda \in \mathbb{N}}$ *(where* $k(\lambda)$ *is a polynomial).*

- **Strong Randomness Guarantee for** B **(Full QSC Simulation of Party** A**):** *There exists a quantum expected polynomial-time simulator* Sim*, such that for any quantum polynomial-size adversary* $\mathsf{A}^* = \{\mathsf{A}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$*,*

$$\{\mathsf{VIEW}_{\mathsf{A}^*_\lambda}\langle \mathsf{A}^*(\rho_\lambda), \mathsf{B}\rangle(1^\lambda, 1^{k(\lambda)})\}_{\lambda \in \mathbb{N}} \approx_c \{\mathsf{Sim}(r, \mathsf{A}^*_\lambda, O_{\rho_\lambda}) \mid r \leftarrow U_{k(\lambda)}\}_{\lambda \in \mathbb{N}} \ ,$$

  where,

  - $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle$ *includes (1) the output* $r \in \{0,1\}^k \cup \{\perp\}$ *of the protocol, (2) the protocol transcript, and (3) the inner quantum state of* A^* *at the end of interaction (whether it ended successfully or not).*
  - *For the "protocol output" part of its simulation output,* $\mathsf{Sim}(r, \mathsf{A}^*_\lambda, O_{\rho_\lambda})$ *always outputs either* r *or* $\perp$*.*
  - O_ρ *is an inputless oracle that upon activation returns a copy of the quantum state* ρ*.*

We now can state our result regarding QMA.

Theorem 4.1. *Given the existence of constant-round QSC zero-knowledge arguments for NP, and constant-round coin-flipping protocols with QSC Full Simulation, there exists constant-round zero-knowledge quantum arguments for QMA.*

The construction and proof of the QMA protocol appear in [BJSW16] and are omitted here. The only change in the proof from [BJSW16] is that our coin-flipping and zero-knowledge NP protocols simulations are quantumly-semi-composable, rather then fully composable. This only makes the proof less modular. In our fully-simulatable coin-flipping protocol we demonstrate how to use the semi-composable definition.

We move to present the fully-simulatable coin-flipping protocol.

Ingredients and notation:

- A non-interactive commitment scheme Com.

- A 2-message function-hiding secure function evaluation scheme $(\mathsf{SFE.Gen}, \mathsf{SFE.Enc}, \mathsf{SFE.Eval}, \mathsf{SFE.Dec})$.

- A constant-round QSC zero-knowledge argument system $(\mathsf{P}_{\mathrm{NP}}, \mathsf{V}_{\mathrm{NP}})$ for NP.

Handling Aborts and Alikes. Like in previous sections, we set a convention to handle publicly checkable misbehaviors by the parties in the protocol.

- For a security parameter λ, for each message in the protocol, it is known (publicly) based on λ, what is the length of each message (or upper and lower bounds on that length). If a party sends a message in an incorrect length, the receiving party fixes it locally and trivially; if the message is too long, it cuts the message in a suitable place, and if it's too short then pads with zeros.

- In the below protocol, whenever a party aborts, the other party ends communication, and the output of the protocol is $\perp$. The output of the protocol is $\perp$ also if either of the parties fail to prove their arguments in steps 2 and 5 below.

We describe the protocol in Figure 2.

Protocol 2

Common Input: A security parameter $\lambda \in \mathbb{N}$ and a number $k \in \mathbb{N}$ of wanted random coins, both given in unary $1^\lambda, 1^k$.

