The Round Complexity of Quantum Zero-Knowledge

Orestis Chardouvelis[*1] and Giulio Malavolta[2]

[1]National Technical University of Athens
[2]Max Planck Institute for Security and Privacy

Abstract

We study the round complexity of zero-knowledge for QMA (the quantum analogue of NP). Assuming the quantum quasi-polynomial hardness of the learning with errors (LWE) problem, we obtain the following results:

- 2-Round statistical witness indistinguishable (WI) arguments for QMA.
- 4-Round statistical zero-knowledge arguments for QMA in the plain model, additionally assuming the existence of quantum fully homomorphic encryption. This is the first protocol for constant-round *statistical* zero-knowledge arguments for QMA.
- 2-Round computational (statistical, resp.) zero-knowledge for QMA in the timing model, additionally assuming the existence of post-quantum non-parallelizing functions (time-lock puzzles, resp.).

All of these protocols match the best round complexity known for the corresponding protocols for NP with post-quantum security. Along the way, we introduce and construct the notions of sometimes-extractable oblivious transfer and sometimes-simulatable zero-knowledge, which might be of independent interest.

[*]Work done while the author was an intern at the Max Planck Institute for Security and Privacy.

1 Introduction

Zero-knowledge (ZK) proofs allow one to prove the veracity of a statement while revealing nothing beyond that. Since their introduction [GMR89], ZK proofs have had a profound impact on cryptography and theoretical computer science at large. Due to their foundational importance and large applicability, ZK proof systems have been the objective of a long series of work aiming at understanding the necessary assumptions and their round complexity: Under standard computational assumptions, any NP statement can be proven in as few as four rounds of interaction [GMW86, GK96].

The situation is however drastically different when moving to the quantum settings: ZK proofs for QMA (the quantum analogue of NP) have been introduced only recently [BJSW16] and the best known result, in terms of round complexity, is from the very recent work of Bitansky and Shmueli [BS20] where they presented a constant-round computational zero-knowledge argument system (i.e. with computational soundness). Given the current state of affairs, one may wonder whether proving QMA statements inherently introduces additional rounds of interaction. In this work, we study this problem and we give strong evidence that this is *not* the case, presenting protocols in a variety of settings that match the round complexity of their classical counterparts in the same adversarial settings, i.e. with security against quantum attackers.

Our Results. We begin by considering a weak version of zero-knowledge, namely, witness indistinguishability (WI), which only guarantees that a distinguisher cannot tell whether the prover used w_0 or w_1, where (w_0, w_1) are two valid witnesses for the given statement. While not immediately meaningful on its own, this notion and protocol will serve as the basis for our further results. We construct a 2-round protocol with statistical WI, assuming the quasi-polynomial hardness of the learning with errors (LWE) problem [Reg05]. This matches the round complexity of statistical WI protocols for NP [KKS18, BFJ$^+$20, GJJM20].

Theorem 1.1 (Informal)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem, there exists a 2-round statistical WI argument for QMA.*

Next, as our main result, we show how to compile the above WI protocols, into a fully-fledged 4-round *statistical* ZK argument for QMA. The protocol is a round compressed version of the [BS20] approach and, as such, also has a non-blackbox simulator.[1] In contrast to [BS20] our protocol achieves statistical ZK and relies on computational assumptions only to argue about soundness. On the flip side, we rely on the (quantum) *quasi-polynomial* security of the LWE problem and of the quantum fully-homomorphic encryption (QFHE).

Instrumental to our result are the notions of sometimes-extractable 3-round oblivious transfer and sometimes-simulatable 3-round ZK proofs, which we define and construct. Our protocol matches the round complexity of the best known ZK proofs/arguments for NP against quantum adversaries.

Theorem 1.2 (Informal)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem and a quasi-polynomially secure QFHE scheme, there exists a 4-round statistical ZK argument for QMA.*

[1] There is evidence [CCLY21] that non-blackbox simulation is necessary for constant-round ZK against quantum adversaries.

Interestingly, plugging in a 2-round WI protocol for NP we obtain a 4-round statistical ZK argument for NP, secure under the same assumptions against quantum adversaries. Prior to our work, post-quantum *statistical* ZK for NP was only known in polynomial rounds [Unr12, ACP20].

Theorem 1.3 (Informal)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem and a quasi-polynomially secure QFHE scheme, there exists a 4-round post-quantum statistical ZK argument for NP.*

Finally we consider the question of 2-round ZK in the timing model: Since 2-round ZK is known to be impossible [GO94] without additional assumptions, a common relaxation is to allow parties to reliably measure time during the execution of the protocol. In this context, we revisit the Dwork-Stockmeyer [DS02] approach and lift it to the quantum settings. In addition to quasi-polynomial LWE, we assume the existence of a post-quantum non-parallelizing function (e.g. repeated hashing).

Theorem 1.4 (Informal)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem, an FHE scheme, and an average-case non-parallelizing function, there exists a 2-round computational ZK argument for QMA (for NP, resp.) with quantum (classical, resp.) communication in the timing model.*

A shortcoming of the above approach is that it only achieves computational ZK. To overcome this issue, we propose a different route to construct statistical ZK in the timing model, which relies on slightly stronger assumptions (namely, post-quantum time-lock puzzles).

Theorem 1.5 (Informal)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem and a quasi-polynomially sequential post-quantum time-lock puzzle, there exists a 2-round statistical ZK argument for QMA (for NP, resp.) with quantum (classical, resp.) communication in the timing model.*

2 Technical Overview

Here we present an overview of the main technical ideas presented in the paper. For further details, we refer the reader to the technical sections.

2.1 Witness-Indistinguishable Arguments

We begin by outlining the construction of a 2-round WI protocol, which will constitute the basis for the following results. 2-round WI protocols for NP under the same assumptions are known [KKS18, GJJM20, BFJ$^+$20], so the main challenge here is to lift them to the QMA settings. Our construction is based on the template from [Shm20], which in turn relies on the sigma protocol for QMA introduced in [BG20]. Such a protocol consists of the canonical three messages: A commitment α, a challenge β, and a response γ. The important property (also used in [Shm20]) is that the computation of β and γ is completely classical. In our protocol we actually use a new version of the [BG20] protocol that achieves statistical ZK, which we construct from the parallel repetition of [BG20] combined with an SBSH commitment (which is explained below). For the sake of this overview though we can ignore this aspect and simply consider a three message sigma protocol.

The basic idea of the protocol is to use a maliciously circuit private (levelled) homomorphic encryption to round-collapse the sigma protocol: The verifier sends to the prover an encrypted challenge β, then the prover computes in plain a commitment α and evaluates homomorphically

the response function to return an encrypted version of γ. The verifier, who knows the secret key of the homomorphic encryption, can decrypt the incoming ciphertext and verify the validity of the transcript (α, β, γ). While intuitively the soundness follows from the semantic security of the homomorphic encryption scheme, turning this into a provably secure scheme requires us to tweak it with some additional tools:

- We let the prover compute a commitment to the random coins used in the homomorphic evaluation procedure. This allows the verifier (in the soundness proof) to check the validity of the transcript without knowing the secret key of the homomorphic encryption scheme. To achieve this while maintaining statistical WI, we use a special kind of sometimes-binding statistically hiding (SBSH) commitment. This is a standard statistically hiding commitment scheme, which has a certain (negligibly small) probability to be perfectly binding. When such event happens, the verifier can extract the committed message. Soundness is then argued by a standard complexity leveraging argument.
- We use a dual-track approach, where we repeat the above process twice and we let the prover show that at least one of the two instances was computed correctly, via a statistical WI (for NP). This is sufficient to prove the overall WI of the protocol since we can "switch" the witness step-by-step for each branch.

All of the above building blocks can be instantiated assuming the quasi-polynomial hardness of the LWE problem. Since this protocol constitutes the basis of the upcoming constructions, they will also be based on the quantum quasi-polynomial hardness of LWE.

2.2 Zero Knowledge Arguments

To achieve ZK, we leverage the generic approach of [AL20, BS20], which introduces a non-black-box quantum extraction technique that allows the simulator to emulate the honest prover without knowing the witness. The extraction protocol consists of constant (> 4) number of rounds and the resulting ZK scheme for QMA achieves only computational ZK, while our objective will be achieving statistical ZK settings while at the same time squeezing the number of rounds down to 4. Throughout the rest of this overview (and for all of the upcoming protocols) the main technical challenge will be to construct a simulator against a quantum verifier (i.e., achieving post-quantum zero-knowledge), so our discussion will mostly concentrate on this aspect. The class of statements that we can prove in zero-knowledge depends on the underlying WI: By plugging in a WI for NP we obtain post-quantum ZK for NP and by plugging in a WI for QMA we obtain ZK for QMA.

Some Cryptographic Tools. Before presenting the construction we recall some necessary tools that we use. The first is a quantum fully homomorphic encryption (QFHE) scheme. Similar to an FHE scheme, this tool allows us to additionally perform homomorphic evaluations of quantum circuits and inputs. We also use a compute-and-compare obfuscation. A compute-and-compare program $\mathbf{CC}[f, s, z]$ where f is a function and s, z are strings, outputs z on every input x such that $f(x) = s$ and rejects the rest of the inputs. A compute-and-compare obfuscator compiles a $\mathbf{CC}$ program to the obfuscated program $\widetilde{\mathbf{CC}}$ and is computationally indistinguishable from a simulated dummy program, that rejects on all inputs. Finally, we use a conditional disclosure of secrets (CDS) protocol. This two-round protocol is parametrized by a statement z and a message m from the sender: The receiver is able to recover m if the statement is correct, whereas m stays

hidden if this is not the case. Simultaneously, the witness w (held by the receiver) for x should be kept secret from the eyes of the sender.

The "Homomorphic Trapdoor" Technique. We now briefly recall the simulation technique from [BS20, AL20]. For simplicity, we consider a verifier that never aborts and that is explainable, i.e. it computes all its messages in the support of algorithms as dictated by the honest protocol. The crux of their protocol consists of the following extractable commitment scheme (where the verifier will later play the role of the sender, and the prover the role of the receiver):

- The sender samples two random strings s, td in addition to:
  - A public and secret key $(\mathsf{pk}, \mathsf{sk})$ of a QFHE scheme and an encryption $c_{td} = \mathsf{QFHE.Enc}(\mathsf{pk}, \mathsf{td})$ of td.
  - The obfuscated program $\widetilde{\mathbf{CC}} \leftarrow Obf(\mathbf{CC}[f, s, (\mathsf{sk}, m)]$, where f is the decryption circuit of QFHE.

  The sender sends $\mathsf{pk}, c_{\mathsf{td}}, \widetilde{\mathbf{CC}}$ to the receiver.
- The receiver encodes a guess y via the CDS protocol.
- The sender responds with a message encrypted via the CDS protocol, such that, if the guess y is equal to td, then the message decrypts to s. Otherwise it returns $\perp$.

Intuitively, such a procedure is binding since the message in the obfuscated program is uniquely determined, and hiding since no receiver guesses td correctly, except with negligible probability. Furthermore, a simulator can extract the message (sk, m) and simulate the sender's view: After the simulator gets the first message, it homomorphically computes the sender's last message using the sender's circuit with inputs the encryption of td and the inner state of the sender. The result of the homomorphic computation is the message encrypted with the CDS, whose statement is satisfied and hence it returns s encrypted under QFHE. This is exactly the input needed for $\widetilde{\mathbf{CC}}$ in order to obtain m. Note that the simulator is able to also produce a valid transcript T without rewinding the adversary, since the $\mathbf{CC}$ program also returns sk, which can be used by the simulator to decrypt the QFHE-encrypted messages.

From WI to ZK in 4 Rounds. Given the above extractable commitment, one can boost a 2-round WI argument into a fully-fledged 4-round ZK protocol, as follows: The verifier in the first round sends a commitment to zero with randomness r (which is the same randomness used in the QFHE keys generation algorithm). Then, they perform the above quantum extraction technique with r as the message m. After the interaction, the prover utilizes the WI argument introduced before and sends a proof that either he knows the randomness r or that $x \in \mathcal{L}$.

To rule out mauling attacks where the prover could maul a QFHE encryption of td into a valid witness for the CDS protocol, we additionally include an SBSH commitment of y, which can be extracted with low probability, thus enabling a reduction against the semantic security of the QFHE scheme. Consistency is guaranteed by checking that the SBSH commitment is well-formed within the CDS protocol (i.e. the prover includes also the randomness of the SBSH commitment as part of the witness).

Sometimes-Extractable SRP Oblivious Transfer. One immediate issue with the above protocol is that existing 2-round CDS protocols only provide computational security for the receiver, which would result in us achieving computational ZK. Since statistical receiver security is impossible in 2 rounds (as it would imply a non-interactive statistically hiding commitment), we turn our attention to a possible 3-round protocol. Towards achieving that goal, we consider the less ambitious objective of constructing a 3-round statistically receiver private oblivious transfer (SRP-OT). For 3-round SRP-OT, even defining security is a non-trivial task since the choice bit of the receiver is not determined in an information theoretic sense. For this reason, we introduce the notion of sometimes-extractable SRP-OT, which provides us with the following guarantees:

- Statistical Receiver Privacy: The choice bit b of the receiver is statistically hidden.
- Sometimes Extractability: With exponentially small probability, the receiver message is in "binding" mode and the choice bit b is uniquely determined and can be efficiently extracted.
- Computational Sender Privacy: Conditioned on the fact that the the above extraction happens, the message $m_{b\oplus 1}$ is computationally hidden from the eyes of the receiver.

This notion seems to inherently require complexity leveraging in order to be fulfilled. On the brighter side, this notion is sufficient for our purposes and are able to provide a construction assuming quasi-polynomial LWE. The scheme is an augmented version of the 3-round OT presented in [GJJM20], where we additionally include an SBSH commitment to the choice bit b (which will enable the above extraction procedure). To ensure that the choice bit is consistently set across the OT and the SBSH commitment, we will also let the verifier compute a statistical WI proof that certifies this.[2]

Equipped with sometimes-extractable SRP-OT we are then able to construct a post quantum CDS protocol with statistical receiver privacy: The receiver sets the bit decomposition of its witness w to be the choice bits for the SRP-OT. Then the sender computes a garbled circuit that, on input w, checks whether $w \in \mathsf{R}_{\mathcal{L}}(x)$ and returns the message m if this is the case. The SRP-OT is then used to transmit the labels corresponding to w to the receiver, which can retrieve the message by locally evaluating a garbled circuit.

Malicious Verifiers. The only remaining problem is that the ZK protocols are simulatable under the assumption that the verifier is non-aborting and explainable. To deal with aborting verifiers, we (as done in [BS20],) define two simulators, an aborting and a non-aborting one, and we let the combined simulator guess which of the two he should use. Watrous' rewinding lemma [Wat09] allows the simulator to rewind until the guess was correct without disturbing the verifier's state. To ensure that the verifier is explainable, we augment the protocol with an additional ZK proof (from the verifier to the prover) that the messages where computed honestly. Note that even in our protocol for QMA the verifier is completely classical, so ZK for NP always suffices. In order to achieve statistical soundness and maintain the statistical ZK property though, we need a *delayed-input* ZK proof (with statistical soundness). This proof needs also to not exceed 3 rounds so as not to increase the rounds of the original protocol. Unfortunately, we do not have a 3-round ZK proof, let alone a post-quantum one.

[2]The astute reader may wonder why the WI guarantee is enough here. To prove receiver privacy we will add a trapdoor statement where the witness can be computed in exponential time. Since the argument is anyway statistical, this does not add any additional assumption.

Sometimes-Simulatable Zero-Knowledge. We observe however, that for our case a weaker notion suffices. In particular, we introduce the notion of *sometimes simulatable* zero-knowledge, where simulation is possible with some (negligibly) small probability. In order to be meaningfully used, one must set the security parameters of other primitives to account for this exponential loss, much like with SBSH commitments. Sometimes-simulatable (SSim) ZK is reminiscent of ZK with super-polynomial simulation (SPS) [Pas03a] but with a crucial difference: In SPS-ZK the simulator runs in super-polynomial time, whereas in SSim-ZK the simulator runs in polynomial time but only has an exponentially small success probability. This difference is important in our settings since (in general) we cannot rewind the state of the verifier and it is therefore important that the simulation is straight-line.

The construction utilizes the sometimes-extractable SRP-OT protocol constructed above in order to "delay" the input of the Blum sigma protocol for Graph Hamiltonicity, similarly as it is done in [JKKR17]. The verifier samples a challenge $\beta \in \{0,1\}^n$ and sets the choice bits of the SRP-OT to be the bit representation of β. Then the prover samples n independent commitment-response tuples $(\alpha, \gamma_0, \gamma_1)$ for the sigma-protocol and sets the i-th pair (γ_0, γ_1) as the messages for the i-th instance of the SRP-OT. The verifier can then verify the sigma protocol by decoding the SRP-OT and checking the validity of the transcript. Note that the challenge β is statistically hidden and therefore the resulting proof is statistically sound, by the statistical soundness of the sigma protocol. To argue about computational ZK, recall that with a certain low-probability the SRP-OT is in extractable mode, which makes it possible for the simulator to recover β and simulate the transcript. By setting the parameters appropriately, we can then reduce against the computational ZK of the sigma protocol.

2.3 Zero Knowledge in the Timing Model

Finally, we investigate how to achieve ZK in two rounds, by moving the protocol to the timing model. In other words, we assume that the parties can reliably measure the lapse of time during the interaction. In order to achieve this, we assume the existence of a non-parallelizing function F. A non-parallelizing function is a function that can be computed in time T, while the result of the function with an input x cannot be predicted by an attacker with depth less than T (i.e it cannot be run quicker in parallel time).

Computational Zero-Knowledge. For our first construction we revisit the [DS02] approach. The protocol is parametrized by a time parameter T and and we assume a sub-exponentially non-parallelizing function F, secure against algorithms with depth less than T. The prover first computes an encryption α of a random string, and its homomorphic evaluation β with the function F. Then, after the verifier sends a random value x^*, the prover sends a proof that either $x \in \mathcal{L}$ or that he knows an encryption α of x^*. Eventually, the verifier accepts if the prover responds in time, the proof is valid and the homomorphic evaluation of α with F is equal to β.

Intuitively, the protocol is secure because the prover doesn't have the time to homomorphically recompute β. Thus, soundness is proven by reducing to breaking the non-parallelizability of F. The zero-knowledge property is easily proven, having in mind that the simulator is allowed to "freeze time" (from the perspective of the verifier) while simulating the accepting transcript. Note that the simulation is straight-line and does not copy nor rewinds the state of the verifier, which makes it suitable for the quantum settings.

Statistical Zero-Knowledge. Assuming slightly stronger assumptions, we propose a different approach, achieving statistical ZK. In particular we assume the existence of a post-quantum time-lock puzzle. A time-lock puzzle essentially provides an encryption that is breakable after time T, but where one cannot gain a significant speedup with parallel computation (similar to the non-parallelizability). For the sake of this overview, we only consider explainable verifiers and the conversion to malicious verifiers can be done with standard techniques [BKP19, CDM20].

In our construction, the verifier sends a commitment to 0 with randomness r, along with a time-lock puzzle encrypting said randomness. Then the prover sends a WI proof proving that either it knows a statement $x \in \mathcal{L}$ or that it knows the randomness r. The verifier accepts if the prover responds in time and the proof is valid. Intuitively, a malicious prover cannot solve the time-lock puzzle in the necessary time, whereas in order to prove ZK, the simulator can again "freeze time" and solve the time-lock puzzle, acquiring the randomness and using it as a witness in the WI proof.

2.4 Related Work

A series of recent works [BG20, CVZ20, ACGH20, CCY20, Shm20, BM21] considers the problem of non-interactive ZK for QMA. All of these works require some notion of trusted setup, which is unavoidable for 1-round protocols. We also mention another line of work [Unr12, HSS11, LN11, ARU14, AL20] that studies the strong notion of arguments of knowledge in the quantum settings. In the multi-prover settings, it is known that NEXP [CFGS18] and MIP* [GSY19] admit perfect ZK interactive proofs (sound against entangled quantum provers). Finally, there exists a construction of a 3-round statistical ZK argument for NP [BP19] based on the protocol from [BKP18] which relies on keyless multi-collision resistant hash and (polynomial) LWE. However, to the best of our knowledge, the protocol is analyzed only in the classical settings and it appears to be a challenging problem to apply the same simulation strategy against quantum verifiers. For a detailed discussion on the challenges of post-quantum zero-knowledge we refer the reader to [BS20].

Comparison with [BG20]. In [BG20] the authors construct a sigma protocol for QMA which satisfies computational honest-verifier zero-knowledge (HVZK) and statistical soundness, while using statistically binding commitments and achieving $1 - 1/\mathsf{poly}(\lambda)$ soundness error. They also claim an extension to statistical HVZK, using a collapse-binding commitment instead of a statistically binding one (without a formal proof). An earlier version of this work used such a protocol as a building block for our 2-round statistical WI argument. However, Fermi Ma pointed out to us a gap in the analysis of [BG20] for their statistical HVZK variant.