1. **Initial Commitment by B:** B sends a commitment to 0: $\mathsf{cmt}_\mathsf{B} \leftarrow \mathsf{Com}(1^\lambda, 0)$.
2. **ZK Argument by B:** B interacts with A through $(\mathsf{P}_\mathsf{NP}, \mathsf{V}_\mathsf{NP})$ to give a ZK argument that cmt_B is indeed a commitment to 0, that is, there exists randomness $r_0 \in \{0,1\}^{\mathrm{poly}(\lambda,1)}$[a] s.t. $\mathsf{cmt}_\mathsf{B} = \mathsf{Com}(1^\lambda, 0; r_0)$.
3. A **Commits to Coins:** A chooses k random coins $a \leftarrow U_k$ and sends a commitment for them $\mathsf{cmt}_\mathsf{A} \leftarrow \mathsf{Com}(1^\lambda, a)$.
4. A **Challenges** B**:** The parties interact so that A can offer to send a if B managed to trick A in the ZK argument.
   (a) B computes $\mathsf{dk} \leftarrow \mathsf{SFE.Gen}(1^\lambda)$ and sends $\mathsf{ct}_\mathsf{B} \leftarrow \mathsf{SFE.Enc}_\mathsf{dk}(0^{\mathrm{poly}(\lambda,1)})$.
   (b) A sends $\hat{\mathsf{ct}} \leftarrow \mathsf{SFE.Eval}\left(C_{1\to a}, \mathsf{ct}_\mathsf{B}\right)$, where $C_{1\to a}$ is the (canonical) circuit that for input $r_1 \in \{0,1\}^{\mathrm{poly}(\lambda,1)}$ s.t. $\mathsf{cmt}_\mathsf{B} = \mathsf{Com}(1^\lambda, 1; r_1)$, outputs a, and for any other input outputs $\perp$.
5. **ZK Argument by A:** A interacts with B through $(\mathsf{P}_\mathsf{NP}, \mathsf{V}_\mathsf{NP})$ to give a ZK argument for the statement that its transcript until the end of step 4b is explainable.
6. **Both Parties Reveal Coins:** B computes $b \leftarrow U_k$ and sends it to A. A responds with sending a along with an opening of the commitment cmt_A. If the opening does not open cmt_A to a, protocol output is $\perp$.
7. **Output:** The output of the protocol is $r = a \oplus b$.

[a] Let $\mathrm{poly}(\lambda, \ell)$ denote the polynomial that represents the amount of randomness the commitment algorithm $\mathsf{Com}(\cdot)$ needs for security parameter λ and message length ℓ.

Figure 2: A constant-round coin-flipping protocol with full simulation of party A and quantum security.

4.1 Randomness Guarantee for A

We prove the (computational) randomness guarantee for party A, which basically says that for any efficient quantum attacker B, the output distribution of the protocol is computationally indistinguishable from one where the protocol output is a truly random string (considering the aborting probability $1-p$ of the attacker).

Claim 4.1. *Protocol 2 has a computational randomness guarantee for* A.

Proof. Let $\mathsf{B}^* = \{\mathsf{B}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ be a quantum polynomial-size adversary (and let $k(\lambda)$ be a polynomial that denotes the number of output bits in the protocol, for security parameter $\lambda \in \mathbb{N}$), and denote by r_λ the random variable in $\{0,1\}^{k(\lambda)} \cup \{\perp\}$ that is the output of the interaction between A and $\mathsf{B}^*(\rho_\lambda)$, and by p_λ the probability the protocol ends with a non-$\perp$ output when A interacts with $\mathsf{B}^*_\lambda(\rho_\lambda)$. We prove that

$$\{r_\lambda\}_{\lambda \in \mathbb{N}} \approx_c \{(p_\lambda, U_{k(\lambda)})\}_{\lambda \in \mathbb{N}} \ .$$

We can assume w.l.o.g. that $\mathsf{B}^*(\rho)$ always sends the same first message, and the reason is as follows: Consider the distribution D over the first message $\mathsf{cmt}_{\mathsf{B}^*}$ of $\mathsf{B}^*(\rho)$, and for a message $\mathsf{cmt}_{\mathsf{B}^*}$ denote by $q_{\mathsf{cmt}_{\mathsf{B}^*}}$ the probability that the protocol ends with a non-$\perp$ output when A interacts with B^*, when the first message of B^* was $\mathsf{cmt}_{\mathsf{B}^*}$. The reasoning is finished by the following steps:

- Recall that $\mathbb{E}_{\mathsf{cmt}_{\mathsf{B}^*} \leftarrow D}\left[q_{\mathsf{cmt}_{\mathsf{B}^*}}\right] = p$.
- Furthermore, note that the following distribution is identical to the distribution (p, U_k): Sample $\mathsf{cmt}_{\mathsf{B}^*} \leftarrow D$ and output a sample from $(q_{\mathsf{cmt}_{\mathsf{B}^*}}, U_k)$.
- Denote by D^* a distinguisher that tells the difference between the protocol's output and (p, U_k). Thus,

$$|\Pr[\mathsf{D}^*(r) = 1] - \Pr[\mathsf{D}^*(p, U_k) = 1]| =$$
$$\left|\mathbb{E}_{\mathsf{cmt}_{\mathsf{B}^*} \leftarrow D}\left[\Pr\left[\mathsf{D}^*(r_{\mathsf{cmt}_{\mathsf{B}^*}}) = 1\right] - \Pr\left[\mathsf{D}^*(q_{\mathsf{cmt}_{\mathsf{B}^*}}, U_k) = 1\right]\right]\right| \ ,$$

  where $r_{\mathsf{cmt}_{\mathsf{B}^*}}$ is the random variable that denotes the protocol's output conditioned that the first message of B^* is $\mathsf{cmt}_{\mathsf{B}^*}$.
- Finally, we can consider the sample $\mathsf{cmt}'_{\mathsf{B}^*}$ and inner quantum state $\rho^{(1)}$ s.t. the gap inside the expectation is maximized, and assume that B^* always uses this pair (as B^* can use n.u. quantum advice).

We furthermore add w.l.o.g. the assumption that the first message of B^*, $\mathsf{cmt}_{\mathsf{B}^*}$, is not only deterministic, but indeed is a commitment to 0. If it wasn't the case that $\mathsf{cmt}_{\mathsf{B}^*}$ commits to 0, then by the soundness of the argument that B^* gives at step 2 of the protocol, the argument will fail to convince A with overwhelming probability and protocol output r is $\perp$, this means that p is negligibly close to 1 and that the distributions r and (p, U_k) are statistically indistinguishable, and our proof is finished.

Now, note that the distribution $\{r_\lambda\}_\lambda$ is exactly the distribution that's generated by the following process: A and $\mathsf{B}^*(\rho)$ interact regularly (as usual, if there is an abort by B^* or that it fails in its argument in step 2, output is $\perp$), and once B^* sends b at step 6, the process output is $a \oplus b$.

We claim that the last process is computationally indistinguishable from the following: $\mathsf{A}(0^k)$[8] and $\mathsf{B}^*(\rho)$ interact regularly, and when B^* sends its string b, the process chooses at random $a \leftarrow U_k$ and outputs $a \oplus b$.

In a nutshell, the reason that the last two distributions are computationally indistinguishable is that the string a that A commits to in step 3 is hidden from computationally bounded quantum adversaries. More precisely, the last informal claim is captured formally in Claim 4.2. It's easy to verify that given Claim 4.2 and an averaging argument over a, the above indistinguishability is proved.

We note that the last distribution is exactly (p_0, U_k), where p_0 is the probability that $\mathsf{B}^*(\rho)$ aborts when interacting with $\mathsf{A}(0^k)$. Finally, observe that (p_0, U_k) is statistically indistinguishable from (p, U_k) (recall that p is the probability for $\mathsf{B}^*(\rho)$ to abort when interacting with A), and this follows from the fact that p and p_0 are negligibly close, which is also implied by Claim 4.2. □

[8] For a string $a' \in \{0,1\}^k$, $\mathsf{A}(a')$ denotes the process A s.t. the string a that it commits to in step 3 is a'.

Claim 4.2 (A's coins are hidden)**.** *For a string $s \in \{0,1\}^k$, let $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s), \mathsf{B}^*(\rho)\rangle$ be the distribution over the transcript and inner quantum state of* B* *at the end of step 5, generated in the process of interaction between $\mathsf{A}(s)$ and $\mathsf{B}^*(\rho)$. Then, for every two $k(\lambda)$-length strings $\{s_{0,\lambda}\}_{\lambda\in\mathbb{N}}$, $\{s_{1,\lambda}\}_{\lambda\in\mathbb{N}}$:*

$$\{\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_{0,\lambda}), \mathsf{B}^*_\lambda(\rho_\lambda)\rangle(1^\lambda, 1^{k(\lambda)})\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_{1,\lambda}), \mathsf{B}^*_\lambda(\rho_\lambda)\rangle(1^\lambda, 1^{k(\lambda)})\}_{\lambda\in\mathbb{N}} \ .$$

Proof. Define the following hybrid distributions on transcripts.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$: Acts like the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle$, except that in step 5, instead of A communicating with B^* to give a ZK argument, we take the ZK simulator $\mathsf{Sim}_{\mathsf{NP}}$ and use it to simulate the argument by A. The instance-advice distribution $(x, \rho^{(x)}) \leftarrow \sigma$ that the simulator will have oracle access to, is the conditional distribution on protocol transcript (x) and inner state of B^* that is generated by the interaction between $\mathsf{A}(s_0)$ and $\mathsf{B}^*(\rho)$ until the end of step 4b ($\rho^{(x)}$), **conditioned** that the transcript is successful so far (no aborts or argument fails from B^*).