In the revised version of this work, we include a new version of the [BG20] protocol that achieves computational soundness and statistical HVZK. The protocol is a k-fold parallel repetition of [BG20] combined with the use of an SBSH commitment (instead of a collapse-binding one). In order to prove soundness (with negligible error), a standard complexity leveraging argument is used to condition on the SBSH commitment being in binding mode. For details, we refer the reader to Section 4.2. We shall mention that, because of the SBSH commitment, the above construction requires the quasi-polynomial hardness of LWE assumption.

3 Preliminaries

We denote by λ the security parameter. 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 $n > N$, $f(n) < n^{-c}$. We recall some standard notation for 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 in λ.
- We sometimes think about PPT Turing machines as polynomial-size uniform families of circuits. A polynomial-size circuit family C is a sequence of circuits $C = \{C_\lambda\}_{\lambda\in\mathbb{N}}$, 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 write $m \in M(x)$ to denote that m is in the support of $M(x; \cdot)$. Finally we write $y \leftarrow\$ M(x)$ to denote the computation of M on input x with some uniformly sampled random coins.

3.1 Quantum Adversaries

We recall some notation for quantum computation and we define the notions of computational and statistical indistinguishability for quantum adversaries. Various parts of what follows are taken almost in verbatim from [BS20].

- 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 they are equivalent models). We call a polynomial-size quantum circuit family $C = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ 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 decohere and are automatically measured in the standard basis.
- A quantum interactive algorithm (in the two-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 holds. 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 for itself.

Throughout this work, we model efficient adversaries as quantum circuits with non-uniform quantum advices. This is denoted by $\mathcal{A}^* = \{\mathcal{A}^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, where $\{\mathcal{A}^*_\lambda\}_{\lambda\in\mathbb{N}}$ 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. We now define the formal notion of computational indistinguishability in the quantum settings.

Definition 3.1 (Computational Indistinguishability)**.** *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 there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$ and all non-uniform QPT distinguishers with quantum advice $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$|\Pr[\mathcal{A}(X;\rho) = 1] - \Pr[\mathcal{A}(Y;\rho) = 1]| \leq \mu(\lambda)$$

where $X \leftarrow\!\$\, X_\lambda$ and $Y \leftarrow\!\$\, Y_\lambda$.

The trace distance between two quantum distributions (X_λ, Y_λ), denoted by $\mathsf{TD}(X_\lambda, Y_\lambda)$, is a generalization of statistical distance to the quantum setting and represents the maximal distinguishing advantage between two quantum distributions by an unbounded quantum algorithm. We define below the notion of statistical indistinguishability.

Definition 3.2 (Statistical Indistinguishability)**.** *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 statistically indistinguishable (denoted by $\mathcal{X} \approx_s \mathcal{Y}$) if there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$, it holds that*

$$\mathsf{TD}(X_\lambda, Y_\lambda) \leq \mu(\lambda).$$

The Class QMA. A language $\mathcal{L} = (\mathcal{L}_{\text{yes}}, \mathcal{L}_{\text{no}})$ in QMA is defined by a tuple $(\mathcal{V}, p, \alpha, \beta)$, where p is a polynomial, $\mathcal{V} = \{V_\lambda\}_{\lambda\in\mathbb{N}}$ is a uniformly generated family of circuits such that for every λ, V_λ takes as input a string $x \in \{0,1\}^\lambda$ and a quantum state $|\psi\rangle$ on $p(\lambda)$ qubits and returns a single bit, and $\alpha, \beta : \mathbb{N} \to [0,1]$ are such that $\alpha(\lambda) - \beta(\lambda) \geq 1/p(\lambda)$. The language is then defined as follows.

- For all $x \in \mathcal{L}_{\text{yes}}$ of length λ, there exists a quantum state $|\psi\rangle$ of size at most $p(\lambda)$ such that the probability that V_λ accepts $(x, |\psi\rangle)$ is at least $\alpha(\lambda)$. We denote the (possibly infinite) set of quantum witnesses that make V_λ accept x by $\mathsf{R}_{\mathcal{L}}(x)$.
- For all $x \in \mathcal{L}_{\text{no}}$ of length λ, and all quantum states $|\psi\rangle$ of size at most $p(\lambda)$, it holds that V_λ accepts on input $(x, |\psi\rangle)$ with probability at most $\beta(\lambda)$.

3.2 Learning with Errors

We recall the definition of the learning with errors (LWE) problem [Reg05].

Definition 3.3 (Learning with Errors)**.** *The LWE problem is parametrized by a modulus $q = q(\lambda)$, polynomials $n = n(\lambda)$ and $m = m(\lambda)$, and an error distribution χ. The LWE problem is hard if it holds that*

$$(\mathbf{A}, \mathbf{A}\cdot\mathbf{s} + \mathbf{e}) \approx_c (\mathbf{A}, \mathbf{u})$$

where $\mathbf{A} \leftarrow\!\$\, \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \leftarrow\!\$\, \mathbb{Z}_q^n$, $\mathbf{u} \leftarrow\!\$\, \mathbb{Z}_q^m$, and $\mathbf{e} \leftarrow\!\$\, \chi^m$.

As shown in [Reg05, PRS17], for any sufficiently large modulus q the LWE problem where χ is a discrete Gaussian distribution with parameter $\sigma = \xi q \geq 2\sqrt{n}$ (i.e. the distribution over $\mathbb{Z}$ where the probability of x is proportional to $e^{-\pi(|x|/\sigma)^2}$), is at least as hard as approximating the shortest independent vector problem (SIVP) to within a factor of $\gamma = \tilde{O}(n/\xi)$ in *worst case* dimension n lattices. In this work we rely on the *quasi-polynomial hardness of LWE*. This is a stronger assumption than plain LWE, where the distinguisher for the two distributions is allowed to run on quasi-polynomial time.

3.3 Pseudorandom Functions

We recall the standard notion of pseudorandom function (PRF) [GGM86].

Definition 3.4 (Pseudorandom Function)**.** *A pseudorandom function* $(\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ *consists of the following efficient algorithms.*

- $\mathsf{PRF.Gen}(1^\lambda)$*: On input the security parameter, the key generation algorithm returns a key* k*.*
- $\mathsf{PRF.Eval}(k, x)$*: On input a key* k *and a string* $x \in \{0,1\}^\lambda$*, the evaluation algorithm returns a string* $y \in \{0,1\}^{e(\lambda)}$*.*

The scheme must be pseudorandom in the following sense.

Definition 3.5 (Pseudorandomness)**.** *A pseudorandom function* $(\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ *is pseudorandom if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$ *and all non-uniform QPT distinguishers with quantum advice* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\left|\Pr\left[\mathcal{A}(\rho)^{\mathsf{PRF.Eval}(k,\cdot)} = 1\right] - \Pr\left[\mathcal{A}(\rho)^{f(\cdot)} = 1\right]\right| \leq \mu(\lambda)$$

where $k \leftarrow\!\!\$\, \mathsf{PRF.Gen}(1^\lambda)$ *and* $f : \{0,1\}^\lambda \to \{0,1\}^{e(\lambda)}$ *is a uniformly sampled truly random function.*

3.4 Garbled Circuits

We recall the definition of a garbling scheme for circuits [Yao86, AIK04, BHR12].

Definition 3.6 (Garbled Circuit)**.** *A garbling scheme for circuits is a tuple of PPT algorithms* $(\mathsf{Garble}, \mathsf{GEval})$ *with the following syntax.*

- $\mathsf{Garble}\left(1^\lambda, C\right)$*:* Garble *takes as input a security parameter* 1^λ*, a circuit* C*, and outputs a* garbled circuit $\widetilde{C}$ *along with labels* $\{\ell_{i,b}\}_{i\in\{1,\ldots,n\},b\in\{0,1\}}$*, where* n *is the length of the input to* C*.*
- $\mathsf{GEval}\left(\widetilde{C}, \{\ell_{i,x_i}\}_{i\in\{1,\ldots,n\}}\right)$*: Given a garbled circuit* $\widetilde{C}$ *and a sequence of input labels* $\{\ell_{i,x_i}\}_{i\in\{1,\ldots,n\}}$*,* GEval *outputs a string* y*.*

We recall the notion of completeness.

Definition 3.7 (Completeness)**.** *A garbling scheme* $(\mathsf{Garble}, \mathsf{GEval})$ *is complete if for any circuit* C *and input* $x \in \{0,1\}^n$ *we have that:*

$$\Pr\left[C(x) = \mathsf{GEval}\left(\widetilde{C}, \{\ell_{i,x_i}\}_{i\in\{1,\ldots,n\}}\right)\right] = 1$$

where $(\widetilde{C}, \{\ell_{i,b}\}_{i\in\{1,\ldots,n\},b\in\{0,1\}}) \leftarrow \mathsf{Garble}\left(1^\lambda, C\right)$*.*

We define the notion of (statistical) simulation security, which is achievable for circuits in NC1.

Definition 3.8 (Security)**.** *A garbling scheme* $(\mathsf{Garble}, \mathsf{GEval})$ *is simulation secure if there exists a PPT simulator* GSim *such that for any circuit* C *and input* $x \in \{0,1\}^n$*, we have that*

$$\left(\widetilde{C}, \{\ell_{i,x_i}\}_{i\in\{1,\ldots,n\}}\right) \approx_s \mathsf{GSim}\left(1^\lambda, 1^{|C|}, 1^n, C(x)\right)$$

where $(\widetilde{C}, \{\ell_{i,b}\}_{i\in\{1,\ldots,n\},b\in\{0,1\}}) \leftarrow \mathsf{Garble}\left(1^\lambda, C\right)$*.*

3.5 Interactive Proofs and Sigma Protocols

We present the definitions of interactive proof systems and sigma protocols. Much of the following material is taken in verbatim from [Shm20]. We denote by (P, V) and interactive protocol between a prover P and a verifier V. The output of the verifier is denoted by $\mathsf{Out}(\mathsf{P}, \mathsf{V})$. For an honest verifier, the output is a classical bit that denotes acceptance or rejection. If the verifier is corrupted, the output can be an arbitrary quantum state. We define completeness in the following.

Definition 3.9 (Completeness)**.** *An interactive protocol (P, V) for a language $\mathcal{L} \in \mathsf{QMA}$ with relation $\mathsf{R}_{\mathcal{L}}$ is complete if there exists a polynomial p and a negligible function μ such that for all $\lambda \in \mathbb{N}$, all $x \in \mathcal{L}$, and all $|w\rangle \in \mathsf{R}_{\mathcal{L}}(x)$, it holds that*

$$\Pr\left[\mathsf{Out}(\mathsf{P}(|w\rangle^{\otimes p(\lambda)}, x), \mathsf{V}(x)) = 1\right] \geq 1 - \mu(\lambda).$$

Next we define the notion of (non-adaptive) computational soundness.

Definition 3.10 (Computational Soundness)**.** *An interactive protocol (P, V) for a language $\mathcal{L} \in \mathsf{QMA}$ with relation $\mathsf{R}_{\mathcal{L}}$ is computationally sound if there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$, all $x \notin \mathcal{L}$, and all non-uniform QPT provers with quantum advice $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$, it holds that*

$$\Pr\left[\mathsf{Out}(\mathcal{A}(x; \rho), \mathsf{V}(x)) = 1\right] \leq \mu(\lambda).$$

Sigma Protocols. We explicitly define sigma protocols (Σ), a special case of interactive protocols for QMA, and we define a special honest-verifier zero knowledge guarantee that is satisfied by some protocols of interest.

Definition 3.11 (Sigma Protocol)**.** *A sigma protocol $(\Sigma.\mathsf{Com}, \Sigma.\mathsf{Chal}, \Sigma.\mathsf{Resp})$ consists of the following efficient algorithms.*

- $\Sigma.\mathsf{Com}(|w\rangle^{\otimes p(\lambda)}; r)$*: On input $p(\lambda)$-many copies of the witness and some (classical) random coins $r \in \{0,1\}^{q(\lambda)}$, the commitment algorithm returns a first commitment $|\alpha\rangle$.*
- $\Sigma.\mathsf{Chal}(x)$*: On input the instance x, the challenge algorithm returns a uniformly sampled (classical) string $\beta \in \{0,1\}^{b(\lambda)}$.*
- $\Sigma.\mathsf{Resp}(\beta, r)$*: On input the challenge β and the classical random coins r, the response algorithm returns a classical response γ.*

We highlight the fact that both the challenge and the response algorithm are completely classical: The only quantum computation needed is for the $\Sigma.\mathsf{Com}$ algorithm and for verifying that $x \in \mathcal{L}$, given the protocol transcript. We now define the notion of computational special honest-verifier zero-knowledge.

Definition 3.12 (Computational Special Honest-Verifier Zero-Knowledge)**.** *A sigma protocol $(\Sigma.\mathsf{Com}, \Sigma.\mathsf{Chal}, \Sigma.\mathsf{Resp}$ satisfies (computational) special honest-verifier zero-knowledge if there exists a QPT simulator $\Sigma.\mathsf{Sim}$ such that for all $\lambda \in \mathbb{N}$, all $x \in \mathcal{L}$, and all $|w\rangle \in \mathsf{R}_{\mathcal{L}}(x)$, it holds that*

$$(\Sigma.\mathsf{Com}(|w\rangle^{\otimes p(\lambda)}; r), \Sigma.\mathsf{Resp}(\beta, r)) \approx_c \Sigma.\mathsf{Sim}(x, \beta)$$

where $r \leftarrow\!\!\$\, \{0,1\}^{q(\lambda)}$ and $\beta \leftarrow\!\!\$\, \{0,1\}^{b(\lambda)}$.

The statistical notion is defined analogously, except that we require statistical indistinguishability between the two distributions. It was recently shown by Broadbent and Grilo [BG20] how to obtain a sigma protocol for QMA satisfying statistical soundness and special honest-verifier zero-knowledge, assuming a (classical) post-quantum non-interactive statistically binding bit commitment scheme [LS19, HW18]. Here we restate the main theorem of such a work.

Lemma 3.13 ([BG20])**.** *Assuming the post-quantum hardness of the LWE problem, there exists a sigma protocol* $(\Sigma.\mathsf{Com}, \Sigma.\mathsf{Chal}, \Sigma.\mathsf{Resp})$ *satisfying statistical soundness and computational special honest-verifier zero-knowledge.*

In this work we are also interested in the reverse guarantees, i.e. computational soundness and stastistical zero-knowledge. Since (classical) statistically hiding commitments notoriously require two rounds of interaction, we extend the syntax of the sigma protocol to have the verifier sampling the commitment key $\mathsf{ck} \leftarrow\!\!\$\ \Sigma.\mathsf{Gen}(1^\lambda)$, which is also given as an input to the $\Sigma.\mathsf{Com}$ algorithm. The definition of special honest-verifier zero-knowledge and soundness are extended accordingly. In Section 4.2 we show how to construct such protocol assuming the quasi-polynomial hardness of LWE.

The Timing Model. In this work we also consider the timing model for some of our protocols. In such a model, all parties in the interaction have access to local clocks. Similarly to [DS02], the simulator controls the clock of all parties (and in particular the one of the verifier) and may decide to stop the clock for an arbitrary amount of time and for and arbitrary number of times.

3.6 Statistical ZAPs for NP

A ZAP protocol is a two-round witness-indistinguishable argument where the first message is instance-independent. We say that the protocol achieves *multi-theorem* security if the first round can be fixed once and for all and can be reused for an unbounded amount of second rounds. In the other hand, if the first round has to be re-initialized for each run of the protocol, we say that the ZAP achieves only *single-theorem* security. Additionally, we say that the protocol is *public coin* if the output of the protocol is publicly computable given the protocol transcript, and otherwise we say that the protocol is *private coin*. We being by defining the syntax of (public coin) statistical ZAPs for NP.

Definition 3.14 (ZAP Protocol for NP)**.** *A ZAP protocol* $(\mathsf{ZAP.Setup}, \mathsf{ZAP.Prove}, \mathsf{ZAP.Verify})$ *for a language* $\mathcal{L} \in \mathsf{NP}$ *with relation* $\mathsf{R}_\mathcal{L}$ *consists of the following efficient algorithms.*

- $\mathsf{ZAP.Setup}(1^\lambda)$*: On input the security parameter* 1^λ*, the setup returns a common reference string* crs *and a trapdoor* td*.*
- $\mathsf{ZAP.Prove}(\mathsf{crs}, w, x)$*: On input a common reference string* crs*, a witness* w*, and a statement* x*, the proving algorithm returns a proof* π*.*
- $\mathsf{ZAP.Verify}(\mathsf{td}, \pi, x)$*: On input a trapdoor* td*, a proof* π*, and a statement* x*, the verification algorithm returns a bit* $\{0, 1\}$*.*

The definitions of completeness and computational soundness are identical to those given for general interactive proof systems (Section 3.5). Note that all definitions that we present here are

for the single-theorem case. This is without loss of generality, since single-theorem soundness (witness indistinguishability, resp.) is equivalent to multi-theorem soundness (witness indistinguishability, resp.) for public coin protocols. In the following we present the notion of (statistical) witness indistinguishability.

Definition 3.15 (Statistical Witness Indistinguishability)**.** *A ZAP protocol* $(\mathsf{ZAP.Setup}, \mathsf{ZAP.Prove}, \mathsf{ZAP.Verify})$ *for a language* $\mathcal{L} \in \mathsf{NP}$ *with relation* $\mathsf{R}_{\mathcal{L}}$ *is witness indistinguishable if for all* $\lambda \in \mathbb{N}$*, all* $x \in \mathcal{L}$*, all pairs of witnesses* $(w_0, w_1) \in \mathsf{R}_{\mathcal{L}}$*, and all common reference strings* crs *it holds that*

$$(\mathsf{crs}, \mathsf{ZAP.Prove}(\mathsf{crs}, w_0, x)) \approx_s (\mathsf{crs}, \mathsf{ZAP.Prove}(\mathsf{crs}, w_1, x)).$$

It was recently shown in [BFJ+20, GJJM20] that statistical ZAPs for NP exist assuming the quasi-polynomial (quantum) hardness of the LWE problem.

Lemma 3.16 ([BFJ+20, GJJM20])**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem, there exists a public coin ZAP for* NP $(\mathsf{ZAP.Setup}, \mathsf{ZAP.Prove}, \mathsf{ZAP.Verify})$.

3.7 Sometimes-Binding Statistically Hiding Commitments

We introduce the notion of sometimes-binding statistically hiding (SBSH) commitments, as defined in [LVW20].

Definition 3.17 (SBSH Commitment)**.** *An SBSH commitment scheme* $(\mathsf{SBSH.Gen}, \mathsf{SBSH.Key}, \mathsf{SBSH.Com})$ *consists of the following efficient algorithms.*

- $\mathsf{SBSH.Gen}(1^\lambda)$*: On input the security parameter* 1^λ*, the generation algorithm returns a partial commitment key* ck_0*.*
- $\mathsf{SBSH.Key}(\mathsf{ck}_0)$*: On input a partial key* ck_0*, the key agreement algorithm returns the complement of the key* ck_1*.*
- $\mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), m)$*: On input a commitment key* $(\mathsf{ck}_0, \mathsf{ck}_1)$ *and a message* m*, the commitment algorithm returns a commitment* c*.*

The commitment must satisfy the notion of statistical hiding.

Definition 3.18 (Statistical Hiding)**.** *An SBSH commitment scheme* $(\mathsf{SBSH.Gen}, \mathsf{SBSH.Key}, \mathsf{SBSH.Com})$ *is statistically hiding if for all* $\lambda \in \mathbb{N}$*, all partial keys* ck_0*, and all pairs of messages* (m_0, m_1)*, it holds that*

$$(\mathsf{ck}_0, \mathsf{ck}_1, \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), m_0)) \approx_s (\mathsf{ck}_0, \mathsf{ck}_1, \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), m_1))$$

where $\mathsf{ck}_1 \leftarrow\!\!\$\, \mathsf{SBSH.Key}(\mathsf{ck}_0)$.

Next we define the notion of sometimes-binding for an SBSH commitment scheme. We define the set $\mathsf{Binding}$ as the set of all commitment keys $(\mathsf{ck}_0, \mathsf{ck}_1)$ such that any resulting commitment is perfectly binding. We present the definition of the property in the following.