  We further describe the operation of the process: It has oracle access O_ρ to the quantum advice ρ of B^*, and if it gets to step 5 and needs to simulate, it will use the generated transcript as instance and inner state of B^* as the advice, and every time the simulator needs another sample, the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$ restarts, samples from O_ρ and generates a transcript and inner state of B^* such that the transcript is successful - this might take many tries, but the process will keep on trying until it gets a successful transcript (if we managed to get to step 5 once, the probability to sample a transcript that's successful until that step, is non-zero).

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)}$: Acts like the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$, except that in step 4b, instead of actually performing an SFE evaluation of $C_{1\to s_0}$, the process performs an SFE evaluation of the circuit $C_\perp$ that always outputs $\perp$.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(3)}$: Acts like the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)}$, except that in step 3, instead of committing to s_0, A commits to s_1.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(4)}$: Acts like the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(3)}$, except that in step 4b, the process performs an SFE evaluation of the circuit $C_{1\to s_1}$, and not of the circuit $C_\perp$.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(5)}$: Acts like the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(4)}$, except that instead of using the ZK simulator for A's argument, the process is a regular communication between $\mathsf{A}(s_1)$ and B^*. Observe that this process is exactly $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_1), \mathsf{B}^*(\rho)\rangle$.

We now explain why $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle$ is computaionally indistinguishable from $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$, and why each consecutive pair of distributions are computationally indistinguishable. By the transitivity of computational indistinguisability, after we show the above, we are done.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle \approx_c \mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$: Follows from the quantumly-semi-composable (QSC) ZK property of definition 3.3.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)} \approx_c \mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)}$: As a short explanation, this indistinguishability follows from performing a cutoff on the ZK simulator's running time (which is expected polynomial time), together with the circuit privacy property of the SFE encryption (that's more easily captured by Claim 2.2). As a longer explanation, we give the full reasoning which will be used (but not written from scratch every time) in every of the following indistinguishabilities.

  Assume towards contradiction that these distributions are distinguishable, and we describe a process for distinguishing between SFE evaluations of $C_{1\to s_0}$ and SFE evaluations of $C_\perp$ which according to Claim 2.2 are statistically indistinguishable because the two circuits have identical functionality (recall we know that the commitment of B^* is to 0, and because of its perfect binding, there is no opening of the commitment to 1, and thus the circuit $C_{1\to s_0}$ always outputs $\perp$).

– **Identifying the Processes Operation Exactly:** For clarity, we start with explicitly describing the operation processes of both $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$, $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)}$. Any of the above processes start regularly, and if a process does not get to step 5 on the first try and B^* aborts somewhere along the way (or fails the ZK argument), the process terminates regularly. However, if a process does make it to the simulation at step 5, then every time the ZK simulator needs a fresh sample, the process "restarts": it erases the transcript so far, samples $\rho \leftarrow O_\rho$ and tries to sample a new successful transcript - this might take a while, because B^* can abort somewhere between the restarting point of the process and restarting point of simulation.

– **Identifying Processes Expected Running Time as Polynomial:** We would now like to explain why both processes still run in expected polynomial time, even when the probability of B^* to succeed is negligible. Let p_1 be the probability that B^* succeeds to get to the simulation part in the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$, from the point that we sample $\rho \leftarrow O_\rho$ (and let p_2 be the same probability for the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)}$), then the average number of tries until getting a successful transcript in the setting of $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$ is $1/p_1$, which might be exponential. The catch is that we perform such repeated sampling only when we already gotten a successful transcript once, which happens with probability p_1. More specifically, the expected running time of the procedure $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$ is,