Definition 3.19 (Sometimes Binding)**.** *An SBSH commitment scheme* $(\mathsf{SBSH.Gen}, \mathsf{SBSH.Key}, \mathsf{SBSH.Com})$ *is* (ε, δ)*-sometimes binding if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$ *and all (stateful) QPT distinguishers* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$*, it holds that*

$$\Pr\left[\mathcal{A}(\mathsf{st}; \rho) = 1 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] = \varepsilon(\lambda) \cdot \Pr\left[\mathcal{A}(\mathsf{st}; \rho) = 1\right] + \delta(\lambda) \cdot \mu(\lambda)$$

where $\mathsf{ck}_0 \leftarrow\!\!\$\, \mathsf{SBSH.Gen}(1^\lambda)$ *and* $(\mathsf{st}, \mathsf{ck}_1) = \mathcal{A}(\mathsf{ck}_0; \rho)$.

We also require the existence of a polynomial-time extractor $\mathsf{SBSH.Ext}$ that, on input the random coins r used in the $\mathsf{SBSH.Gen}$ algorithm, extracts the committed message m from the protocol transcript if $(\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}$. The works of [KS17, KKS18, BFJ+20, GJJM20] construct SBSH commitment schemes (using a slightly different syntax) for quasi-polynomial (ε, δ) assuming the quasi-polynomial hardness of two-round statistically sender private oblivious transfer. Thus we can state the following lemma.

Lemma 3.20 ([KS17, KKS18, BFJ+20, GJJM20])**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem and quasi-polynomial (ε, δ), there exists an (ε, δ)-sometimes binding SBSH commitment scheme* $(\mathsf{SBSH.Gen}, \mathsf{SBSH.Key}, \mathsf{SBSH.Com})$.

3.8 Compute-and-Compare Obfuscation

Here we define compute-and-compare circuits (CC) and obfuscators for said circuits (Obf). The definitions are taken in verbatim from [BS20].

Definition 3.21 (Compute-and-Compare Circuit)**.** *Let $f : \{0,1\}^n \to \{0,1\}^\lambda$ and let $u \in \{0,1\}$, $z \in \{0,1\}^*$ be strings. Then $\mathbf{CC}[f,u,z](x)$ is a circuit that returns z if $f(x) = y$, and $\perp$ otherwise. $\mathbf{CC}$ has a canonical description from which f, u and z can be read.*

For the following definition, Obf is a PPT algorithm that takes as input a CC circuit and outputs a new circuit $\widetilde{\mathbf{CC}}$.

Definition 3.22 (Correctness)**.** *A PPT algorithm* Obf *is a correct compute-and-compare obfuscator if for any circuit $f : \{0,1\}^n \to \{0,1\}^\lambda$, $u \in \{0,1\}$, $z \in \{0,1\}^*$*

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

We define simulation security in the following.

Definition 3.23 (Simulation Security)**.** *A PPT algorithm* Obf *is a simulation secure compute-and-compare obfuscator if there exists a PPT Simulator* Sim *such for every two polynomials $\ell_1(\cdot), \ell_2(\cdot)$,*

$$\left\{\widetilde{\mathbf{CC}} \mid u \leftarrow \{0,1\}^\lambda, \widetilde{\mathbf{CC}} \leftarrow \mathsf{Obf}(\mathbf{CC})\right\}_{\lambda,f,z} \approx_c \left\{\mathsf{Sim}(1^{\ell_1(\lambda)}, 1^{\ell_2(\lambda)}, 1^\lambda)\right\}_{\lambda,f,z}$$

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

Constructions based on the quantum hardness of LWE can be found in [GKW17, WZ17, GKVW19].

3.9 Quantum One-Time Pad

We recall the quantum one-time pad (QOTP) construction [AMTDW00] for quantum states. We explicitly consider the scheme that allows one to encrypt an n-qubit quantum state with unconditional security.

Definition 3.24 (Quantum One-Time Pad)**.** *A quantum one-time pad* $(\mathsf{QOTP.Gen}, \mathsf{QOTP.Enc}, \mathsf{QOTP.Dec})$ *consists of the following efficient algorithms.*

- $\mathsf{QOTP.Gen}(1^n)$: *For all $i = 1 \dots n$ sample two classical bits $(x_i, z_i) \leftarrow\!\!\$\ \{0,1\}^2$. Return the one-time key $\mathsf{otk} = (x_1, z_1, \dots, x_n, z_n)$.*
- $\mathsf{QOTP.Enc}(\mathsf{otk}, |\psi\rangle)$: *On input a one-time key otk and an n-qubit state $|\psi\rangle$, apply the Pauli transformation $X^{x_i} Z^{z_i}$ to the i-th qubit, for all $i = 1 \dots n$. Return the resulting state $|\phi\rangle$.*
- $\mathsf{QOTP.Dec}(\mathsf{otk}, |\phi\rangle)$: *On input a one-time key otk and an n-qubit state $|\phi\rangle$, apply the reverse Pauli transformation $Z^{z_i} X^{x_i}$ qubit-by-qubit to recover the original state.*

More explicitly, the (single qubit) Pauli transformation $X^{x_i} Z^{z_i}$ is the following unitary:

$$(\alpha_0 |0\rangle + \alpha_1 |1\rangle) \rightarrow (\alpha_0 |x_i\rangle + (-1)^{z_i} \alpha_1 |x_i \oplus 1\rangle).$$

As shown in [AMTDW00], the above scheme can be used to transform *any* n-qubit quantum state into a totally mixed state (no matter if some of its initial qubits are in an entangled state).

3.10 Homomorphic Encryption

We recall the notion of homomorphic encryption [Gen09].

Definition 3.25 (Homomorphic Encryption)**.** *A homomorphic encryption scheme* $(\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ *consists of the following efficient algorithms.*

- $\mathsf{FHE.Gen}(1^\lambda)$: *On input the security parameter, the key generation algorithm returns secret/public key pair* $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{FHE.Enc}(\mathsf{pk}, m)$: *On input the public key pk and a message m, the encryption algorithm returns a ciphertext c.*
- $\mathsf{FHE.Eval}(\mathsf{pk}, C, c)$: *On input the public key pk, a (classical) circuit C, and a ciphertext c, the evaluation algorithm returns an evaluated ciphertext $\tilde{c}$.*
- $\mathsf{FHE.Dec}(\mathsf{sk}, c)$: *On input the secret key sk and a ciphertext c, the decryption algorithm returns a message m.*

We say that a scheme is fully homomorphic (FHE) if the evaluation algorithm supports all polynomial-size classical circuits (without posing an a-priori bound on the size of $|C|$). If the size of C needs to be fixed at the time of key generation, then we say that the scheme is *levelled* homomorphic. It is well-known that levelled FHE schemes can be based on the hardness of the (plain) LWE problem [BV11, BV14]. Throughout this work, we are mostly going to consider levelled FHE schemes and we will simply refer to them as FHE schemes whenever it is clear from the context. We recall the notion of (single-hop) evaluation correctness in the following.

Definition 3.26 (Evaluation Correctness)**.** *A homomorphic encryption scheme* $(\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ *is correct if for all $\lambda \in \mathbb{N}$, all $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{FHE.Gen}(1^\lambda)$, all messages m, and all polynomial-size circuits C, it holds that*

$$\Pr\left[\mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{FHE.Eval}(\mathsf{pk}, C, \mathsf{FHE.Enc}(\mathsf{pk}, m))) = C(m)\right] = 1$$

We recall the notion of semantic security for public-key encryption.

Definition 3.27 (Semantic Security)**.** *A homomorphic encryption scheme* $(\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ *is semantically secure if for all* $\lambda \in \mathbb{N}$ *and all pairs of messages* (m_0, m_1)*, it holds that*

$$\mathsf{FHE.Enc}(\mathsf{pk}, m_0) \approx_c \mathsf{FHE.Enc}(\mathsf{pk}, m_1)$$

where $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\, \mathsf{FHE.Gen}(1^\lambda)$.

Finally we define the notion of (malicious) statistical circuit privacy circuit privacy for FHE [OPP14].

Definition 3.28 (Statistical Circuit Privacy)**.** *A homomorphic encryption scheme* $(\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ *is (malicious) statistically circuit private if there exists a pair of unbounded algorithms* $\mathsf{FHE.Ext}$ *and* $\mathsf{FHE.Sim}$ *such that for all* $\lambda \in \mathbb{N}$*, all public keys* pk^**, all ciphertexts* c^**, and all circuits* C*, it holds that*

$$\mathsf{FHE.Eval}(\mathsf{pk}^*, C, c^*) \approx_s \mathsf{FHE.Sim}(1^\lambda, \mathsf{pk}^*, c^*, C(x^*))$$

where $x^* = \mathsf{FHE.Ext}(1^\lambda, \mathsf{pk}^*, c^*)$.

It is shown in [OPP14] that any FHE scheme with semi-honest circuit privacy can be converted into one with malicious circuit privacy generically, by additionally assuming a two-round statistically sender-private oblivious transfer. The latter can in turn be instantiated from LWE [BD18, DGI+19, BDGM19]. Taken together, these give us the following result.

Lemma 3.29 ([BD18])**.** *Assuming the post-quantum hardness of the LWE problem, there exists an FHE scheme* $(\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ *with (malicious) statistical circuit privacy.*

3.11 Non-Parallelizing Functions

We recall the definition of average-case non-parallelizing functions. Non-parallelizing functions can be instantiated via repeated hashing or via the universal construction of [JMR20], additionally assuming an FHE scheme.

Definition 3.30 (Average-Case Non-Parallelizing Functions [BGJ+16])**.** *A function family* $\{F_{\lambda,T} : \mathcal{X}_{\lambda,T} \to \mathcal{Y}_{\lambda,T}\}_{\lambda,T\in\mathbb{N}}$ *is* T*-non-parallelizing with gap* $\zeta < 1$*, if for all* $x \in \mathcal{X}$*,* $F_{\lambda,T}(x)$ *can be computed in time* T *and there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$ *and all non-uniform QPT algorithm with quantum advice* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ *of depth at most* T^ζ*, it holds that*

$$\Pr\left[\mathcal{A}(x;\rho) = F_{\lambda,T}(x) \mid x \leftarrow\!\!\$\, \mathcal{X}_\lambda\right] = \mu(\lambda).$$

In this work we are interested in an even stronger variant, where we assume that the above holds also against sub-exponential size (but still depth bounded) adversaries, and we refer to this variant as *sub-exponential* average-case non-parallelizing functions.

3.12 Time-Lock Puzzles

We recall the definition of time-lock puzzles [RSW96] in the following.

Definition 3.31 (Time-Lock Puzzles)**.** *A time-lock puzzle* $(\mathsf{TLP.Gen}, \mathsf{TLP.Solve})$ *consists of the following efficient algorithms.*

- $\mathsf{TLP.Gen}(1^\lambda, T, m)$*: On input the security parameter, a time parameter* T*, and a message* m*, the puzzle generation algorithm returns a puzzle* Z*.*

- $\mathsf{TLP.Solve}(Z)$*: On input a puzzle* Z*, the solving algorithm returns a message* m*.*

In terms of efficiency, we only require that the algorithm $\mathsf{TLP.Gen}$ runs in time polynomial in λ and at most logarithmic in T. Whereas for correctness, we require that for all $\lambda \in \mathbb{N}$, all polynomials T, all messages m it holds that

$$\mathsf{TLP.Solve}(\mathsf{TLP.Gen}(1^\lambda, T, m)) = m$$

and the algorithm $\mathsf{TLP.Solve}$ runs in time linear in T. We recall the definition of security below.

Definition 3.32 (Sequentiality [BGJ+16])**.** *A time-lock puzzle* $(\mathsf{TLP.Gen}, \mathsf{TLP.Solve})$ *is* T*-sequential with gap* $\zeta < 1$ *if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$ *and all non-uniform QPT algorithm with quantum advice* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ *of depth at most* T^ζ*, it holds that*

$$\Pr\left[\mathcal{A}(Z;\rho) = b \;\middle|\; b \leftarrow\!\$\, \{0,1\}; Z \leftarrow\!\$\, \mathsf{TLP.Gen}(1^\lambda, T, b)\right] = 1/2 + \mu(\lambda).$$

4 Witness-Indistinguishable Arguments for QMA

This section is devoted to the definition and description of our 2-round witness indistinguishable (WI) argument for QMA.

4.1 Definition

We recall the definition of 2-round WI for QMA. We consider a variant where the first message is instance-independent and we define directly this notion.

Definition 4.1 (2-Round WI for QMA)**.** *A WI protocol* $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ *for a language* $\mathcal{L} \in \mathsf{QMA}$ *with relation* $\mathsf{R}_\mathcal{L}$ *consists of the following efficient algorithms.*

- $\mathsf{WI.Setup}(1^\lambda)$*: On input the security parameter* 1^λ*, the setup returns a classical common reference string* crs *and a classical trapdoor* td*.*
- $\mathsf{WI.Prove}(\mathsf{crs}, |w\rangle^{\otimes p(\lambda)}, x)$*: On input a common reference string* crs*,* $p(\lambda)$*-many copies of the witness* $|w\rangle$*, and a statement* x*, the proving algorithm returns a quantum state* $|\pi\rangle$*.*
- $\mathsf{WI.Verify}(\mathsf{td}, |\pi\rangle, x)$*: On input a trapdoor* td*, a quantum state* $|\pi\rangle$*, and a statement* x*, the verification algorithm returns a classical bit* $\{0,1\}$*.*

For the definition of completeness we refer the reader to Section 3.5. In the following we define the notion of (non-adaptive) multi-theorem computational soundness for private-coin ZAPs.

Definition 4.2 (Computational Soundness)**.** *A WI protocol* $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ *for a language* $\mathcal{L} \in \mathsf{QMA}$ *with relation* $\mathsf{R}_\mathcal{L}$ *is computationally sound if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$*, all* $x \notin \mathcal{L}$*, and all non-uniform QPT provers with quantum advice* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\Pr\left[|\pi\rangle = \mathcal{A}(\mathsf{crs}, x; \rho) \wedge \mathsf{WI.Verify}(\mathsf{td}, |\pi\rangle, x) = 1\right] \leq \mu(\lambda)$$

where $(\mathsf{crs}, \mathsf{td}) \leftarrow\!\$\, \mathsf{WI.Setup}(1^\lambda)$*.*

We now define the notion of (statitical) witness indistinguishability.

Definition 4.3 (Statistical Witness Indistinguishability)**.** *A WI protocol* $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ *for a language* $\mathcal{L} \in \mathsf{QMA}$ *with relation* $\mathsf{R}_{\mathcal{L}}$ *is statistically witness indistinguishable if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$ *and all (stateful) admissible distinguishers* $\mathcal{A}$*, it holds that*

$$\left|\Pr\left[\mathcal{A}(\mathsf{crs},\mathsf{st})^{\mathsf{WI.Prove}^0(\mathsf{crs},\cdot,\cdot,\cdot)}=1\right]-\Pr\left[\mathcal{A}(\mathsf{crs},\mathsf{st})^{\mathsf{WI.Prove}^1(\mathsf{crs},\cdot,\cdot,\cdot)}=1\right]\right|\leq\mu(\lambda).$$

where $(\mathsf{st},\mathsf{crs})=\mathcal{A}(1^\lambda)$ *and the oracle* $\mathsf{WI.Prove}^b$ *takes as input a statement* x *and* $p(\lambda)$*-many copies of two witnesses* $|w_0\rangle$ *and* $|w_1\rangle$ *and returns* $\mathsf{WI.Prove}(\mathsf{crs},|w_b\rangle^{\otimes p(\lambda)},x)$*. We say that the distinguisher* $\mathcal{A}$ *is admissible if it holds that* $(|w_0\rangle^{\otimes p(\lambda)},|w_1\rangle^{\otimes p(\lambda)})\in\mathsf{R}_{\mathcal{L}}(x)$*.*

4.2 Statistically Zero-Knowledge Sigma Protocol

In the following we show a new variant of the sigma protocol from [BG20] that achieves statistical zero-knowledge (and negligible soundness error).[3] Before presenting the protocol we recall their main information theoretic result.

Lemma 4.4 ([BG20])**.** *Let* $\mathcal{L}\in\mathsf{QMA}$ *be a language with relation* $\mathsf{R}_{\mathcal{L}}$*, then there exist two polynomials* m *and* p *such that for all* $x\in\mathcal{L}$ *there exists an efficient deterministic algorithm that computes* $m(\lambda)$ *5-qubits POVMs* $\{\Pi_1,I-\Pi_1\}\dots\{\Pi_{m(\lambda)},I-\Pi_{m(\lambda)}\}$ *that acts on a state of size* $p(\lambda)$ *such that the following properties are satisfied.*

- *(Completeness) For all* $\lambda\in\mathbb{N}$ *and all* $x\in\mathcal{L}$ *there exists a negligible function* μ *and an efficiently computable* $p(\lambda)$*-qubits state* ω *such that for all* $c\in\{1,\dots m(\lambda)\}$ *it holds that*

$$\mathsf{Tr}(\Pi_c\omega)\geq 1-\mu(\lambda).$$

- *(Simulatability) For all* $\lambda\in\mathbb{N}$ *and all* $x\in\mathcal{L}$ *there exists a set of 5-qubit density matrices* $\rho(x,S)$ *such that for every* $S\subseteq\{1,\dots p(\lambda)\}$ *and* $|S|=5$ *it holds that*

$$\mathsf{Tr}_{\bar{S}}(\omega)\approx_s\rho(x,S)$$

  where $\mathsf{Tr}_{\bar{S}}(\omega)$ *denotes the state* ω *tracing out all qubits not in* S*.*

- *(Soundness) For all* $\lambda\in\mathbb{N}$ *and all* $x\notin\mathcal{L}$ *there exists a polynomial* q *such that for all* $p(\lambda)$*-qubit states* ω *it holds that*

$$\frac{1}{m(\lambda)}\sum_{c=1\dots m(\lambda)}\mathsf{Tr}(\Pi_c\omega)\leq 1-\frac{1}{q(\lambda)}.$$

We are now ready to present our statistically zero-knowledge sigma protocol, which is essentially a k-fold parallel repetition of [BG20] combined with an SBSH commitment. Specifically, we assume an SBSH commitment scheme $(\mathsf{SBSH.Gen},\mathsf{SBSH.Key},\mathsf{SBSH.Com})$ that satisfies $(\varepsilon(\lambda),\varepsilon(\lambda)^2)$-sometimes binding, for some fixed negligible function $\varepsilon(\lambda)$. The protocol is shown in Figure 1.

[3] Although [BG20] also claimed a statistical zero-knowledge variant, the analysis had a gap. See Section 2.4 for a more comprehensive discussion.

Statistical ZK Sigma Protocol for QMA

- **Setup:** The setup algorithm ($\Sigma.\mathsf{Gen}$) samples and returns an SBSH commitment key $\mathsf{ck}_0 \leftarrow\!\$\, \mathsf{SBSH.Gen}(1^\lambda)$.
- **Commitment:** The commitment algorithm ($\Sigma.\mathsf{Com}$) takes as input k copies of the witness ω and samples a commitment key $\mathsf{ck}_1 \leftarrow\!\$\, \mathsf{SBSH.Key}(\mathsf{ck}_0)$. Then, for $i = 1 \dots k$ does the following:
  - Sample a one-time key $\mathsf{otk}_i \leftarrow\!\$\, \mathsf{QOTP.Gen}(1^\lambda)$.
  - Compute $\psi_i = \mathsf{QOTP.Enc}(\mathsf{otk}_i, \omega)$.
  - Parse $\mathsf{otk}_i = (x_{i,1}, z_{i,1}, \dots, x_{i,p(\lambda)}, z_{i,p(\lambda)})$. Then, for all $j = 1 \dots p(\lambda)$, compute $d_{i,j} = \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), (x_{i,j}, z_{i,j}); r_{i,j})$, where $r_{i,j}$ are uniformly sampled random coins.

  Return ck_1 and $\{\psi_i, d_{i,1}, \dots, d_{i,p(\lambda)}\}_{i=1\dots k}$.
- **Challenge:** The challenge algorithm ($\Sigma.\mathsf{Chal}$) samples and returns $c_i \leftarrow\!\$\, \{1, \dots, m(\lambda)\}$, for all $i = 1 \dots k$.
- **Response:** The response algorithm ($\Sigma.\mathsf{Resp}$) for all $i = 1 \dots k$ does the following:
  - Let S_i be the set on which Π_{c_i} acts non-trivially.
  - Return $(x_{i,j}, z_{i,j}, r_{i,j})$ for all $j \in S_i$.
- **Verify:** The verifier accepts if for all $i = 1 \dots k$ the following holds:
  - For all $j \in S_i$, $r_{i,j}$ is a valid opening for $(x_{i,j}, z_{i,j})$.
  - Measure $X^{\mathbf{x}_i} Z^{\mathbf{z}_i} \psi_i Z^{\mathbf{z}_i} X^{\mathbf{x}_i}$, where $(\mathbf{x}_i, \mathbf{z}_i)$ denote the set $\{x_{i,j}, z_{i,j}\}_{i \in S_i}$, with POVMs $\{\Pi_{c_i}, I - \Pi_{c_i}\}$ and accept if the outcome is Π_{c_i}.

Figure 1: Statistical Zero Knowledge Sigma Protocol for QMA.

Computational Soundness. We show that the protocol is computationally sound.

Theorem 4.5 (Soundness)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem, the protocol in Figure 1 is computationally sound.*

Proof. Let $x \notin \mathcal{L}$ be the challenge statement and let Cheat be the event where the prover causes the verifier to accept x. Assume towards contradiction that

$$\Pr[\mathsf{Cheat}] \geq \varepsilon(\lambda).$$

Then, by the $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes binding property of the SBSH commitment scheme, we have that

$$\Pr[\mathsf{Cheat} \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda)) \tag{1}$$

for some negligible function $\mu(\lambda)$. Running the extractor $\mathsf{SBSH.Ext}$ on the first message of the prover, we obtain the committed keys $\{\mathsf{otk}_i = (x_{i,1}, z_{i,1}, \ldots, x_{i,p(\lambda)}, z_{i,p(\lambda)})\}_{i=1\ldots k}$. Let ζ be the state returned by the prover. Conditioned on the prover opening to the correct strings (as otherwise the verifier would reject) the acceptance probability is

$$\prod_{i=1\ldots k} \frac{1}{m(\lambda)} \sum_{c_i=1\ldots m(\lambda)} \mathsf{Tr}\left(\Pi_{c_i} X^{\mathbf{x}_i} Z^{\mathbf{z}_i} \zeta_i Z^{\mathbf{z}_i} X^{\mathbf{x}_i}\right)$$

where $\zeta_1 = \zeta$ and $\zeta_{i+1} = \Pi_{c_i} X^{\mathbf{x}_i} Z^{\mathbf{z}_i} \zeta_i Z^{\mathbf{z}_i} X^{\mathbf{x}_i}$, for all $i = 1 \ldots k-1$. Also recall that $(\mathbf{x}_i, \mathbf{z}_i)$ is defined as $\{x_{i,j}, z_{i,j}\}_{j \in S_i}$. Equivalently, we can rewrite

$$\prod_{i=1\ldots k} \frac{1}{m(\lambda)} \sum_{c_i=1\ldots m(\lambda)} \mathsf{Tr}\left(\Pi_{c_i} X^{\mathbf{x}_i} Z^{\mathbf{z}_i} \zeta_i Z^{\mathbf{z}_i} X^{\mathbf{x}_i}\right) = \prod_{i=1\ldots k} \frac{1}{m(\lambda)} \sum_{c_i=1\ldots m(\lambda)} \mathsf{Tr}\left(\Pi_{c_i} \mathsf{QOTP.Dec}(\mathsf{otk}_i, \zeta_i)\right)$$

redefining $\zeta_{i+1} = \Pi_{c_i} \mathsf{QOTP.Dec}(\mathsf{otk}_i, \zeta_i)$. Finally, by Lemma 4.4 we can bound

$$\begin{aligned}\prod_{i=1\ldots k} \frac{1}{m(\lambda)} \sum_{c_i=1\ldots m(\lambda)} \mathsf{Tr}\left(\Pi_{c_i} \mathsf{QOTP.Dec}(\mathsf{otk}_i, \zeta_i)\right) &\leq \prod_{i=1\ldots k} \mathsf{max}_{\phi_i} \frac{1}{m(\lambda)} \sum_{c_i=1\ldots m(\lambda)} \mathsf{Tr}\left(\Pi_{c_i} \phi_i\right) \\ &\leq \left(1 - \frac{1}{q(\lambda)}\right)^k.\end{aligned}$$

For a large enough k, this is a contradicts Equation (1) and concludes our proof. □

Statistical Special Honest Verifier Zero-Knowledge. We show that our protocol satisfies statistical special honest-verifier zero-knowledge. Note that this does not follow via a generic parallel repetition argument, since the commitment key is reused across different executions.

Theorem 4.6 (Zero-Knowledge)**.** *The protocol in Figure 1 satisfies statistical special honest-verifier zero-knowledge.*

Proof. Before describing the simulator, observe that the view of the honest execution of the protocol (for a uniformly sampled $(c_1, \ldots, c_k) \leftarrow\!\!\$\, \{1, \ldots, m(\lambda)\}^k$) consists of

$$\mathsf{ck}_1 \bigotimes_{i=1\ldots k} \begin{array}{l} \psi_i = \mathsf{QOTP.Enc}(\mathsf{otk}_i, \omega) \\ \otimes \{\mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), (x_{i,j}, z_{i,j}); r_{i,j})\}_{j=1\ldots p(\lambda)} \\ \otimes \{(x_{i,j}, z_{i,j}, r_{i,j})\}_{j \in S_i} \end{array}$$

where S_i is the set of 5 qubits on which Π_{c_i} acts non-trivially. On the other hand, the simulator encrypts a dummy state which contains the simulatable density matrices for the positions where Π_{c_i} acts non-trivially. Furthermore all of the commitments corresponding to the complement of these positions are also computed for dummy values. More precisely, the output of the simulator is defined to be

$$\mathsf{ck}_1 \bigotimes_{i=1\ldots k} \begin{array}{l} \psi_i = \mathsf{QOTP.Enc}(\mathsf{otk}_i, \rho(x, S_i) \otimes |0\rangle\langle 0|^{\bar{S}_i}) \\ \otimes \{\mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), (x_{i,j}, z_{i,j}); r_{i,j})\}_{j \in S_i} \\ \otimes \{\mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), (0,0); r_{i,j})\}_{j \notin S_i} \\ \otimes \{(x_{i,j}, z_{i,j}, r_{i,j})\}_{j \in S_i}. \end{array}$$

We now argue that the two distributions are statistically close. This follows by observing that

$$\begin{aligned}\psi_i &= \mathsf{QOTP.Enc}(\mathsf{otk}_i, \rho(x, S_i) \otimes |0\rangle\langle 0|^{\bar{S}_i}) \\ &\approx_s \mathsf{QOTP.Enc}(\mathsf{otk}_i, \mathsf{Tr}_{\bar{S}_i}(\omega) \otimes |0\rangle\langle 0|^{\bar{S}_i}) \\ &\equiv \mathsf{QOTP.Enc}(\mathsf{otk}_i, \mathsf{Tr}_{\bar{S}_i}(\omega)) \otimes I^{\bar{S}_i} \\ &\equiv \mathsf{QOTP.Enc}(\mathsf{otk}_i, \omega)\end{aligned}$$

where the first equality follows from Lemma 4.4 and the second and third follow from the security of the QOTP (note that the view is formally independent of the keys for all $j \notin S_i$). Finally, by the statistical hiding of the SBSH commitment we have that

$$\mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), (0,0); r_{i,j}) \approx_s \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), (x_{i,j}, z_{i,j}); r_{i,j})$$

for all $i = 1 \dots k$ and $j \notin S_i$. Applying these two facts, we obtain that the output of the simulator is statistically close to the real distribution. □

4.3 2-Round Witness-Indistinguishable Arguments for QMA

In the following we describe our protocol for statistical WI for QMA. Let $\varepsilon(\lambda)$ be a (fixed) negligible function. We assume the existence of the following building blocks (all secure against quantum adversaries):

- A sigma protocol $(\Sigma.\mathsf{Gen}, \Sigma.\mathsf{Com}, \Sigma.\mathsf{Chal}, \Sigma.\mathsf{Resp})$ for QMA satisfying statistical special honest-verifier zero-knowledge and with $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ soundness error.
- A public coin ZAP $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ for NP with statistical witness indistinguishability and $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ soundness error.
- A pseudorandom function $(\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ with distinguishing advantage $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$.
- A maliciously circuit private classical (levelled) FHE scheme $(\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ with distinguishing advantage $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$.
- An SBSH commitment scheme $(\mathsf{SBSH.Gen}, \mathsf{SBSH.Key}, \mathsf{SBSH.Com})$ that satisfies $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes binding.

Where $\mu(\lambda)$ is some negligible function and κ is the security parameter of the primitives with super-polynomially bounded disitinguishing advantage. Our protocol is formally described in Figure 2. Completeness of the protocol follows by a standard argument.

Soundness. We show that our protocol satisfies (non-adaptive) soundness. We also note that the proof can be lifted to the adaptive setting (i.e. where the prover can choose the challenge statement adaptively) using complexity leveraging, albeit at the cost of a stronger assumption for the security of the underlying primitives.

Theorem 4.7 (Soundness)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem, the WI argument described in Figure 2 satisfies computational soundness.*

Statistical WI Arguments for QMA

- **Setup:** The setup algorithm samples a PRF key $k \leftarrow\$ \mathsf{PRF.Gen}(1^\kappa)$ and an FHE key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{FHE.Gen}(1^\kappa)$. Additionally it samples a commitment key $\mathsf{ck} \leftarrow\$ \Sigma.\mathsf{Gen}(1^\kappa)$, an SBSH commitment key $\mathsf{ck}_0 \leftarrow\$ \mathsf{SBSH.Gen}(1^\lambda)$, and a common reference string $\mathsf{crs}_{\mathsf{ZAP}} \leftarrow\$ \mathsf{ZAP.Setup}(1^\kappa)$. The algorithm computes $c_k \leftarrow\$ \mathsf{FHE.Enc}(\mathsf{pk}, k)$ and sets the common reference string and the trapdoor as

$$\mathsf{crs} = (\mathsf{pk}, c_k, \mathsf{ck}, \mathsf{ck}_0, \mathsf{crs}_{\mathsf{ZAP}}) \text{ and } \mathsf{td} = (\mathsf{sk}, k).$$

- **Prove:** On input $2p(\lambda)$-many copies of the witness $|w\rangle^{2p(\lambda)}$ and a statement x, the proving algorithm does the following. First, it samples a commitment key $\mathsf{ck}_1 \leftarrow\$ \mathsf{SBSH.Key}(\mathsf{ck}_0)$, then for $b \in \{0,1\}$, it samples a classical string $r_{\Sigma,b} \leftarrow\$ \{0,1\}^\kappa$ and computes the first $|\alpha_b\rangle = \Sigma.\mathsf{Com}(|w\rangle^{\otimes p(\lambda)}, \mathsf{ck}; r_{\Sigma,b})$. Then it evaluates homomorphically the response function of the sigma protocol sampling the challenge from the PRF, i.e. it computes

$$c_{\gamma,b} = \mathsf{FHE.Eval}(\mathsf{pk}, \Sigma.\mathsf{Resp}(\mathsf{PRF.Eval}(\cdot, x\|b), r_{\Sigma,b}), c_k; r_{\mathsf{FHE},b}).$$

  where $r_{\mathsf{FHE},b}$ are some classical random coins. In addition, it computes an SBSH commitment to $r_{\Sigma,b}$ as $c_{r,b} = \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), r_{\Sigma,b}; r_{\mathsf{SBSH},b})$, where $r_{\mathsf{SBSH},b}$ are also uniformly sampled coins. Finally it computes a statistical ZAP π for the classical statement

$$\left\{ \exists\, (b, w_\Sigma, w_{\mathsf{FHE}}, w_{\mathsf{SBSH}}) \text{ s.t. } \begin{array}{l} c_{\gamma,b} = \mathsf{FHE.Eval}(\mathsf{pk}, \Sigma.\mathsf{Resp}(\mathsf{PRF.Eval}(\cdot, x\|b), w_\Sigma), c_k; w_{\mathsf{FHE}}) \\ \wedge\, c_{r,b} = \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), w_\Sigma; w_{\mathsf{SBSH}}) \end{array} \right\}$$

  using $(0, r_{\Sigma,0}, r_{\mathsf{FHE},0}, r_{\mathsf{SBSH},0})$ as a witness. The output of the algorithm is defined as

$$|\pi\rangle = (\mathsf{ck}_1, |\alpha_0\rangle, |\alpha_1\rangle, c_{\gamma,0}, c_{\gamma,1}, c_{r,0}, c_{r,1}, \pi).$$

- **Verify:** The verification algorithm checks that the ZAP π against the common reference string $\mathsf{crs}_{\mathsf{ZAP}}$, then for $b \in \{0,1\}$ does the following. It recomputes the challenge for the sigma protocol $\beta_b = \mathsf{PRF.Eval}(k, x\|b)$ and it recovers the response $\gamma_b = \mathsf{FHE.Dec}(\mathsf{sk}, c_{\gamma,b})$ by decrypting the corresponding FHE ciphertext. Then it checks whether $(|\alpha_b\rangle, \beta_b, \gamma_b)$ is a valid transcript for the sigma protocol. If all of the above conditions are satisfied, the algorithm returns 1, otherwise it returns 0.

Figure 2: Description of a statistical WI argument for QMA.

Proof. We are going to show that the prover success probability is bounded by a negligible function $\varepsilon(\lambda)$. Let $x \notin \mathcal{L}$ be the challenge statement and let Cheat be the event where the prover causes the verifier to accept x. Assume towards contradiction that

$$\Pr[\mathsf{Cheat}] \geq \varepsilon(\lambda).$$

Then, by the $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes binding property of the SBSH commitment scheme, we have

that

$$\Pr\left[\mathsf{Cheat} \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$$

for some negligible function $\mu(\lambda)$. Let $r_0^* = \mathsf{SBSH.Ext}(r, \mathsf{ck}_0, \mathsf{ck}_1, c_{\tau,0})$ and $r_1^* = \mathsf{SBSH.Ext}(r, \mathsf{ck}_0, \mathsf{ck}_1, c_{\tau,1})$ denote the outputs of the extractor on such a transcript, where r denote the random coins used in the $\mathsf{SBSH.Gen}$ algorithm. We now gradually change the verification procedure and we argue that the probability that the above event happens does not decrease significantly.

- The verifier no longer decrypts the FHE ciphertext, instead, for $b \in \{0, 1\}$, it computes $\gamma_b = \Sigma.\mathsf{Resp}(\mathsf{PRF.Eval}(k, x\|b), r_b^*)$ and checks whether the transcript $(|\alpha_b\rangle, \mathsf{PRF.Eval}(k, x\|b), \gamma_b)$ is accepting. If at least one of the two transcripts is accepting and the ZAP π correctly verifies, then the verifier returns 1, otherwise it returns 0. Let Cheat_1 be the event that the prover causes the modified verifier to accept on some $x \notin \mathcal{L}$. We want to argue that

  $$\Pr\left[\mathsf{Cheat}_1 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$$

  for some negligible function $\mu(\lambda)$. To show this, it suffices to consider the case where the prover passes the original verification procedure but fails the modified one. This implies that the prover has computed two inconsistent commitments $(c_{\tau,0}, c_{\tau,1})$ but the ZAP π correctly verifies. Thus, if the inequality above does not hold, then we obtain a contradiction against the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-soundness of the ZAP argument.

- The verifier computes c_k as an encryption of 0 (padded to the appropriate length), i.e. it computes $c_k \leftarrow\$ \mathsf{FHE.Enc}(\mathsf{pk}, 0)$. Let Cheat_2 be the event that the prover causes the modified verifier to accept on some $x \notin \mathcal{L}$. Recall that the modified verifier no longer uses the FHE secret key in its routine. Thus, by the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-semantic security of the FHE scheme we have that

  $$\Pr\left[\mathsf{Cheat}_2 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda)).$$

- Instead of computing $\beta_b = \mathsf{PRF.Eval}(k, x\|b)$, the verifier samples (β_0, β_1) uniformly. By the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-pseudorandomness of the pseudorandom function, we have that

  $$\Pr\left[\mathsf{Cheat}_3 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$$

  where Cheat_3 denotes the event that the prover causes the modified verifier to accept on some $x \notin \mathcal{L}$.

The last inequality implies that either of the sigma protocols $(|\alpha_0\rangle, \beta_0, \gamma_0)$, $(|\alpha_1\rangle, \beta_1, \gamma_1)$ is accepting for some $x \notin \mathcal{L}$, where β_0 and β_1 are sampled uniformly and independently of $|\alpha_0\rangle$ and $|\alpha_1\rangle$, with probability at least $\varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$. This contradicts the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-soundness of the sigma protocol and concludes our proof. □

Witness Indistinguishability. We show that our protocol satisfies statistical witness indistinguishability.

Theorem 4.8 (Statistical Witness Indistinguishability)**.** *The WI argument described in Figure 2 satisfies statistical witness indistinguishability.*

Proof. We begin by fixing the challenge bit $b = 0$ and we gradually modify the experiment through a series of hybrids that we show to be statistically close.

- Hybrid $\mathcal{H}_0$: This is the original experiment with the challenge bit fixed to $b = 0$, i.e. the oracle always uses the witness $|w_0\rangle$.
- Hybrid $\mathcal{H}_1$: In this hybrid we modify the answers to all queries of the adversary to compute $c_{\tau,1}$ as a commitment to 0, i.e. $c_{\tau,1} \leftarrow\$ \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), 0)$. Note that the randomness of the commitment is never used in the proof and thus, by the statistically hiding property of the SBSH commitment we have that

  $$\mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), r_{\Sigma,1}) \approx_s \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), 0).$$

  It follows that the two hybrids are statistically indistinguishable.
- Hybrid $\mathcal{H}_2$: In this hybrid we first run the (unbounded) extractor given by the malicious circuit privacy of the FHE scheme $k^* = \mathsf{FHE.Ext}(1^\kappa, \mathsf{pk}, c_k)$, then we compute the evaluated cirphertext as $c_{\gamma,1} \leftarrow\$ \mathsf{FHE.Sim}(1^\kappa, \mathsf{pk}, c_k, \Sigma.\mathsf{Resp}(\mathsf{PRF.Eval}(k^*, x\|1), r_{\Sigma,1}))$. By the statistical circuit privacy of the FHE scheme we have that

  $$\begin{aligned}&\mathsf{FHE.Eval}(\mathsf{pk}, \Sigma.\mathsf{Resp}(\mathsf{PRF.Eval}(\cdot, x\|1), r_{\Sigma,1}), c_k)\\ &\approx_s \mathsf{FHE.Sim}(1^\kappa, \mathsf{pk}, c_k, \Sigma.\mathsf{Resp}(\mathsf{PRF.Eval}(k^*, x\|1), r_{\Sigma,1}))\end{aligned}$$

  and thus the two hybrids are statistically close.
- Hybrid $\mathcal{H}_3$: In this hybrid we compute $\beta_1 = \mathsf{PRF.Eval}(k^*, x\|1)$ and we use the challenge to simulate the response for the sigma protocol. I.e. we compute $(|\alpha_1\rangle, \gamma_1) \leftarrow\$ \Sigma.\mathsf{Sim}(x, \beta_1)$ and we set $c_{\gamma,1} \leftarrow\$ \mathsf{FHE.Sim}(1^\kappa, \mathsf{pk}, c_k, \gamma_1)$. Note that the only difference with respect to the previous hybrid is that we do compute a simulated transcript of the sigma protocol instead of an honest one. By the statistical special honest-verifier zero-knowledge property of the sigma protocol we have that

  $$(\Sigma.\mathsf{Com}(|w_0\rangle^{\otimes p(\lambda)}; r_{\Sigma,1}), \Sigma.\mathsf{Resp}(\beta_1, r_{\Sigma,1})) \approx_s \Sigma.\mathsf{Sim}(x, \beta_1)$$

  and therefore the two hybrids are statistically close.
- Hybrid $\mathcal{H}_4$: In this hybrid we switch the computation of $|\alpha_1\rangle$ and $c_{\gamma,1}$ to use again an honest witness, except that we use $|w_1\rangle$ instead of $|w_0\rangle$. Specifically we compute the commitment of the sigma protocol as $|\alpha_1\rangle \leftarrow\$ \Sigma.\mathsf{Com}(|w_1\rangle^{\otimes p(\lambda)}; r_{\Sigma,1})$ and the simulated ciphertext as $c_{\gamma,1} \leftarrow\$ \mathsf{FHE.Sim}(1^\kappa, \mathsf{pk}, c_k, \Sigma.\mathsf{Resp}(\mathsf{PRF.Eval}(k^*, x\|1), r_{\Sigma,1}))$. The two hybrids are statistically indistinguishable by the statistical special honest-verifier zero-knowledge property of the sigma protocol (same argument as $\mathcal{H}_2 \approx_s \mathcal{H}_3$).
- Hybrid $\mathcal{H}_5$: In this hybrid we switch back to a correctly evaluated FHE ciphertext, i.e. we compute $c_{\gamma_1} \leftarrow\$ \mathsf{FHE.Eval}(\mathsf{pk}, \Sigma.\mathsf{Resp}(\mathsf{PRF.Eval}(\cdot, x\|1), r_{\Sigma,1}), c_k)$. By the (malicious) statistical circuit privacy of the FHE scheme, the two hybrids are statistically close (same argument as $\mathcal{H}_1 \approx_s \mathcal{H}_2$).
- Hybrid $\mathcal{H}_6$: In this hybrid we revert the changes to the SBSH commitment, i.e. we compute $c_{\tau,1} \leftarrow\$ \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), r_{\Sigma,1})$. By the statistical hiding of the SBSH commitment we have that the two hybrids are statistically indistinguishable (same argument as $\mathcal{H}_0 \approx_s \mathcal{H}_1$).