$$(1-p_1)\cdot \mathrm{poly}(\lambda) + p_1 \cdot q(|\mathsf{B}^*|)\cdot \frac{1}{p_1}\cdot \mathrm{poly}(\lambda) \le \mathrm{poly}(\lambda) + q(|\mathsf{B}^*|)\cdot \mathrm{poly}(\lambda) =: q'(\lambda) \ ,$$

where $\mathrm{poly}(\lambda)$ is a polynomial that denotes the running time of the protocol interaction (executing the algorithms A and B^* until the simulation part), and the number $\frac{1}{p_1}\cdot \mathrm{poly}(\lambda)$ thus denotes the expected running time to get a fresh sample for the ZK simulator, which runs in expected time $q(|\mathsf{B}^*|)$. By similar reasoning, we obtain that the expected running time of the process $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)}$ is also bounded by the polynomial $q'(\lambda)$.

– **Using the Distinguisher's Advantage to Define Strinct Polynomial-Time Processes:** A distinguisher D^* that tells the difference between the two processes above, distinguishes with some noticeable probability $\epsilon(\lambda)$ (for infinitely many security parameters). We will show how to turn the simulators to fixed polynomial-time ones that use a polynomial-size quantum advice. Now, in either of the processes (whether we get an evaluation of $C_{1\to s_0}$ or of $C_\perp$), the expectation of the running time of the process is $q'(\lambda)$, where $q'(\cdot)$ is fixed polynomial. By Markov's inequality, the probability that the process's running time exceeds $m(\lambda) := q'(\lambda)\cdot 4\cdot \frac{1}{\epsilon}$ is bounded by $\frac{\epsilon}{4}$, and thus if we consider an alternative process that halts after exceeding this time bound (and in particular doesn't use oracle access O_ρ to ρ samples, but a fixed $m(\lambda)$ samples of ρ), its output distribution has statistical distance bounded by $\frac{\epsilon}{4}$ to that of the original process.

– **Observing that the Distinguisher Tells the Difference Between the Cutoff Processes:** For the distinguisher D^* that tells the difference between the two original processes (that take expected polynomial time to execute), the distinction advantage is at least ϵ, and thus, because each of the cutoff processes is $\frac{\epsilon}{4}$-close to its original version, the distinction advantage of D^* between the two cutoff processes is at least $\frac{\epsilon}{2}$.

– **Connecting the Processes with a Cryptographic Security Guarantee:** We now connect the distinguisher with the circuit privacy property of the SFE. Note that by Claim 2.2 and by the fact that the circuits $C_{1\to s_0}$ and of $C_\perp$ have identical functionality, we can claim that for any distribution D on $(|\phi\rangle, \mathsf{ct}^*)$ where $|\phi\rangle$ is a quantum state and ct^* is a possibly invalid ciphertext, the distributions $(|\phi\rangle, \mathsf{ct}^*, \mathsf{SFE.Eval}(C_{1\to s_0}, \mathsf{ct}^*))$ and $(|\phi\rangle, \mathsf{ct}^*, \mathsf{SFE.Eval}(C_\perp, \mathsf{ct}^*))$ are also statistically indistinguishable, this is true by a simple averaging argument. The above implies that for any polynomial number of independent samples of the abovementioned tuples, evaluations of $C_{1\to s_0}$ on samples from D and evaluations of $C_\perp$ on samples from D are still statistically indistinguishable, this can be verified by a hybrid argument.

In particular, consider the distribution D that samples a transcript and inner quantum state for B^* generated at the end of step 4a, that is, $|\phi\rangle$ is the transcript and the inner quantum state of B^*, not including the message $\mathsf{ct}_{\mathsf{B}^*}$ from step 4a, and $\mathsf{ct}^* := \mathsf{ct}_{\mathsf{B}^*}$, and if there was an abort before step 4a in the interaction of that sample, define $\mathsf{ct}^* := 0$. $m(\lambda)$ is a polynomial and such number of samples is statistically indistinguishable - we show that we can distinguish, in contradiction.

- **Security Reduction:** We describe a distinguisher $\tilde{\mathsf{D}}^*$ that will use D^*: it gets a sample $(z_1, z_2, \cdots, z_{m(\lambda)})$ which is either $m(\lambda)$ tuples of evaluations of $C_{1\to s_0}$ or of $C_\perp$. If the transcript in z_1 is aborting we just give D^* the transcript and inner state of B^* (this corresponds to the scenario where B^* aborted before we even got to the ZK simulation for the first time). Otherwise, we initiate the ZK simulator and when it needs new sample from its instance-advice oracle, then we use the rest of the samples $z_2, z_3, \cdots, z_{m(\lambda)}$ to feed it. It can be verified that when the tuples come from evaluations of $C_{1\to s_0}$ then the transcripts that $\tilde{\mathsf{D}}^*$ generates for D^* come from the output distribution of the cutoff process of $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)}$, and when tuples come from evaluations of $C_\perp$ then the transcripts that $\tilde{\mathsf{D}}^*$ generates for D^* come from the output distribution of the cutoff process of $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)}$. Thus D^* tells the difference with advantage $\frac{\epsilon}{2}$ and so does $\tilde{\mathsf{D}}^*$, in contradiction.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)} \approx_c \mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(3)}$: Follows from performing a cutoff on the ZK simulator's running time, together with the hiding of the commitment $\mathsf{Com}(\cdot)$.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(3)} \approx_c \mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(4)}$: Similarly to the indistinguishability $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(1)} \approx_c \mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(2)}$, this one follows from performing a cutoff on the ZK simulator's running time, together with the circuit privacy property of the SFE encryption.

- $\mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(4)} \approx_c \mathsf{VIEW}_{\mathsf{B}^*}\langle \mathsf{A}(s_0), \mathsf{B}^*(\rho)\rangle^{(5)}$: Follows again from the QSC ZK property of definition 3.3.

□

4.2 Full Simulation of Party A

We show a quantum expected polynomial-time simulator Sim s.t. for any polynomial-size quantum adversary $\mathsf{A}^* = \{\mathsf{A}^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, simulates the protocol as described in definition 4.2.

Handling Aborts in Simulation. In the below simulation, if at any part of the simulation A^* either aborts, or fails in its argument statement (for messages of incorrect length, the simulator acts like B and trivially corrects the length, as describe above the protocol) the simulator halts, outputs $\perp$ as the simulation coin-flipping output, the transcript so far as the simulation transcript output, and the inner state of A^* as the simulation adversary state output.

$\mathsf{Sim}(r, \mathsf{A}^*, O_\rho)$:

1. **Simulation of Initial Commitment:** Sim samples $\rho \leftarrow O_\rho$ and sets ρ to be the inner state of A^*. Sim then sends to A^* a commitment to 1: $\mathsf{cmt}_{\mathsf{Sim}} = \mathsf{Com}(1^\lambda, 1; r_1)$, where $r_1 \in \{0,1\}^{\mathrm{poly}(\lambda,1)}$ is the random string used as the randomness of the commitment algorithm.

2. B**'s ZK Argument Simulation:** Sim uses the argument's ZK simulator $\mathsf{Sim}_{\mathrm{NP}}$ to simulate for A^* the communication with B. It executes $\mathsf{Sim}_{\mathrm{NP}}(\mathsf{A}^*, O_\sigma)$, where σ is the instance-advice distribution $(x, \rho^{(x)})$ that is generated s.t. x is a commitment to 1 by Sim (from step 1 of the simulation), and $\rho_{(x)}$ is the inner state ρ of A^* at the beginning of the protocol.

3. **Extraction Attempt from** A^***:**

   - A^* outputs $\mathsf{cmt}_{\mathsf{A}^*}$.
   - Sim computes $\mathsf{dk} \leftarrow \mathsf{SFE.Gen}(1^\lambda)$ and sends $\mathsf{ct}_{\mathsf{Sim}} \leftarrow \mathsf{SFE.Enc}_{\mathsf{dk}}(r_1)$.
   - A^* outputs a response $\hat{\mathsf{ct}}$.

   Sim then decrypts the evaluated ciphertext to get a'.

4. **ZK Argument by Malicious** A^***:** Sim takes the role of the honest B in the ZK argument A^* gives.