- Hybrid $\mathcal{H}_7$: This hybrid is identical to the previous one, except that we compute the statistical ZAP argument using $(1, r_{\Sigma,1}, r_{\mathsf{FHE},1}, r_{\mathsf{SBSH},1})$. Note that the the messages are indeed well-formed and thus statistical indistinguishability follows by the statistical witness indistinguishability of the ZAP argument system.
- Hybrids $\mathcal{H}_8 \dots \mathcal{H}_{13}$: In this series of hybrids we change how we compute $(|\alpha_0\rangle, c_{\gamma,0}, c_{\tau,0})$ analogously as we did in hybrids $\mathcal{H}_1 \dots \mathcal{H}_6$, i.e. using $|w_1\rangle$ instead of $|w_0\rangle$. Note that the underlying random coins are no longer used in the computation of the ZAP argument and thus to indistinguishability follows along the same lines as what we discussed above.

Observe that hybrid $\mathcal{H}_{13}$ is identical to $\mathcal{H}_0$ except that the challenge bit is fixed to $b = 1$ and in particular the oracle uses the witness $|w_1\rangle$ to compute the ZAP argument. It follows that our protocol satisfies statistical witness indistinguishability. □

5 Zero-Knowledge for QMA

In the following we present a 4-Round statistical zero-knowlege protocol for QMA. Before delving into the description of our protocol, we introduce a few cryptographic tools that are going to be useful for our main protocol.

5.1 Sometimes-Extractable SRP Oblivious Transfer

Oblivious transfer allows one to condition the transfer of two messages (m_0, m_1) on some secret choice bit b. In this work we are interested in 3-round statistically receiver private (SRP) protocols, where the security of the choice bit is protected statistically [GJJM20]. We recall the syntax of OT in the following. For simplicity we only define the single-bit variant and the multi-bit variant follows as a natural extension.

Definition 5.1 (3-Round SRP-OT)**.** *An SRP-OT protocol* $(\mathsf{OT.Setup}, \mathsf{OT.Rec}, \mathsf{OT.Send}, \mathsf{OT.Dec})$ *consists of the following efficient algorithms.*

- $\mathsf{OT.Setup}(1^\lambda)$*: On input the security parameter* 1^λ*, the setup returns a first message* ot_0 *, as well as a state* st_S *to the sender.*
- $\mathsf{OT.Rec}(\mathsf{ot}_0, b)$*: On input a first message* ct_0 *and a choice bit* b*, the receiver algorithm returns a second message* ot_1 *and a key* k*.*
- $\mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, (m_0, m_1))$*: On input a second message* ot_1*, the state* st_S *and a pair of messages* (m_0, m_1)*, the sender algorithm returns a third message* ot_2*.*
- $\mathsf{OT.Dec}(\mathsf{ot}_0, \mathsf{ot}_2, k)$*: On input a first message* ot_0*, a third message* ot_2 *and a key* k*, the decryption algorithm returns a message* m

We define completeness.

Definition 5.2 (Completeness)**.** *An SRP-OT protocol* $(\mathsf{OT.Setup}, \mathsf{OT.Rec}, \mathsf{OT.Send}, \mathsf{OT.Dec})$ *is complete if for all* $\lambda \in \mathbb{N}$*, all* $b \in \{0,1\}$*, and all messages* (m_0, m_1) *it holds that*

$$\Pr\left[\mathsf{OT.Dec}(\mathsf{ot}_0, \mathsf{ot}_2, \mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, (m_0, m_1))) = m_b\right] = 1.$$

where $(\mathsf{ot}_0, \mathsf{st}_S) \leftarrow\!\$\, \mathsf{OT.Setup}(1^\lambda)$ *and* $(\mathsf{ot}_1, k) \leftarrow\!\$\, \mathsf{OT.Rec}(\mathsf{ot}_0, b)$*.*

Statistical receiver privacy requires that the choice bit is hidden in a statistical sense, for any choice of the first message.

Definition 5.3 (Statistical Receiver Privacy)**.** *An SRP-OT protocol* $(\mathsf{OT.Setup}, \mathsf{OT.Rec}, \mathsf{OT.Send}, \mathsf{OT.Dec})$ *is statistically receiver private if for all* $\lambda \in \mathbb{N}$ *and all first messages* ot_0 *the following distributions are statistically indistinguishable*

$$(\mathsf{ot}_0, c_0) \approx_s (\mathsf{ot}_0, c_1)$$

where $(c_0, k_0) \leftarrow\!\!\$\, \mathsf{OT.Rec}(\mathsf{ot}_0, 0)$ *and* $(c_1, k_1) \leftarrow\!\!\$\, \mathsf{OT.Rec}(\mathsf{ot}_0, 1)$.

We define the notion of computational sender privacy for SRP-OT as stated in [GJJM20, ACP20].

Definition 5.4 (Computational Sender Privacy)**.** *An SRP-OT protocol* $(\mathsf{OT.Setup}, \mathsf{OT.Rec}, \mathsf{OT.Send}, \mathsf{OT.Dec})$ *is computationally sender private if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$*, and all non-uniform QPT receivers with quantum advice* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\mathbb{E}\left[\mathsf{min}\{p_0, p_1\}\right] \leq \mu(\lambda)$$

where

$$p_0 = \left| \Pr\left[\mathcal{A}(\mathsf{OT.Send}(ot_1, \mathsf{st}_S, (m_{b_0}, m_1)), \mathsf{st}_\mathsf{R}; \rho) = b_0 \;\middle|\; \begin{array}{l} (\mathsf{ot}_0, \mathsf{st}_S) \leftarrow\!\!\$\, \mathsf{OT.Setup}(1^\lambda) \\ b_0 \leftarrow\!\!\$\, \{0,1\} \\ (\mathsf{st}_\mathsf{R}, \mathsf{ot}_1) = \mathcal{A}(\mathsf{ot}_0; \rho) \end{array} \right] - \frac{1}{2} \right|$$

and

$$p_1 = \left| \Pr\left[\mathcal{A}(\mathsf{OT.Send}(ot_1, \mathsf{st}_S, (m_0, m_{b_1})), \mathsf{st}_\mathsf{R}; \rho) = b_1 \;\middle|\; \begin{array}{l} (\mathsf{ot}_0, \mathsf{st}_S) \leftarrow\!\!\$\, \mathsf{OT.Setup}(1^\lambda) \\ b_1 \leftarrow\!\!\$\, \{0,1\} \\ (\mathsf{st}_\mathsf{R}, \mathsf{ot}_1) = \mathcal{A}(\mathsf{ot}_0; \rho) \end{array} \right] - \frac{1}{2} \right|.$$

Such a scheme can be constructed assuming the quantum hardness of the LWE problem.

Lemma 5.5 ([GJJM20])**.** *Assuming the post-quantum hardness of the LWE problem, there exists an SRP-OT scheme* $(\mathsf{OT.Setup}, \mathsf{OT.Rec}, \mathsf{OT.Send}, \mathsf{OT.Dec})$.

PKE with Certifiable Keys. Before describing our scheme, we need to introduce an additional building block: A public-key encryption (PKE) with certifiable keys. This is a standard PKE scheme $(\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ that, in addition to the canonical notion of correctness and semantic security, satisfies the following additional constraint: There exists an algorithm $\mathsf{PKE.Verify}$ and a polynomial $\ell(\lambda)$ such that:

(1) For all $\lambda \in \mathbb{N}$ and all $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{PKE.Gen}(1^\lambda)$ there does not exist any $w \in \{0,1\}^{\ell(\lambda)}$ such that $\mathsf{PKE.Verify}(1^\lambda, \mathsf{pk}, w) = 1$.

(2) For all $\lambda \in \mathbb{N}$ and all (possibly malformed) pk, if there does not exist any $w \in \{0,1\}^{\ell(\lambda)}$ such that $\mathsf{PKE.Verify}(1^\lambda, \mathsf{pk}, w) = 1$, then there exist an sk such that for all messages m it holds that

$$\Pr\left[\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m)) = m\right] = 1.$$

The work of [KNYY21] shows how to construct a post-quantum PKE with certifiable keys.

Lemma 5.6 ([KNYY21])**.** *Assuming the post-quantum hardness of the LWE problem, there exists an PKE* $(\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ *with cerifiable keys.*

Sometimes-Extractable SRP-OT. Next we define and construct an enhanced version of the above OT, namely sometimes-extractable (SE) SRP-OT. For convenience we define directly the multi-bit variant, where the receiver has a set of choice bits $(b_1, \ldots, b_n)$. The OT is required to be statistically receiver private, except with some small probability where $(\mathsf{ot}_0, \mathsf{ot}_1)$ is a perfectly binding commitment to $(b_1, \ldots, b_n)$. We define the set $\mathsf{Binding}$ as the set of all messages $(\mathsf{ot}_0, \mathsf{ot}_1)$ that uniquely determine the choice bits of the receiver.

Definition 5.7 (Sometimes Extractability)**.** *An SRP-OT* $(\mathsf{OT.Setup}, \mathsf{OT.Rec}, \mathsf{OT.Send}, \mathsf{OT.Dec})$ *is* (ε, δ)*-sometimes extractable if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$ *and all (stateful) QPT distinguishers* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\Pr\left[\mathcal{A}(\mathsf{st}_\mathsf{R}; \rho) = 1 \wedge (\mathsf{ot}_0, \mathsf{ot}_1) \in \mathsf{Binding}\right] = \varepsilon(\lambda) \cdot \Pr\left[\mathcal{A}(\mathsf{st}_\mathsf{R}; \rho) = 1\right] + \delta(\lambda) \cdot \mu(\lambda)$$

where $(\mathsf{ot}_0, \mathsf{st}_S) \leftarrow\!\!\$\ \mathsf{OT.Setup}(1^\lambda)$ *and* $(\mathsf{st}_\mathsf{R}, \mathsf{ot}_1) = \mathcal{A}(\mathsf{ot}_0; \rho)$*. Furthermore, we require the existence of a polynomial-time algorithm* $\mathsf{OT.Ext}$ *that, on input the random coins* r *used in the* $\mathsf{OT.Setup}$ *algorithm, outputs a set of choice bits* $(b_1, \ldots, b_n)$ *from the protocol transcript if* $(\mathsf{ot}_0, \mathsf{ot}_1) \in \mathsf{Binding}$ *such that*

$$\mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, (m_{0,1}, m_{1,1}, \ldots, m_{0,n}, m_{1,n})) \approx_c \mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, (m_{b_1,1}, m_{b_1,1}, \ldots, m_{b_n,n}, m_{b_n,n})).$$

Our Protocol. Let $\varepsilon(\lambda)$ be a (fixed) negligible function. We assume the existence of the following building blocks (all secure against quantum adversaries):

- A 2-round WI argument $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ for NP with statistical witness indistinguishability and $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ soundness error.
- A 3-Round SRP-OT $(\mathsf{OT.Setup}, \mathsf{OT.Rec}, \mathsf{OT.Send}, \mathsf{OT.Dec})$ with computational sender privacy's advantage bounded by $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$.
- A PKE $(\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ with certifiable keys an $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ security.
- An SBSH commitment scheme $(\mathsf{SBSH.Gen}, \mathsf{SBSH.Key}, \mathsf{SBSH.Com})$ that satisfies $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes binding.

Where $\mu(\lambda)$ is some negligible function and κ is the security parameter of the primitives with super-polynomially bounded disitinguishing advantage. Our protocol is shown in Figure 3. In a slight abuse of notation, we let the OT protocol handle n simultaneous choice bits.

Statistical Receiver Privacy. We now show that the protocol still satisfies statistical receiver privacy.

Theorem 5.8 (Statistical Receiver Privacy)**.** *The protocol described in Figure 3 satisfies statistical receiver privacy.*

Proof. We consider the following series of hybrid distributions.

- Hybrid $\mathcal{H}_0$: This is the original receiver message.
- Hybrid $\mathcal{H}_1$: Here we (inefficiently) extract a secret key sk or a non-membership witness w for pk. Note that, by the certifiability of PKE, either of them must exist. We then compute $\mathsf{cmt}_\mathsf{sk} \leftarrow\!\!\$\ \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), \{\mathsf{sk}, w\})$. Statistical indistinguishability follows from the statistical hiding of the SBSH commitment.

Sometimes-Extractable SRP-OT

- **OT Setup:** The setup algorithm samples an SBSH commitment key $\mathsf{ck}_0 \leftarrow\$ \mathsf{SBSH.Gen}(1^\lambda)$ and n copies of the OT setup $(\mathsf{ot}_{0,1}, \mathsf{st}_{S,1}, \ldots, \mathsf{ot}_{0,n}, \mathsf{st}_{S,n}) \leftarrow\$ \mathsf{OT.Setup}(1^\kappa)$. It additionally samples a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow\$ \mathsf{PKE.Gen}(1^\kappa)$ and the first message of a WI argument $(\mathsf{crs}, \mathsf{td}) \leftarrow\$ \mathsf{WI.Setup}(1^\kappa)$. It returns $(\mathsf{ck}_0, \mathsf{ot}_{0,1}, \ldots, \mathsf{ot}_{0,n}, \mathsf{pk}, \mathsf{crs})$ as the first message.
- **OT Receive:** Compute $(\mathsf{ot}_{1,1}, k_1, \ldots, \mathsf{ot}_{1,n}, k_n) \leftarrow\$ \mathsf{OT.Rec}((\mathsf{ot}_{0,1}, \ldots, \mathsf{ot}_{0,n}), (b_1, \ldots, b_n))$. Then sample $\mathsf{ck}_1 \leftarrow\$ \mathsf{SBSH.Key}(\mathsf{ck}_0)$ and compute

  $$\mathsf{cmt}_k \leftarrow\$ \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), (b_1, k_1, \ldots, b_n, k_n)) \text{ and } \mathsf{cmt}_{\mathsf{sk}} \leftarrow\$ \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), 0).$$

  Compute a WI proof π for the statement

  $$\mathsf{stmt} = \left\{ \begin{array}{l} \mathsf{cmt}_k \in \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), (b_1, k_1, \ldots, b_n, k_n)) \\ \wedge\ (\mathsf{ot}_{1,1}, k_1, \ldots, \mathsf{ot}_{1,n}, k_n) \in \mathsf{OT.Rec}((\mathsf{ot}_{0,1}, \ldots, \mathsf{ot}_{0,n}), (b_1, \ldots, b_n)) \\ \vee\ \mathsf{cmt}_{\mathsf{sk}} \in \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), \mathsf{sk}) \wedge (\mathsf{pk}, \mathsf{sk}) \in \mathsf{PKE.Gen}(1^\kappa) \\ \vee\ \mathsf{cmt}_{\mathsf{sk}} \in \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), w) \wedge \mathsf{PKE.Verify}(1^\kappa, \mathsf{pk}, w) = 1 \end{array} \right\}$$

  using the witness of the first branch, then return $(\mathsf{ot}_{1,1}, \ldots, \mathsf{ot}_{1,n}, \mathsf{cmt}_k, \mathsf{cmt}_{\mathsf{sk}}, \pi)$.
- **OT Send:** Check that $\mathsf{WI.Verify}(\mathsf{crs}, \mathsf{stmt}, \pi) = 1$, then send

  $$(\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,n}) \leftarrow\$ \mathsf{OT.Send}((\mathsf{ot}_{1,1}, \mathsf{st}_{S,1}, \ldots, \mathsf{ot}_{1,n}, \mathsf{st}_{S,n}), (m_{0,1}, m_{1,1}, \ldots, m_{0,n}, m_{1,n})).$$

- **OT Decrypt:** Return $\mathsf{OT.Dec}(((\mathsf{ot}_{0,1}, \ldots, \mathsf{ot}_{0,n})), (\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,n}), (k_1, \ldots, k_n))$.

Figure 3: Description of our SESRP-OT protocol

- Hybrid $\mathcal{H}_2$: We compute the proof π using the witness of the second or third branch, depending on whether we extracted sk or w from pk. This distribution is statistically close to the previous one, by the statistical witness indistinguishability of the WI argument.
- Hybrid $\mathcal{H}_3$: We compute $\mathsf{cmt}_k \leftarrow\$ \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), 0)$. Statistical indistinguishability follows from the statistical hiding of the SBSH commitment.
- Hybrid $\mathcal{H}_4$: We compute $(\mathsf{ot}_{1,1}, k_1, \ldots, \mathsf{ot}_{1,n}, k_n)$ via $\mathsf{OT.Rec}((\mathsf{ot}_{0,1}, \ldots, \mathsf{ot}_{0,n}), (0, \ldots, 0))$. Statistical indistinguishability follows by an invocation of the statistical receiver privacy of the SRP-OT.

The proof is concluded by observing that in the last distribution no information about the choice bits is present in the receiver's message. □

Sometimes Extractability. We show that the protocol is sometimes extractable.

Theorem 5.9 (Sometimes Extractability)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem, the protocol described in Figure 8 is $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes extractable.*

Proof. It is easy to see that if $(\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}$ then it also holds that $(\mathsf{ot}_0, \mathsf{ot}_1) \in \mathsf{Binding}$ and therefore it holds that

$$\Pr\left[\mathcal{A}(\mathsf{st}_\mathsf{R}; \rho) = 1 \wedge (\mathsf{ot}_0, \mathsf{ot}_1) \in \mathsf{Binding}\right] = \Pr\left[\mathcal{A}(\mathsf{st}_\mathsf{R}; \rho) = 1 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right].$$

Thus all that is left to be shown is to define an extractor $\mathsf{OT.Ext}$ that outputs the correct set of choice bits $(b_1, \ldots, b_n)$. The extractor runs $\mathsf{SBSH.Ext}$ on cmt_k to obtain a set $(b_1, k_1, \ldots, b_n, k_n)$ and returns $(b_1, \ldots, b_n)$. Conditioned on the SBSH-commitment being in $\mathsf{Binding}$ mode, we want to bound the probability that the event Fail happens where

$$\begin{aligned}
&\Pr[\mathsf{Fail} \mid (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}] \\
&= \Pr[(\mathsf{ot}_{1,1}, k_1, \ldots, \mathsf{ot}_{1,n}, k_n) \notin \mathsf{OT.Rec}((\mathsf{ot}_{0,1}, \ldots, \mathsf{ot}_{0,n}), (b_1, \ldots, b_n))] \\
&\leq \varepsilon(\lambda).
\end{aligned}$$

Assume towards contradiction that the opposite is true, then since the SBSH commitment is sometimes binding we have that

$$\Pr[\mathsf{Fail} \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda)).$$

for some negligible function $\mu(\lambda)$. Therefore it must be the cast that either

- $\mathsf{cmt}_\mathsf{sk} \in \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), \mathsf{sk})$, where sk is a valid secret key for pk, or
- π is a proof for a false statement.

The former is a contradiction to the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ security of PKE, whereas the latter contradicts the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ soundness of the WI argument. This establishes that, conditioned on the commitment being in $\mathsf{Binding}$ mode, we have that

$$\Pr[(\mathsf{ot}_{1,1}, k_1, \ldots, \mathsf{ot}_{1,n}, k_n) \in \mathsf{OT.Rec}((\mathsf{ot}_{0,1}, \ldots, \mathsf{ot}_{0,n}), (b_1, \ldots, b_n)) \mid (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}] \geq \varepsilon(\lambda).$$

By another invocation of the sometimes binding property of the SBSH commitment we obtain that

$$\begin{aligned}
&\Pr[(\mathsf{ot}_{1,1}, k_1, \ldots, \mathsf{ot}_{1,n}, k_n) \in \mathsf{OT.Rec}((\mathsf{ot}_{0,1}, \ldots, \mathsf{ot}_{0,n}), (b_1, \ldots, b_n)) \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}] \\
&\geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda)).
\end{aligned}$$

By the correctness of the SRP-OT we have that $(k_1, \ldots, k_n)$ allows anyone to recover $(m_{b_1,1}, \ldots, m_{b_n,n})$, and in particular to win the experiment for computational sender privacy with probability $p_{b_i} = 1$. Thus, for all $i \in \{1, \ldots, n\}$ it holds that

$$\mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, (m_{0,i}, m_{1,i})) \approx_c \mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, (m_{b_i,i}, m_{b_i,i}))$$

by the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ sender privacy of the SRP-OT. The indistinguishability of the views follows by a hybrid argument. □

5.2 Post-Quantum Conditional Disclosure of Secrets

Conditional disclosure of secrets (CDS) [AIR01] for a language $\mathcal{L}$ in NP with relation $\mathsf{R}_\mathcal{L}$ is the interactive analogue of witness encryption [GGSW13]: Given a statement x and a message m from the sender, the receiver is able to recover m if $x \in \mathcal{L}$, whereas m stays hidden if this is not the case. Furthermore, the witness w for x should be kept secret from the eyes of the sender.

Definition. We recall the definition of a CDS protocol. In this work we consider two variants: A 3-round statistically-receiver private (SRP) CDS and a 2-round statistically sender private (SSP) CDS. The syntax below is defined for the 3-round variant and the 2-round protocol can be defined analogously by omitting the first algorithm.

Definition 5.10 (CDS Protocol for NP)**.** *A CDS protocol* $(\mathsf{Setup}, \mathsf{R}, \mathsf{S}, \mathsf{D})$ *for a language* $\mathcal{L} \in \mathsf{NP}$ *with relation* $\mathsf{R}_{\mathcal{L}}$ *consists of the following efficient algorithms.*

- $\mathsf{Setup}(1^\lambda)$*: On input the security parameter* 1^λ*, the setup returns a first message* ct_0 *and a state* $\mathsf{st}_{\mathsf{cds}}$*.*
- $\mathsf{R}(\mathsf{ct}_0, w)$*: On input a first message* ct_0 *and a witness* w*, the receiver algorithm returns a second message* ct_1 *and a key* k*.*
- $\mathsf{S}(\mathsf{ct}_1, \mathsf{st}_{\mathsf{cds}}, x, m)$*: On input a second message* ct_1*, a state* $\mathsf{st}_{\mathsf{cds}}$*, a statement* x*, and a message* m*, the sender algorithm returns a third message* ct_2*.*
- $\mathsf{D}(\mathsf{ct}_2, k)$*: On input a third message* ct_2 *and a key* k*, the decryption algorithm returns a message* m

We define completeness for a CDS protocol.

Definition 5.11 (Completeness)**.** *A CDS protocol* $(\mathsf{Setup}, \mathsf{R}, \mathsf{S}, \mathsf{D})$ *for a language* $\mathcal{L} \in \mathsf{NP}$ *with relation* $\mathsf{R}_{\mathcal{L}}$ *is complete if for all* $\lambda \in \mathbb{N}$*, all* $x \in \mathcal{L}$*, all* $x \in \mathsf{R}_{\mathcal{L}}(x)$*, and all messages* m *it holds that*

$$\Pr\left[\mathsf{D}(\mathsf{S}(\mathsf{ct}_1, \mathsf{st}_{\mathsf{cds}}, x, m), k) = m\right] = 1.$$

where $(\mathsf{ct}_0, \mathsf{st}_{\mathsf{cds}}) \leftarrow\!\$\, \mathsf{Setup}(1^\lambda)$ *and* $(\mathsf{ct}_1, k) \leftarrow\!\$\, \mathsf{R}(\mathsf{ct}_0, w)$*.*

Next we define the notion of (computational and statistical) receiver privacy.

Definition 5.12 (Receiver Privacy)**.** *A CDS protocol* $(\mathsf{Setup}, \mathsf{R}, \mathsf{S}, \mathsf{D})$ *for a language* $\mathcal{L} \in \mathsf{NP}$ *with relation* $\mathsf{R}_{\mathcal{L}}$ *is computationally (statistically, resp.) receiver private if for all* $\lambda \in \mathbb{N}$*, all strings* w*, and all first messages* ct_0 *the following distributions are computationally (statistically, resp.) indistinguishable*

$$(\mathsf{ct}_0, c_0) \approx (\mathsf{ct}_0, c_1)$$

where $(c_0, k_0) \leftarrow\!\$\, \mathsf{R}(\mathsf{ct}_0, 0)$ *and* $(c_1, k_1) \leftarrow\!\$\, \mathsf{R}(\mathsf{ct}_0, 1)$*.*

Finally we define the notion of (computational and statistical) sender privacy.

Definition 5.13 (Sender Privacy)**.** *A CDS protocol* $(\mathsf{Setup}, \mathsf{R}, \mathsf{S}, \mathsf{D})$ *for a language* $\mathcal{L} \in \mathsf{NP}$ *with relation* $\mathsf{R}_{\mathcal{L}}$ *is computationally (statistically, resp.) sender private if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$*, all* $x \notin \mathcal{L}$*, and all non-uniform QPT (unbounded, resp.) receivers with quantum advice* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$*, it holds that*

$$\left|\Pr\left[\mathcal{A}(\mathsf{S}(\mathsf{ct}_1, \mathsf{st}_{\mathsf{cds}}, x, m), \mathsf{st}; \rho) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{S}(\mathsf{ct}_1, \mathsf{st}_{\mathsf{cds}}, x, 0), \mathsf{st}; \rho) = 1\right]\right| \leq \mu(\lambda).$$

where $(\mathsf{ct}_0, \mathsf{st}_{\mathsf{cds}}) \leftarrow\!\$\, \mathsf{Setup}(1^\lambda)$ *and* $(\mathsf{st}, \mathsf{ct}_1) = \mathcal{A}(\mathsf{ct}_0; \rho)$*.*

It is well-known that a 2-round SSP-CDS can be built from any 2-round oblivious transfer and information-theoretically secure randomized encodings [IK00]. Thus we have the following fact.

Lemma 5.14 ([BD18])**.** *Assuming the post-quantum hardness of the LWE problem, there exists an SSP-CDS scheme* $(\mathsf{R}, \mathsf{S}, \mathsf{D})$ *with computational receiver privacy and statistical sender privacy.*

Post-Quantum SRP-CDS. We give a simple construction of post-quantum SRP-CDS for NP assuming an $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes extractable SRP-OT and a simulation secure garbling scheme $(\mathsf{Garble}, \mathsf{GEval})$ for NC1 circuits.[4] The scheme is given in Figure 4 and next we prove that it satisfies the definition of an SRP-CDS.

Statistically Receiver Private CDS

- **CDS Setup:** The setup algorithm samples the first message of an n-bit SESRP-OT $(\mathsf{ot}_0, \mathsf{st}_S) \leftarrow\$ \mathsf{OT.Setup}(1^\lambda)$.
- **CDS Receive:** Compute $(\mathsf{ot}_1, k) \leftarrow\$ \mathsf{OT.Rec}(\mathsf{ot}_0, (w_1, \ldots, w_n))$ where $(w_1, \ldots, w_n)$ is the bit decomposition of the witness w.
- **CDS Send:** Let Φ be the circuit that, on input a witness w, checks whether $w \in \mathsf{R}_{\mathcal{L}}(x)$ and returns the message m if that is the cases and 0 otherwise. Compute

$$\left(\widetilde{\Phi}, \{\ell_{i,b}\}_{i\in\{1,\ldots,n\},b\in\{0,1\}}\right) \leftarrow\$ \mathsf{Garble}\left(1^\lambda, \Phi\right)$$

  and return $\widetilde{\Phi}$ and $\mathsf{ot}_2 \leftarrow\$ \mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, , \{\ell_{i,b}\}_{i\in\{1,\ldots,n\},b\in\{0,1\}})$.
- **CDS Decrypt:** Compute $(\ell_1, \ldots, \ell_n) = \mathsf{OT.Dec}(\mathsf{ot}_0, \mathsf{ot}_2, k)$ and return $\mathsf{GEval}(\widetilde{\Phi}, (\ell_1, \ldots, \ell_n))$.

Figure 4: Description of our SRP-CDS protocol

Theorem 5.15. *Assuming the quantum quasi-polynomial hardness of the LWE problem, the protocol in Figure 4 satisfies statistical receiver privacy and computational sender privacy.*

Proof. Statistical receiver privacy follows immediately from the receiver privacy of the SESRP-OT. For sender privacy, by the sometimes extractablility of SESRP-OT, there exist an extractor $\mathsf{OT.Ext}$ that returns the choice bits $(b_1, \ldots, b_n)$ of the receiver. Note that $(b_1, \ldots, b_n)$ cannot encode a valid witness since $x \notin \mathcal{L}$. We can then run the simulator $(\widetilde{\Phi}, (\ell_1, \ldots, \ell_n)) \leftarrow\$ \mathsf{GSim}\left(1^\lambda, 1^{|\Phi|}, 1^n, 0\right)$ and return $(\widetilde{\Phi}, \mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, , (\ell_1, \ell_1, \ldots, \ell_n, \ell_n)))$ to the attacker. Computational indistinguishability follows from a reduction to the computational sender privacy of the SESRP-OT. □

5.3 Sometimes-Simulatable Zero-Knowledge

We construct an interactive delayed input (3-round) ZK proof system for NP that satisfies statistical soundness and the notion of sometimes simulatability (SSim-ZK). This can be thought as the straight-line equivalent of super-polynomial simulation [Pas03b] and it is formally defined in the following.

[4] Note that NC1 circuits suffice here, since it is well known that the validity of any NP statement can be verified by an NC1 circuit.

Definition 5.16 (Sometimes Simulatability)**.** *An interactive protocol* (P, V) *for a language* $\mathcal{L} \in \mathsf{NP}$ *with relation* $\mathsf{R}_{\mathcal{L}}$ *is* (ε, δ)*-sometimes simulatable if there exists a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$ *and all (stateful) QPT distinguishers* $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$*, it holds that*

$$\Pr\left[\mathcal{A}(\mathsf{st}; \rho) = 1 \wedge (\mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}\right] = \varepsilon(\lambda) \cdot \Pr\left[\mathcal{A}(\mathsf{st}; \rho) = 1\right] + \delta(\lambda) \cdot \mu(\lambda)$$

where $(\mathsf{zk}_0, \mathsf{zk}_1)$ *are the first two messages of the protocol and* $\mathsf{Simulation}$ *defines a set. Furthermore, we require the existence of a polynomial-time algorithm* Sim *such that, conditioned on the event* $(\mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}$*, it holds that*

$$\mathsf{Sim}(1^\lambda, r, x) \approx_c \mathsf{zk}_2$$

where r *are the random coins* r *used to compute* zk_0 *and* zk_2 *is the honestly computed third message.*

Let $\varepsilon(\lambda)$ be a (fixed) negligible function. We build our protocol assuming the existence of the following primitives:

- A 3-Round $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes extractable SRP-OT $(\mathsf{OT.Setup}, \mathsf{OT.Rec}, \mathsf{OT.Send}, \mathsf{OT.Dec})$.
- A 3-Round Sigma protocol $(a, b, (c_0, c_1))$ for NP with binary challenge, $1/2$ soundness gap against an unbounded prover, and that satisfies $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-computational zero-knowledge. An example of such protocols is Blum's protocol for Graph Hamiltonicity.
- A perfectly binding commitment Com that satisfies $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-computational hiding.

Where $\mu(\lambda)$ is some negligible function and κ is the security parameter of the primitives with super-polynomially bounded disitinguishing advantage. Our protocol is shown in Figure 5.

Sometimes-Simulatable ZK Proof

- **First Message (P $\to$ V):** Sample the first message of an n-bit SESRP-OT $(\mathsf{ot}_0, \mathsf{st}_S) \leftarrow\!\$\, \mathsf{OT.Setup}(1^\lambda)$.
- **Second Message (V $\to$ P):** Sample a set of n challenges $(b_1, \ldots, b_n) \leftarrow\!\$\, \{0,1\}^n$ and compute $(\mathsf{ot}_1, k) \leftarrow\!\$\, \mathsf{OT.Rec}(\mathsf{ot}_1, (b_1, \ldots, b_n)$. Send ot_1 to the prover.
- **Third Message (P $\to$ V):** Sample n independent commitment-response tuples for the sigma protocol $(a_1, c_{0,1}, c_{1,1}, \ldots, a_n, c_{0,n}, c_{1,n})$. Then for all $i \in \{1, \ldots, n\}$ and all $b \in \{0,1\}$ compute $\mathsf{cmt}_{b,i} = \mathsf{Com}(1^\kappa, c_{b,i}; r_{b,i})$, for some uniformly sampled $r_{i,b}$, and define $d_{b,i} = (c_{b,i}, r_{b,i})$. Return

  $$(a_1, \ldots, a_n, \mathsf{cmt}_{0,1}, \mathsf{cmt}_{1,1}, \ldots, \mathsf{cmt}_{0,n}, \mathsf{cmt}_{1,n}, \mathsf{OT.Send}(\mathsf{ot}_1, \mathsf{st}_S, (d_{0,1}, d_{1,1}), \ldots, (d_{0,n}, d_{1,n}))).$$

- **Verify:** Compute $(d_1, \ldots, d_n) = \mathsf{OT.Dec}(\mathsf{ot}_0, \mathsf{ot}_2, k)$ then for all $i \in \{1, \ldots, n\}$ check whether $d_i = (c_i, r_i)$ is a valid opening for $\mathsf{cmt}_{b_i,i}$ and that (a_i, b_i, c_i) is an accepting transcript for the sigma protocol.

Figure 5: Description of our sometimes-simulatable ZK protocol

Sometimes-Simulatability. We show that our protocol satisfies the notion of $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes simulatability.

Theorem 5.17. *Assuming the quantum quasi-polynomial hardness of the LWE problem, the protocol in Figure 5 satisfies $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes simulatability.*

Proof. We define the set $\mathsf{Simulation}$ to be the set of first and second messages $(\mathsf{zk}_0, \mathsf{zk}_1)$ such that $(\mathsf{ot}_0, \mathsf{ot}_1) \in \mathsf{Binding}$. We then define the simulator to run the extractor $\mathsf{OT.Ext}$ and obtain the challenge bits $(b_1, \ldots, b_n)$. The simulator then computes a set of simulated transcripts $(a_1, b_1, c_1), \ldots, (a_n, b_n, c_n)$ and sets $\mathsf{cmt}_{b_i \oplus 1, i} \leftarrow\!\!\$\, \mathsf{Com}(1^\lambda, 0)$ whereas $\mathsf{cmt}_{b_i, i}$ are computed honestly. It then computes the ot_2 message setting each message pair to $(d_i, d_i) = ((c_i, r_i), (c_i, r_i))$. Computational indistinguishability can be shown by a standard hybrid argument against the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-hiding of the commitment scheme and the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-computational zero-knowledge of the sigma protocol. □

Soundness. We show that our protocol satisfies statistical soundness.

Theorem 5.18. *The protocol in Figure 5 is statistically sound.*

Proof. Consider a modified verifier that samples a uniform challenge $(b_1, \ldots, b_n)$ but computes the SRP-OT receiver message to some fixed string 0^λ. It then (inefficiently) extracts c_{b_i} from the commitments and verifies the validity of all n transcripts. By the statistical receiver privacy of the SRP-OT, the accepting probability of such verifier is statistically close to that of the original one. Note that such a verifier does not reveal any information about the challenge $(b_1, \ldots, b_n)$ and therefore the success probability of the (possibly unbounded) prover is negligibly close to $1/2^n$, by the statistical soundness of the sigma protocol. □

5.4 4-Round Zero-Knowledge for QMA

We assume the existence of the following building blocks (all secure against quantum adversaries):

- A circuit-private classical QFHE scheme $(\mathsf{QFHE.Gen}, \mathsf{QFHE.Enc}, \mathsf{QFHE.Eval}, \mathsf{QFHE.Dec})$ with distinguishing advantage $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$.
- A non-interactive perfectly binding commitment Com with hiding advantage $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$.
- An SBSH commitment scheme $(\mathsf{SBSH.Gen}, \mathsf{SBSH.Key}, \mathsf{SBSH.Com})$ that satisfies $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes binding.
- A 2-round WI argument $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ for QMA, with statistical witness indistinguishability and $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ soundness error.
- a 3-round statistically receiver private conditional disclosure of secrets scheme $(\mathsf{SRP\text{-}CDS.Setup}, \mathsf{SRP\text{-}CDS.R}, \mathsf{SRP\text{-}CDS.S}, \mathsf{SRP\text{-}CDS.D})$ for NP, with $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ computational sender privacy.
- A CC obfuscator Obf with $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ simulatability.
- A 3-round sometimes simulatable statistical ZK protocol $(\mathsf{SSim\text{-}ZK.Setup}, \mathsf{SSim\text{-}ZK.R}, \mathsf{SSim\text{-}ZK.S})$ that satisfies $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes simulatability.

Our protocol is formally described in Figure 6.

4-Round (Statistical) ZK Argument for QMA

- **First Message (V $\to$ P):** The verifier samples $(\mathsf{td}, r, s) \leftarrow\!\$\, \{0,1\}^\kappa$ and computes:
  - $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\, \mathsf{QFHE.Gen}(1^\kappa; r)$ and $c_{td} = \mathsf{QFHE.Enc}(\mathsf{pk}, \mathsf{td})$.
  - the obfuscated program $\widetilde{\mathbf{CC}} \leftarrow Obf(\mathbf{CC}[\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), s, (r, \mathsf{sk})])$.
  - a commitment $c = \mathsf{Com}(0; r)$ and an SBSH commitment key $\mathsf{ck}_0 \leftarrow\!\$\, \mathsf{SBSH.Gen}(1^\lambda)$.
  - the first message of the 3-round CDS, $(\mathsf{ct}_0, \mathsf{st}_{\mathsf{cds}}) \leftarrow\!\$\, \mathsf{SRP\text{-}CDS.Setup}(1^\kappa)$.
  - the first message of SSim-ZK, $(\mathsf{zk}_0, \mathsf{st}_{\mathsf{SSim\text{-}ZK}}) \leftarrow\!\$\, \mathsf{SSim\text{-}ZK.Gen}(1^\lambda)$.

  It sends $\left(\mathsf{pk}, c, c_{\mathsf{td}}, \widetilde{\mathbf{CC}}, \mathsf{ct}_0, \mathsf{ck}_0, \mathsf{zk}_0\right)$ to the prover.
- **Second Message (P $\to$ V):** The prover samples $y \leftarrow\!\$\, 0^\kappa$ and computes:
  - a commitment key $\mathsf{ck}_1 \leftarrow\!\$\, \mathsf{SBSH.Key}(\mathsf{ck}_0)$ and $\mathsf{cmt}_y \leftarrow \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), y; r_{\mathsf{cmt}_y})$
  - $(\mathsf{zk}_1, \mathsf{td}_{\mathsf{SSim\text{-}ZK}}) \leftarrow \mathsf{SSim\text{-}ZK.R}(\mathsf{zk}_0)$.
  - the second message of the SRP-CDS, $(\mathsf{ct}_R, k) \leftarrow \mathsf{SRP\text{-}CDS.R}(\mathsf{ct}_0, (y, r_{\mathsf{cmt}_y}))$.

  It sends to the verifier $(\mathsf{ck}_1, \mathsf{zk}_1, \mathsf{ct}_R)$.
- **Third Message (V $\to$ P):** The verifier computes
  - $\mathsf{ct}_S \leftarrow \mathsf{SRP\text{-}CDS.S}(\mathsf{ct}_R, \mathsf{st}_{\mathsf{cds}}, z_1, s)$, where the statement z_1 attests to $y = \mathsf{td}$ and $\mathsf{cmt}_y = \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), y; r_{\mathsf{cmt}_y})$.
  - the first message of a WI argument, $(\mathsf{crs}_{wi}, \mathsf{td}_{wi}) \leftarrow\!\$\, \mathsf{WI.Setup}(1^\lambda)$.
  - $\mathsf{zk}_2 \leftarrow \mathsf{SSim\text{-}ZK.S}(\mathsf{zk}_1, \mathsf{st}_{\mathsf{SSim\text{-}ZK}}, z_2, r_{\mathsf{SSim\text{-}ZK}})$, where z_2 is the statement that all of the verifier's messages so far are explainable, with the random coins $r_{\mathsf{SSim\text{-}ZK}}$ as witness.

  It sends to the prover $(\mathsf{ct}_S, \mathsf{crs}_{wi}, \gamma, \mathsf{ck}_2)$.
- **Fourth Message (P $\to$ V):** The prover first verifies $\mathsf{SSim\text{-}ZK.Verify}(\mathsf{td}_{\mathsf{SSim\text{-}ZK}}, \mathsf{zk}_0, \mathsf{zk}_2)$. If the verification is not successful it aborts. Else, on input $p(\lambda)$-many copies of the witness $|w\rangle^{\otimes p(\lambda)}$ and a statement x, it sends a WI proof $|\pi\rangle$:

$$\{x \in \mathcal{L} \vee \exists\, r : c = \mathsf{Com}(0; r)\}\,.$$

- **Verify:** The verifier accepts if $\mathsf{WI.Verify}(\mathsf{td}_{wi}, |\pi\rangle\,, x) = 1$.

Figure 6: Description of a 4-round ZK argument for QMA (plain model)

Soundness. We show that the protocol satisfies computational soundness.