5. **Matching** B**'s Coins to** A^***'s Coins:** Sim sends $b = a' \oplus r$ to A^*. A^* sends back a and r_a, Sim verifies that r_a indeed opens $\mathsf{cmt}_{\mathsf{A}^*}$ to a, and thus decides if the output of the protocol is $a \oplus b$ or $\perp$. Finally, Sim outputs the inner state of A^* along with the protocol output (either $a \oplus b$ or $\perp$) and transcript.

It is clear that given oracle access to copies of ρ, the expected running time of the simulator is polynomial, as it performs polynomial-time computations plus executing the ZK argument simulator $\mathsf{Sim}_{\mathrm{NP}}$, which also runs in expected polynomial time. It remains to explain why the above simulation process yields an output that's computationally indistinguishable from what's generated in a real interaction between $\mathsf{A}^*(\rho)$ and B.

Claim 4.3 (CF Simulator Output Validity)**.** *Let* $\mathsf{A}^* = \{\mathsf{A}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ *be a polynomial-size quantum adversary, then,*

$$\{\mathsf{VIEW}_{\mathsf{A}^*_\lambda}\langle \mathsf{A}^*(\rho_\lambda), \mathsf{B}\rangle(1^\lambda, 1^{k(\lambda)})\}_{\lambda \in \mathbb{N}} \approx_c \{\mathsf{Sim}(r, \mathsf{A}^*_\lambda, O_{\rho_\lambda}) \mid r \leftarrow U_{k(\lambda)}\}_{\lambda \in \mathbb{N}} \ .$$

Proof. Define the following hybrid processes:

- $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(1)}$: This process acts like the regular interaction $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle$, except that it has oracle access O_ρ to the inner state ρ of A^*, and in step 2, instead of executing the argument between the parties, we take the ZK simulator $\mathsf{Sim}_{\mathrm{NP}}$ to simulate A^*'s view. To use the simulator we need to define what is the instance-advice distribution $(x, \rho^{(x)}) \leftarrow \sigma$, and let $\mathsf{Sim}_{\mathrm{NP}}$ have oracle access to O_σ.

  – The distribution $(x, \rho^{(x)}) \leftarrow \sigma$ is the distribution on protocol transcript (x) and inner state $(\rho^{(x)})$ of A^* that is generated by the interaction between $\mathsf{A}^*(\rho)$ and B until the end of step 1, that is, x is a commitment to 0 by B, and $\rho^{(x)}$ is simply the inner state ρ of A^* before protocol starts.

- $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(2)}$: Acts like $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(1)}$, except that in step 1, B commits to 1 instead of to 0.

- $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(3)}$: Acts like $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(2)}$, except that in step 4a, B sends an SFE encryption ct_{B} of the randomness that it used in step 1 when it committed for 1.

- $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(4)}$: Acts like $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(3)}$, except that in step 6, B outputs $b = a' \oplus r$ for a random $r \leftarrow U_{k(\lambda)}$, instead of a random string $b \leftarrow U_{k(\lambda)}$. Observe that this distribution, given the fact the r is sampled uniformly and is not used in the process before this last step, is exactly $\{\mathsf{Sim}(r, \mathsf{A}^*_\lambda, O_{\rho_\lambda}) \mid r \leftarrow U_{k(\lambda)}\}$.

We now explain why $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle$ is computationally indistinguishable from $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(1)}$, and why each consecutive pair of distributions are computationally indistinguishable. By the transitivity of computational indistinguisability, after we show the above, we are done.

- $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle \approx_c \mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(1)}$: Follows from the ZK property of definition 3.3.
- $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(1)} \approx_c \mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(2)}$: Follows from performing a cutoff on the ZK simulator's running time (the full technical reasoning can be seen as part of the proof of Claim 4.2), together with the hiding of the commitment algorithm $\mathsf{Com}(\cdot)$.
- $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(2)} \approx_c \mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(3)}$: Follows from the input privacy (encryption security) property of the SFE encryption.
- $\mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(3)} \equiv \mathsf{VIEW}_{\mathsf{A}^*}\langle \mathsf{A}^*(\rho), \mathsf{B}\rangle^{(4)}$: These two distributions in fact yield exactly the same distribution, because in both cases the random coins of B in step 6 distribute exactly like U_k.

□

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

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

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 106–115, 2001.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 320–331, 2018.

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

[BGJ+13] Elette Boyle, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, and Amit Sahai. Secure computation against adaptive auxiliary information. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 316–334, 2013.