Theorem 5.19. *[Soundness] Assuming the quantum quasi-polynomial hardness of the LWE problem and the existence of a quantum quasi-polynomial semantically secure FHE, the protocol described in Figure 6*

satisfies computational soundness.

Proof. Let P^* be a malicious prover that produces an accepting state $(\mathsf{ck}_3, \mathsf{cmt}_{\mathsf{otk}}, |\psi\rangle, ct'_S, |\pi_2\rangle)$ for some statement $x \notin \mathcal{L}$. We define the aforementioned event as Cheat and assume towards contradiction that the probability of P^* succeeding in cheating is

$$\Pr[\mathsf{Cheat}] \geq \varepsilon(\lambda).$$

Then, by the $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes binding property of the SBSH commitment scheme and the $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes simulatability of the SSim-ZK, we have that

$$\Pr[\mathsf{Cheat} \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding} \wedge (\mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}] \geq \varepsilon(\lambda)^4 \cdot (1 + \mu(\lambda))^2$$

for some negligible function $\mu(\lambda)$. Let $\tilde{y} = \mathsf{SBSH.Ext}(r_{\mathsf{Gen}}, \mathsf{ck}_0, \mathsf{ck}_1, \mathsf{cmt}_y)$ and $z\tilde{k}_2 \leftarrow \mathsf{SSim\text{-}ZK.Sim}(1^\lambda, r_{\mathsf{Setup}}, z_2)$, where r_{Gen} and r_{Setup} are the randomnesses used in the respective first messages. We can now gradually change the procedure and we argue that the probability that the above defined event happens does not decrease significantly.

- The verifier computes and sends a simulated $\tilde{\mathsf{zk}}_2$ instead of zk_2. If we define Cheat_1 as the event that this modified version accepts, we want to argue that

  $$\Pr[\mathsf{Cheat}_1 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding} \wedge \mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}] \geq \varepsilon(\lambda)^4 \cdot (1 + \mu(\lambda))^2.$$

  Observe that the events Cheat and Cheat_1 only differ in case $\tilde{\mathsf{zk}}_2 \neq \mathsf{zk}_2$. If we assume that the inequality doesn't hold we get a contradiction against the sometimes simulatability of SSim-ZK.

- The verifier's third message of the SRP-CDS, ct_S, returns always zero. If we define Cheat_2 as the event that this modified version accepts, we want to prove that

  $$\Pr[\mathsf{Cheat}_2 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding} \wedge \mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}] \geq \varepsilon(\lambda)^4 \cdot (1 + \mu(\lambda))^2.$$

  The proof is presented in Lemma 5.20.

- The verifier's obfuscated program in the first message always returns 0. If we define Cheat_3 as the event that this modified version accepts we want to prove that

  $$\Pr[\mathsf{Cheat}_3 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding} \wedge \mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}] \geq \varepsilon(\lambda)^4 \cdot (1 + \mu(\lambda))^2.$$

  This is true due to the $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ compute and compare obfuscation security

- The verifier's commitment c in the first message is changed to $c = \mathsf{Com}(1, r)$ instead of a commitment to zero. If we define Cheat_4 as the event that this modified version accepts we want to prove that

  $$\Pr[\mathsf{Cheat}_4 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding} \wedge \mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}] \geq \varepsilon(\lambda)^4 \cdot (1 + \mu(\lambda))^2.$$

  The inequality holds due to the $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ hiding of the commitment.

This last inequality implies that the WI proof is accepting with probability at least $\varepsilon(\lambda)^4 \cdot (1+\mu(\lambda))^2$, when neither of the clauses is satisfied. This contradicts the $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ soundness of the WI proof and concludes our proof. □

Lemma 5.20. *Given the definition of the events* Cheat_1 *and* Cheat_2 *in Theorem 5.19 and assuming that*

$$\Pr\left[\mathsf{Cheat}_1 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding} \wedge \mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}\right] \geq \varepsilon(\lambda)^4 \cdot (1 + \mu(\lambda))^2,$$

then

$$\Pr\left[\mathsf{Cheat}_2 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding} \wedge \mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}\right] \geq \varepsilon(\lambda)^4 \cdot (1 + \mu(\lambda))^2.$$

Proof. Consider the interaction where the verifier extracts $\tilde{y}$ using the $\mathsf{SBSH.Ext}$ algorithm, and if $\tilde{y} = \mathsf{td}$ the verifier aborts (denote this event by Abort); otherwise it continues with the interaction. If the event does not happen, then the desired inequality follows by a reduction against the $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ computational sender privacy of the SRP-CDS.

We are now going to show that the probability that Abort happens is negligibly smaller than $\Pr\left[\mathsf{Cheat}_2 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding} \wedge \mathsf{zk}_0, \mathsf{zk}_1) \in \mathsf{Simulation}\right]$. In order to do that we consider the following hybrid distributions:

- Hybrid $\mathcal{H}_a$: This is the protocol we presented above.
- Hybrid $\mathcal{H}_b$: This hybrid process is identical to the above except that the the CC obfuscated program $\widetilde{\mathbf{CC}}$ returns always 0. These processes are computationally indistinguishable given the $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ security of $\widetilde{\mathbf{CC}}$.
- Hybrid $\mathcal{H}_c$: This hybrid process is identical to the above except that the verifier, instead of sending an encryption of td in the first message, it sends $\mathsf{QFHE.Enc}(\mathsf{pk}, 0)$. Computational indistinguishability follows from the $\varepsilon(\lambda)^4 \cdot \mu(\lambda)$ semantic security of QFHE.

In the last hybrid process, we have no information about td, and hence the probability to guess $y = \mathsf{td}$ is negligible. This concludes our proof. □

Quantum Rewinding Lemma Before we move on to zero-knowledge, recall the definition of the Quantum Rewinding Lemma (Lemma 9 from [Wat09]), which constructs a quantum algorithm for amplifying the success probability of quantum sampler circuits under certain conditions. The below definition is taken directly from the modified version in [BS20].

Lemma 5.21. *[Quantum Rewinding Lemma] There is a quantum algorithm* R *that gets as input:*

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

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

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

- *Amplification executes for enough time:* $t \geq \frac{\log(1/\epsilon)}{4 \cdot p_0(1-p_0)}$*,*

- *There is some minimal probability that $b = 0$: For every n-qubit state $|\psi\rangle$, $p_0 \leq p(\psi)$,*
- *$p(\psi)$ is input independant, up to ϵ distance: For every n-qubit state, $|\psi\rangle$, $|p(\psi) - q| < \epsilon$, and*
- *q is closer to $\frac{1}{2}$: $p_0(1-p_0) \leq q(1-q)$,*

then for every n-qubit state $|\psi\rangle$,

$$\mathsf{TD}\left(\mathsf{Q}_\psi, D_\psi\right) \leq 4\sqrt{\epsilon}\frac{\log(1/\epsilon)}{p_0(1-p_0)}$$

where TD *denotes the trace distance.*

Zero-Knowledge. Here we show that the scheme satisfies statistical zero-knowledge. In order to prove ZK, we follow the technique presented in [BS20], so as to simulate aborting verifiers as well. More specifically, we describe two simulators Sim_a and Sim_{na}, that on input (x, V^*, ρ) simulate different types of interactions. Sim_a tries to simulate an aborting interaction and Sim_{na} a non-aborting interaction. Formally, an aborting interaction is an interaction where the verifier aborts or fails to prove the SSim-ZK, whereas a non-aborting interaction is one where the verifier doesn't abort before the fourth message and also the SSim-ZK is succesful. Then, we describe a combined Simulator Sim_{comb}, which randomly chooses $b \leftarrow\$ \{a, na\}$ and uses Sim_b to simulate the interaction. We prove that the output of Sim_{comb} is indistinguishable from the output of the real interaction, as long as it doesn't fail (i.e. picks the correct b), which happens with probability negligibly close to $\frac{1}{2}$. Lastly, it is proven that Sim_{comb} satisfies the required conditions for applying Watrous' quantum rewinding lemma, so that the success probability can be amplified negligibly close to 1.

The simulator $\mathsf{Sim}_a(x, V^*, \rho)$ proceeds as follows:

- In the second message, it computes $\mathsf{cmt}_y = \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), 0^\lambda; r_0)$.
- If at some point before the fourth message the verifier aborts or fails to prove the SSimZK proof, Sim_a outputs the verifier's output. Otherwise it outputs Fail.

The simulator $\mathsf{Sim}_{na}(x, V^*, \rho)$ proceeds as follows:

- It encrypts the the inner state of the verifier at that point $\rho^{(1)}$ under QFHE with public key pk ($\mathsf{ct}_{\rho^{(1)}} = \mathsf{QFHE.Enc}(\mathsf{pk}, \rho^{(1)})$) and proceeds to compute

$$\mathsf{ct}_{\mathsf{cds}_R} \leftarrow \mathsf{QFHE.Eval}(\mathsf{pk}, \mathsf{SRP\text{-}CDS.R}(\mathsf{ct}_0, (\cdot, r_{\mathsf{cmt}_y})), \mathsf{td}) \text{ and}$$

$$\mathsf{ct}_{\mathsf{cmt}_y} \leftarrow \mathsf{QFHE.Eval}(\mathsf{pk}, \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), \cdot\,; r_{\mathsf{cmt}_y}), \mathsf{td}).$$

- Then, Sim_{na} continues to homomorphically evaluate the verifier's response

$$(\mathsf{ct}_{arg3}, \mathsf{ct}_{\rho^{(2)}}) \leftarrow \mathsf{QFHE.Eval}(\mathsf{pk}, V^*, ((\mathsf{ct}_{\mathsf{cds}_R}, \mathsf{ct}_{\mathsf{cmt}_y}), \mathsf{ct}_{\rho^{(1)}})),$$

  where ct_{arg3} is the third message of the verifier V^* and $\mathsf{ct}_{\rho^{(2)}}$ is the new inner state of V^*, both encrypted.
- From the encrypted result of the SRP-CDS.V, given that y was equal to td (under encryption), it gets $\mathsf{QFHE.Enc}(s)$ by running SRP-CDS.D homomorphically. Thus, it can compute $(r, \mathsf{sk}') \leftarrow \widetilde{\mathbf{CC}}(\mathsf{QFHE.Enc}(s))$.

- Subsequently, Sim_{na} checks the validity of $(\mathsf{pk}', \mathsf{sk}) = \mathsf{QFHE.Gen}(1^\lambda; r)$. If $\mathsf{pk}' \neq \mathsf{pk}$ or $\mathsf{sk} \neq \mathsf{sk}'$ then it halts the simulation. Otherwise it obtains the inner state of the verifier by decrypting, using the acquired secret key. The simulator also simulates the missing transcript in the second message with the same values and randomnesses used in the homomorphic computations.
- Lastly, Sim_{na} continues with the protocol by computing and sending the WI proof. It uses as witness the randomness r.

The simulator $\mathsf{Sim}_{comb}(x, V^*, \rho)$ proceed as follows:

- First, it samples $b \leftarrow\!\$ \{a, na\}$.
- Then it runs $\mathsf{Sim}_b(x, V^*, \rho)$.

At last, $\mathsf{Sim}(x, V^*, \rho)$ proceeds as follows:

- Generates the circuit $\mathsf{Sim}_{comb,x,V^*}$, which is the implementation of Sim_{comb} with the inputs x, V^* hardwired, so that ρ is the only input.
- The output of the simulation is $\mathcal{R}\left(\mathsf{Sim}_{comb,x,V^*}, \rho, \lambda\right)$, where $\mathcal{R}$ is the algorithm from Lemma 5.21.

Proposition 5.22. *[Similarity of Aborting Plan] Let $V^* = V^*_\rho$ be an unbounded quantum verifier and let $\mathsf{OUT}_{V^*_a}$ be the verifier's output at the end of the protocol such that if V^* does not abort the output is Fail. We show that*

$$\left\{\mathsf{OUT}_{V^*_a}\left(P(|w\rangle^{\otimes p(\lambda)}, x), V^*(\rho, x)\right)\right\}_{\lambda,x,w} \approx_s \left\{\mathsf{Sim}_a(x, V^*, \rho))\right\}_{x,w},$$

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

Proof. The two distributions are identical, since both Sim_a and the prover act exactly the same up to the fourth message. In an aborting interaction the verifier would have aborted before this message. In the case of a non-aborting interaction, both outputs would be Fail. □

Proposition 5.23. *[Similarity of Non-Aborting Plan] Let V^* be an unbounded quantum verifier and let $\mathsf{OUT}_{V^*_{na}}$ be the verifier's output at the end of the protocol such that if V^* aborts the output is Fail. We show that*

$$\left\{\mathsf{OUT}_{V^*_{na}}\left(P(|w\rangle^{\otimes p(\lambda)}, x), V^*(\rho, x)\right)\right\}_{\lambda,x,w} \approx_s \left\{\mathsf{Sim}_{na}(x, V^*, \rho))\right\}_{x,w},$$

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

Proof. We consider the following hybrid distributions, which we prove that are statistically indistinguishable:

- Hybrid $\mathcal{H}_0$: This is the output distribution of Sim_{na}
- Hybrid $\mathcal{H}_1$: This process is identical to the above except that in the WI proof the simulator proves the first statement ($x \in \mathcal{L}$). Indistinguishability follows from the witness-indistinguishability property of the WI proofs.

- Hybrid $\mathcal{H}_2$: This hybrid process is identical to the above except that if the verifier's messages are not explainable and its SSim-ZK proof fails, then the process chooses to fail and outputs Fail. Otherwise, after performing the homomorphic computations, instead of getting the sk from the $\widetilde{\mathbf{CC}}$, it computes it inefficiently. It also computes s inefficiently and if $\mathsf{QFHE.Dec}(\mathsf{QFHE.Enc}(s)) \neq s$ (where $\mathsf{QFHE.Enc}(s)$ is part of ct_{arg_3}) it outputs Fail. Else, it continues with the simulation.

  Statistical indistinguishability will follow from the perfect correctness of the CC obfuscation, the perfect correctness of the QFHE and from the soundness of the SSim-ZK that the verifier sends. Assume that the two distributions are distinguishable and fix a partial transcript T' and a verifier's inner state $\rho^{(1)}$ that maximize the distinguishability.

  - In case T' is not expainable, the SSim-ZK will fail, and so will the hybrid process, resulting in a contradiction (since both outputs would be Fail).
  - In case T' is expainable, in both hybrids we can check if s is correct after obtaining the sk, either inneficiently or through $\widetilde{\mathbf{CC}}$. Hence, the statistical distance between them is bounded by the probability that the check in one hybrid process fails and succeeds in the other, which in turn is bounded by the result of the SRP-CDS not being equal to s. Given the statistical correctness of the QFHE scheme (under which the homomorphic evaluations are performed), this leads to a contradiction.

- Hybrid $\mathcal{H}_3$: In this hybrid distribution we get rid of the homomorphic evaluation altogether. If the verifier's messages are explainable (and thus specifically $\mathsf{QFHE.Enc}(\mathsf{td})$) then the simulator sends cmt_y and ct_R in the clear (similar as in the original protocol, using td in place of $y = 0^\kappa$). If the verifier's SRP-CDS decrypted is equal to the precomputed s then the process continues. Otherwise, the process outputs Fail.

  Statistical indistinguishability follows directly from the perfect correctness of the QFHE and the soundness of the SSim-ZK. Assume that the two distributions are distinguishable and fix a partial transcript T' and a verifier's inner state $\rho^{(1)}$ that maximize the distinguishability.

  - In case T' is not explainable, the SSim-ZK as well as the hybrid process would output Fail, resulting in a contradiction.
  - In case T' is explainable, the difference in the distributions is that in one the verifier's response is computed homomorphically and in the other in the clear. By the QFHE correctness, this leads to a contradiction.

- Hybrid $\mathcal{H}_4$: This process is identical to the previous except that the simulator does not check the verifier's SRP-CDS response and always continues with the process. Assume that the two distributions are distinguishable and fix a partial transcript T'.

  - If T' is not explainable then both hybrids would output Fail and are thus identical.
  - If T' is explainable and the result of the SRP-CDS is equal to s then the hybrids are identical. Alternatively, if the result of the SRP-CDS is not equal to s, then $\mathcal{H}_3$ would output Fail, but, in the current hybrid, due to the correctness of the SRP-CDS, T' should not be explainable. Given the soundness of the SSim-ZK we reach a contradiction.

- Hybrid $\mathcal{H}_5$: This hybrid process is identical to the previous except that the prover always sends $\mathsf{cmt}_y \leftarrow \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), 0^\lambda; r_{\mathsf{cmt}_y})$ in the second message. Statistical indistinguishability follows from the statistical hiding of the commitment and the SRP-CDS statistical privacy. Assume towards contradiction that the distributions are distinguishable and fix a partial transcript T′.
  - If T′ is not explainable then both hybrids would output Fail thanks to the soundness of the SSim-ZK.
  - In case T′ is explainable, we reach a contradiction due to the statistical security of the SBSH commitment and the statistical privacy of SRP-CDS.
- Hybrid $\mathcal{H}_6$: This hybrid process is identical to the previous except that instead of getting s and sk inefficiently and verifying V^*'s messages (with the Gen algorithm), it always sends $\mathsf{ct}_R \leftarrow \mathsf{SRP\text{-}CDS.R}(\mathsf{ct}_0, (0^\lambda, r_{\mathsf{cmt}_y}))$ in the second message. Assume towards contradiction that the distributions are distinguishable and fix partial transcript T′.
  - If T′ is not explainable then both hybrids would output Fail thanks to the soundness of the SSim-ZK.
  - In case T′ is explainable we reach a contradiction due to the statistical privacy of SRP-CDS.

  Note that this last process is exactly the output of the interaction with a prover.

□

Next we prove that the output of of a successful Sim_{comb} is indistinguishable from a real interaction. The Proposition is identical to Proposition 3.4 in [BS20], with the necessary changes in order to argue statistical zero knowledge.

Proposition 5.24. *[The output of of a successful Sim_{comb} is Indistinguishable from Real Interaction] Let V^* be a verifier. For $x \in \mathcal{L}$, let $\widetilde{\mathsf{Sim}_{comb}}(x, V^*_\lambda, \rho_\lambda)$ denote the conditional distribution of $\mathsf{Sim}_{comb}(x, V^*_\lambda, \rho_\lambda)$, conditioned on the simulation being successful. Then,*

$$\left\{\mathsf{OUT}\left(P(|w\rangle^{\otimes p(\lambda)}, x), V^*(\rho, x)\right)\right\}_{\lambda,x,w} \approx_s \left\{\widetilde{\mathsf{Sim}_{comb}}(x, V^*, \rho))\right\}_{x,w},$$

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

Proof. Denote the following conditional distributions.

- $\mathsf{A}_{\mathsf{Sim}}$: A conditional distribution of $\mathsf{Sim}_a(x, V^*, \rho)$, conditioned on that the output is not Fail (might be an empty distribution, if $a(x, \rho) = 0$).
- $\mathsf{S}_{\mathsf{Sim}}$: A conditional distribution of $\mathsf{Sim}_{na}(x, V^*, \rho)$, conditioned on that the output is not Fail (might be an empty distribution, if $b(x, \rho) = 1$).
- $\mathsf{A}_{\langle P,V^*\rangle} = \left\{\mathsf{A}_{\langle P,V^*\rangle\lambda}\right\}_{\lambda\in\mathbb{N}}$: A conditional distribution of $\mathsf{OUT}_{V^*_a}\langle P, V^*\rangle$, conditioned on that the output is not Fail (might be an empty distribution, if $c(x, \rho, |w\rangle^{\otimes p(\lambda)}) = 0$).
- $\mathsf{S}_{\langle P,V^*\rangle} = \left\{\mathsf{S}_{\langle P,V^*\rangle\lambda}\right\}_{\lambda\in\mathbb{N}}$:A conditional distribution of $\mathsf{OUT}_{V^*_{na}}\langle P, V^*\rangle$, conditioned on that the output is not Fail (might be an empty distribution, if $c(x, \rho, |w\rangle^{\otimes p(\lambda)}) = 1$).

where the probabilities a, b, c are defined as follows:

- $a(x,\rho)$: The probability that the simulation of $\mathsf{Sim}_a(x, V^*, \rho)$ was aborting.
- $b(x,\rho)$: The probability that the simulation of $\mathsf{Sim}_{na}(x, V^,\rho)$ was aborting.
- $c(x,\rho,|w\rangle^{\otimes p(\lambda)})$: The probability that the interaction $\left(P(|w\rangle^{\otimes p(\lambda)}), V^*(\rho)\right)(x)$ was aborting.