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

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

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

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, 1984.

[BOCG$^+$06] Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 249–260. IEEE, 2006.

[BP15] Nir Bitansky and Omer Paneth. On non-black-box simulation and the impossibility of approximate obfuscation. *SIAM J. Comput.*, 44(5):1325–1383, 2015.

[Bra18] Zvika Brakerski. Quantum fhe (almost) as secure as classical. In *Annual International Cryptology Conference*, pages 67–95. Springer, 2018.

[CFGS18] Alessandro Chiesa, Michael A. Forbes, Tom Gur, and Nicholas Spooner. Spatial isolation implies zero knowledge even in a quantum world. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 755–765, 2018.

[CLP13] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero knowledge from p-certificates. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 50–59, 2013.

[CPS16] Kai-Min Chung, Rafael Pass, and Karn Seth. Non-black-box simulation from one-way functions and applications to resettable security. *SIAM J. Comput.*, 45(2):415–458, 2016.

[DFS04] Ivan Damgård, Serge Fehr, and Louis Salvail. Zero-knowledge proofs and string commitments withstanding quantum attacks. In *Annual International Cryptology Conference*, pages 254–272. Springer, 2004.

[DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 251–260, 2009.

[DNS10] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In *Annual Cryptology Conference*, pages 685–706. Springer, 2010.

[DNS12] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Annual Cryptology Conference*, pages 794–811. Springer, 2012.

[FP96] Christopher A. Fuchs and Asher Peres. Quantum-state disturbance versus information gain: Uncertainty relations for quantum information. *Phys. Rev. A*, 53:2038–2045, Apr 1996.

[GHKW17] Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, pages 537–566, 2017.

[GK96a] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for np. *Journal of Cryptology*, 9(3):167–189, 1996.

[GK96b] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996.

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

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

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all np statements in zero-knowledge and a methodology of cryptographic protocol design. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 171–185. Springer, 1986.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229, 1987.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

[Goy13] Vipul Goyal. Non-black-box simulation in the fully concurrent setting. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 221–230, 2013.

[GSY19] Alex Bredariol Grilo, William Slofstra, and Henry Yuen. Perfect zero knowledge for quantum multiprover interactive proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:86, 2019.

[HIK$^+$11] Iftach Haitner, Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions of protocols for secure computation. *SIAM J. Comput.*, 40(2):225–266, 2011.

[HSS11] Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In *Annual Cryptology Conference*, pages 411–428. Springer, 2011.

[Kob03] Hirotada Kobayashi. Non-interactive quantum perfect and statistical zero-knowledge. In *Algorithms and Computation, 14th International Symposium, ISAAC 2003, Kyoto, Japan, December 15-17, 2003, Proceedings*, pages 178–188, 2003.

[Liu06] Yi-Kai Liu. Consistency of local density matrices is qma-complete. In *Approximation, randomization, and combinatorial optimization. algorithms and techniques*, pages 438–449. Springer, 2006.

[LN11] Carolin Lunemann and Jesper Buus Nielsen. Fully simulatable quantum-secure coin-flipping and applications. In *International Conference on Cryptology in Africa*, pages 21–40. Springer, 2011.

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

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

[Mah18b] Urmila Mahadev. Classical verification of quantum computations. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 259–267, 2018.

[MHNF15] Tomoyuki Morimae, Masahito Hayashi, Harumichi Nishimura, and Keisuke Fujii. Quantum merlin-arthur with clifford arthur. *arXiv preprint arXiv:1506.06447*, 2015.

[OPCPC14] Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private fhe. In *Annual Cryptology Conference*, pages 536–553. Springer, 2014.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 366–375, 2002.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

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

[Unr16a] Dominique Unruh. Collapse-binding quantum commitments without random oracles. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 166–195. Springer, 2016.

[Unr16b] Dominique Unruh. Computationally binding quantum commitments. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 497–527. Springer, 2016.

[VDGC97] Jeroen Van De Graaf and C Crepeau. *Towards a formal definition of security for quantum protocols*. Université de Montréal, 1997.

[Wat02] John Watrous. Limits on the power of quantum statistical zero-knowledge. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 459–468. IEEE, 2002.

[Wat09] John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

[WZ82] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.

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