Proof. □

The distribution $\widetilde{\mathsf{Sim}_{comb}}(x, V^*, \rho)$ is the distribution generated by outputting a sample from $\mathsf{A}_{\mathsf{Sim}}$ with probability $\frac{a(x,\rho}{1+a(x,\rho)-b(x,\rho)}$, and a sample from $\mathsf{S}_{\mathsf{Sim}}$ with probability $\frac{1-b(x,\rho}{1+a(x,\rho)-b(x,\rho)}$. In addition, the distribution $\mathsf{OUT}_{V^*}(P, V^*)(x)$ is the distribution generated by outputting a sample from $\mathsf{A}_{(P,V^*)}$ with probability $c(x,\rho,|w\rangle^{\otimes p(\lambda)})$ and from $\mathsf{S}_{\langle P,V^*\rangle}$ with probability $1 - c(x,\rho,|w\rangle^{\otimes p(\lambda)})$. We will show that the two distributions are *statistically* indistinguishable by a hybrid argument. Consider the following distributions:

- Hyb_0: This is the distribution $\widetilde{\mathsf{Sim}_{comb}}(x, V^*, \rho)$.
- Hyb_1: This process is identical to the above with the exception that instead of sampling from $\mathsf{A}_{\mathsf{Sim}}$ with probability $\frac{a(x,\rho}{1+a(x,\rho)-b(x,\rho)}$ and from $\mathsf{S}_{\mathsf{Sim}}$ with probability $\frac{1-b(x,\rho}{1+a(x,\rho)-b(x,\rho)}$, it samples from $\mathsf{A}_{\mathsf{Sim}}$ with probability $a(x,\rho)$ and from $\mathsf{S}_{\mathsf{Sim}}$ with probability $1 - a(x,\rho)$.
- Hyb_2: This process is identical to the above, but the probability $a(x,\rho)$ is changed to $c(x,\rho,|w\rangle^{\otimes p(\lambda)})$.
- Hyb_3: This process is identical to the above except that with probability $c(x,\rho,|w\rangle^{\otimes p(\lambda)})$ the process outputs a sample from $\mathsf{A}_{\langle P,V^*\rangle}$ instead of $\mathsf{A}_{\mathsf{Sim}}$.
- Hyb_4: This process is identical to the above except that with probability $1 - c(x,\rho,|w\rangle^{\otimes p(\lambda)})$ the process outputs a sample from $\mathsf{S}_{\langle P,V^*\rangle}$ instead of $\mathsf{S}_{\mathsf{Sim}}$.

Following the proof from [BS20] while using propositions 5.22 and 5.23 we prove statistical indistinguishability between the above hybrid distributions. □

Theorem 5.25 (Zero Knowledge)**.** *Let $V^* = V^*_\rho$ be an unbounded quantum verifier. The protocol described in Figure 6 satisfies statistical zero-knowledge:*

$$\left\{\mathsf{OUT}\left(P(|w\rangle^{\otimes p(\lambda)}, x), V^*(\rho, x)\right)\right\}_{\lambda,x,w} \approx_s \{\mathsf{Sim}(x, V^*, \rho))\}_{x,w},$$

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

Proof. The proof is identical with the proof of Proposision 3.5 in [BS20], where the authors use Watrous' Rewinding Lemma for $\mathsf{Sim}_{comb,x,V^*}$, which has probability success negligibly close to $1/2$. If we denote the success probability for input ρ by $p(\rho)$ and denote $\epsilon \coloneqq \mathsf{negl}(\lambda) + 2^{-\lambda\frac{3}{4}}, p_0 \coloneqq \frac{1}{4}$ and $q \coloneqq \frac{1}{2}$, the conditions for the Quantum Rewinding Lemma [cite it] are satisfied.

Thus the trace distance between $\mathsf{Sim}_{comb}(x, V^*, \rho)$ and $\mathsf{R}\left(\mathsf{Sim}_{comb,x,V^*,\rho}(x, V^*, \rho)\right) = \mathsf{Sim}(x, V^*, \rho)$ is bounded by a negligible function. Finally, observe that as proven in Proposition 5.24, $\mathsf{Sim}_{comb}(x, V^*, \rho))$ is statistically indistinguishable from $\mathsf{OUT}_{V^*}\left\langle P(|w\rangle^{\otimes p(\lambda)}), V^*(\rho)\right\rangle(x)$, which concludes the proof. □

6 Zero-Knowledge for QMA in the Timing Model

In this section we present two zero-knowledge arguments for QMA languages in the timing model: The first satisfies computational zero-knowledge, whereas the latter satisfies statistical zero-knowledge but requires slightly stronger assumptions.

6.1 Computational Zero-Knowledge

Our Protocol. Here we describe our first 2-round ZK argument. We assume the existence of the following building blocks (all secure against quantum adversaries):

- A circuit-private classical FHE scheme $(\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$.
- A sub-exponentially secure average-case T-non-parallelizing function $F : \mathcal{X} \to \mathcal{Y}$ secure against algorithms of size $O(2^\lambda)$ and depth less than T^ζ.
- A 2-round WI argument $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ for QMA.

Our protocol is parametrized by a time-parameter T and it is formally described in Figure 7. Completeness follows immediately from the completeness of the 2-round WI argument.

2-Round (Computational) ZK Arguments for QMA

- **Prover Precomputation:** Sample an FHE key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{FHE.Gen}(1^\lambda)$ and an input $x' \leftarrow\$ \mathcal{X}$. Compute $\alpha \leftarrow\$ \mathsf{FHE.Enc}(\mathsf{pk}, x')$ and $\beta = \mathsf{FHE.Eval}(\mathsf{pk}, F, \alpha)$.
- **First Message (V $\to$ P):** Sample a uniform input $x^* \leftarrow\$ \mathcal{X}$ and a first message $(\mathsf{crs}, \mathsf{td}) \leftarrow\$ \mathsf{WI.Setup}(1^\lambda)$ and return (x^*, crs).
- **Second Message (P $\to$ V):** On input $p(\lambda)$-many copies of the witness $|w\rangle^{\otimes p(\lambda)}$ and a statement z, compute a WI proof $|\pi\rangle$ for the statement

$$\mathsf{stmt} = \left\{ x \in \mathcal{L} \vee \mathsf{pk} \in \mathsf{FHE.Gen}(1^\lambda) \wedge \alpha \in \mathsf{FHE.Enc}(\mathsf{pk}, x^*) \right\}$$

  using $|w\rangle^{\otimes p(\lambda)}$ as the witness. The prover sends $(|\pi\rangle, \mathsf{pk}, \alpha, \beta)$ to the verifier.
- **Verify:** The verifier accepts if the following conditions are satisfied.
  1. The prover responds before time T^ζ.
  2. $\mathsf{WI.Verify}(\mathsf{td}, |\pi\rangle, \mathsf{stmt}) = 1$.
  3. $\mathsf{FHE.Eval}(\mathsf{pk}, F, \alpha) = \beta$.

Figure 7: Description of a 2-round (computational) ZK argument for QMA (timing model)

Soundness. In the following we show that the protocol satisfies computational soundness.

Theorem 6.1 (Soundness)**.** *Assuming that F is sub-exponentially average-case non-parallelizable and that* $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ *is computationally sound, the protocol described in Figure 7 satisfies single-theorem computational soundness.*

Proof. Consider a prover (running in time less than T^ζ) that produces an accepting state $(|\pi\rangle, \mathsf{pk}, \alpha, \beta)$ for some statement $x \notin \mathcal{L}$. By the computational soundness of the WI proof, it must be the case that

$$\mathsf{pk} \in \mathsf{FHE.Gen}(1^\lambda) \wedge \alpha \in \mathsf{FHE.Enc}(\mathsf{pk}, x^*) \tag{2}$$

as otherwise it would produce a valid WI second message for a false statement. We can use the prover to define an algorithm that breaks the (sub-exponential) non-parellelizability of F as follows: The reduction sets x^* to be the challenge input and proceeds with the protocol in the same way as the verifier would. Once the prover returns $(\mathsf{pk}, \alpha, \beta)$, the reduction recovers the sk in time $O(2^\lambda)$, by e.g. testing all random strings of the $\mathsf{FHE.Gen}$ algorithm in parallel. Then it and uses sk to decrypt β and returns whatever the decrypted message is.

Observe that, by Equation (1), α is indeed an encryption of x^*. By the evaluation correctness of the FHE scheme, we have that

$$\mathsf{FHE.Dec}(\mathsf{sk}, \beta) = \mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{FHE.Eval}(\mathsf{pk}, F, \alpha)) = F(x^*).$$

Thus, the reduction returns the correct output. What is left to be shown is that the depth of the reduction is asymptotically smaller than T^ζ. Observe that the process of recovering sk can be computed by a circuit of depth $O(\lambda)$, by testing all random coins of the $\mathsf{FHE.Gen}$ in parallel and then selecting the matching secret key with a binary tree. The depth of the decryption procedure is bounded by a fixed polynomial in λ and is in particular independent of T. Thus, the depth of the reduction is only an additive term $\mathsf{poly}(\lambda)$ higher than the depth of the prover. For a large enough T, this contradicts the non-parallelizability of F. □

Zero-Knowledge. Finally, we show that the scheme satisfies zero-knowledge in the timing model. Recall that in the timing model [DS02] the simulator is allowed to "freeze time" while simulating the accepting transcript.

Theorem 6.2 (Zero-Knowledge)**.** *Assuming that* $(\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ *is semantically secure, the protocol described in Figure 7 satisfies computational zero-knowledge in the timing model.*

Proof. The simulator computes α as an encryption of x^*, then it computes β as $\mathsf{FHE.Eval}(\mathsf{pk}, F, \alpha)$ and uses the corresponding random coins as a witness to compute the WI argument. Recall that the simulator is allowed to perform computations without letting time elapsing, from the perspective of the verifier. To show that the simulation is computationally indistinguishable from the real proof, we consider the following hybrid distributions.

- Hybrid $\mathcal{H}_0$: This is the honestly computed proof $(|\pi\rangle, \mathsf{pk}, \alpha, \beta)$.
- Hybrid $\mathcal{H}_1$: Here we change α to be the encryption of x^*, instead of x'. Computational indistinguishability follows immediately from the semantic security of the FHE scheme.

- Hybrid $\mathcal{H}_2$: Here we use the random coins used to sample pk and to encrypt α to compute the WI proof, as opposed to the witness $|w\rangle^{\otimes p(\lambda)}$. By the statistical indistinguishability of the WI argument, the distributions are statistically close.

The proof is concluded by observing that the distribution induced by $\mathcal{H}_2$ is the same as the one induced by the simulator. □

6.2 Statistical Zero-Knowledge

We show a different protocol that achieves statistical zero-knowledge at the cost of requiring slightly stronger assumptions, namely the existence of a post-quantum time-lock puzzle. At present, we only know how to construct (presumably) post-quantum time-lock puzzles from succinct randomized encodings [BGJ+16].

Our Protocol. Let $\varepsilon(\lambda)$ be a (fixed) negligible function. We assume the existence of the following building blocks (all secure against quantum adversaries):

- A perfectly binding commitment Com which is hiding with $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ advantage.
- A 2-round WI argument $(\mathsf{WI.Setup}, \mathsf{WI.Prove}, \mathsf{WI.Verify})$ for QMA with statistical witness indistinguishability and $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ soundness error.
- A two-round statistically sender private conditional disclosure of secrets scheme $(\mathsf{SSP\text{-}CDS.R}, \mathsf{SSP\text{-}CDS.S}, \mathsf{SSP\text{-}CDS.D})$ for NP with $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ receiver security.
- A time-lock puzzle $(\mathsf{TLP.Gen}, \mathsf{TLP.Solve})$ T-sequential with advantage bounded by $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$.
- An SBSH commitment scheme $(\mathsf{SBSH.Gen}, \mathsf{SBSH.Key}, \mathsf{SBSH.Com})$ that satisfies $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes binding.

Where $\mu(\lambda)$ is some negligible function. Note that, with the exception of the time-lock puzzles, all other building blocks can be instantiated assuming the quantum hardness of quasi-polynomial LWE. We define T to be the time parameter of the scheme and we describe our protocol in Figure 8.

Soundness. We show that our protocol satisfies (non-adaptive) soundness.

Theorem 6.3 (Soundness)**.** *Assuming the quantum quasi-polynomial hardness of the LWE problem and quasi-polynomially T-sequential time-lock puzzles, the ZK argument described in Figure 8 satisfies computational soundness.*

Proof. We show that the success probability of the prover is bounded by a negligible function $\varepsilon(\lambda)$. Let $x \notin \mathcal{L}$ be the false statement and let Cheat be the event where the prover causes the verifier to accept x. Assume towards contradiction that

$$\Pr\left[\mathsf{Cheat}\right] \geq \varepsilon(\lambda).$$

Then, by the $(\varepsilon(\lambda), \varepsilon(\lambda)^2)$-sometimes binding property of the SBSH commitment scheme, we have that

$$\Pr\left[\mathsf{Cheat} \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$$

2-Round (Statistical) ZK Arguments for QMA

- **First Message (V $\to$ P):** Sample an SBSH commitment key $\mathsf{ck}_0 \leftarrow\$ \mathsf{SBSH.Gen}(1^\lambda)$ and the first messages of two WI arguments $(\mathsf{crs}_1, \mathsf{td}_1), (\mathsf{crs}_2, \mathsf{td}_2) \leftarrow\$ \mathsf{WI.Setup}(1^\lambda)$. Sample a uniform $c \leftarrow\$ \mathsf{Com}(1^\lambda, 0; r)$ and compute $Z \leftarrow\$ \mathsf{TLP.Gen}(1^\lambda, T, r; \tilde{r})$ and compute the first message of a CDS $\mathsf{ct} \leftarrow\$ \mathsf{SSP\text{-}CDS.R}(1^\lambda, (r, \tilde{r}))$. Return $(\mathsf{ck}_0, \mathsf{crs}_1, \mathsf{crs}_2, c, Z, \mathsf{ct})$.

- **Second Message (P $\to$ V):** On input $p(\lambda)$-many copies of the witness $|w\rangle^{\otimes p(\lambda)}$ and a statement x, compute a WI proof $|\pi_1\rangle$ (with respect to crs_1) for the statement

$$\mathsf{stmt}_1 = \left\{ x \in \mathcal{L} \vee \exists\, r : c = \mathsf{Com}(1^\lambda, 0 : r) \right\}.$$

  Sample $\mathsf{otk} \leftarrow\$ \mathsf{QOTP.Gen}(1^\lambda)$ and calculate $|\psi\rangle = \mathsf{QOTP.Enc}(\mathsf{otk}, |\pi_1\rangle)$. Then sample $\mathsf{ck}_1 \leftarrow\$ \mathsf{SBSH.Key}(\mathsf{ck}_0)$ and compute $\mathsf{cmt} \leftarrow\$ \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), \mathsf{otk})$. Compute the CDS message ct' for otk, conditioned the statement

$$\mathsf{stmt}_0 = \left\{ \exists\, r : Z \in \mathsf{TLP.Gen}(1^\lambda, T, r) \wedge c = \mathsf{Com}(1^\lambda, 0; r) \right\}.$$

  Finally compute WI proof $|\pi_2\rangle$ (with respect to crs_2) for the statement

$$\mathsf{stmt}_2 = \left\{ x \in \mathcal{L} \vee \mathsf{cmt} \in \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), \mathsf{otk}) \wedge \mathsf{ct}' \in \mathsf{SSP\text{-}CDS.S}(\mathsf{ct}, \mathsf{stmt}_0, \mathsf{otk}) \right\}$$

  using the witness for the second branch. Return $(\mathsf{ck}_1, \mathsf{cmt}, \mathsf{ct}', |\psi\rangle, |\pi_2\rangle)$.

- **Verify:** The verifier computes $\mathsf{otk}' = \mathsf{SSP\text{-}CDS.D}(\mathsf{ct}', (r, \tilde{r}))$ and $|\pi_1\rangle = \mathsf{QOTP.Dec}(\mathsf{otk}', |\psi\rangle)$. The verifier accepts if the following conditions are satisfied.
  1. The prover responds before time T^ζ.
  2. $\mathsf{WI.Verify}(\mathsf{td}_1, |\pi_1\rangle, \mathsf{stmt}_1) = 1$.
  3. $\mathsf{WI.Verify}(\mathsf{td}_2, |\pi_2\rangle, \mathsf{stmt}_2) = 1$.

Figure 8: Description of a 2-round (statistical) ZK argument for QMA (timing model)

for some negligible function $\mu(\lambda)$. Let $\tilde{\mathsf{otk}} = \mathsf{SBSH.Ext}(r, \mathsf{ck}_0, \mathsf{ck}_1, \mathsf{cmt})$ be the output of the extractor, where r denote the random coins used in the $\mathsf{SBSH.Gen}$ algorithm. We now gradually change the verification procedure and we argue that the probability that the above defined event happens does not decrease significantly.

- The verifier computes $|\pi_1\rangle = \mathsf{QOTP.Dec}(\tilde{\mathsf{otk}}, |\psi\rangle)$, instead of recovering otk' from the CDS protocol. Let us now define Cheat_1 as the event where the modified verifier accepts. We want to argue that

$$\Pr\left[\mathsf{Cheat}_1 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$$

for some negligible function $\mu(\lambda)$. Note that the events Cheat and Cheat_1 only differ in the case where $\tilde{\mathsf{otk}} \neq \mathsf{otk}'$. Thus if the inequality above does not hold, we obtain a contradiction against the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-soundness of the WI argument.

- The verifier computes $\mathsf{ct} \leftarrow\!\$\, \mathsf{SSP\text{-}CDS.R}(1^\lambda, 0)$ and we define Cheat_2 as the event where the modified verifier accepts. We can show that

$$\Pr\left[\mathsf{Cheat}_2 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$$

  by a reduction against the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$-receiver hiding of the CDS scheme.

- The verifier computes $Z \leftarrow\!\$\, \mathsf{TLP.Gen}(1^\lambda, T, 0)$ and we define Cheat_3 as the event where the modified verifier accepts. We can show that

$$\Pr\left[\mathsf{Cheat}_3 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$$

  by a reduction against the T-sequentiality of the time-lock puzzle with advantage $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$.

- The verifier computes $c \leftarrow\!\$\, \mathsf{Com}(1^\lambda, 1)$ and we define Cheat_4 as the event where the modified verifier accepts. We have that

$$\Pr\left[\mathsf{Cheat}_4 \wedge (\mathsf{ck}_0, \mathsf{ck}_1) \in \mathsf{Binding}\right] \geq \varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$$

  by the $\varepsilon(\lambda)^2 \cdot \mu(\lambda)$ hiding of the commitment scheme Com.

The last inequality implies that the prover produces a valid $|\pi_1\rangle$ for a false statement with probability greater than $\varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$, which is a contradiction to the $\varepsilon(\lambda)^2 \cdot (1 + \mu(\lambda))$-soundness of the WI argument and concludes the proof. □

Zero-Knowledge. We now argue that the protocol is zero-knowledge in the timing model.

Theorem 6.4 (Zero-Knowledge)**.** *The protocol described in Figure 8 satisfies statistical zero-knowledge in the timing model.*

Proof. The simulator recovers a randomness r from Z (by computing $\mathsf{TLP.Solve}$) and checks whether $c = \mathsf{Com}(1^\lambda, 0; r)$. If this is the case it uses it as the witness to compute $|\pi_1\rangle$, otherwise it sets $|\pi_1\rangle$ to be the all 0 state (padded to the appropriate length). The simulator then proceeds as in the real protocol.

To show that the transcript produced by the simulator is statistically close to the one produced by the real prover, we consider the following hybrid distributions.

- Hybrid $\mathcal{H}_0$: This is the simulated transcript.

- Hybrid $\mathcal{H}_1$: We change the simulation to compute $|\pi_1\rangle$ using the real witness of the statement z, but only in the case where $c = \mathsf{Com}(1^\lambda, 0; r)$. By the statistical witness indistinguishability of the WI argument, this change is statistically indistinguishable.

- Hybrid $\mathcal{H}_2$: Here we compute $|\pi_2\rangle$ using the real witness of the statement z. This change is statistically indistinguishable to the eyes of the verifier by the statistical witness indistinguishability of the WI argument.

- Hybrid $\mathcal{H}_3$: If $c \neq \mathsf{Com}(1^\lambda, 0; r)$ we compute the CDS second message ct' with the message fixed to 0 (padded to the appropriate length), instead of otk. Note that the condition $c \neq \mathsf{Com}(1^\lambda, 0; r)$ implies that stmt is false, and therefore the distribution induced by this hybrid is statistically close to that of the previous one.
- Hybrid $\mathcal{H}_4$: If $c \neq \mathsf{Com}(1^\lambda, 0; r)$ we compute $\mathsf{cmt} \leftarrow\!\!\$\, \mathsf{SBSH.Com}((\mathsf{ck}_0, \mathsf{ck}_1), 0)$. This change is statistically indistinguishable by the statistical hiding property of the SBSH commitment.
- Hybrid $\mathcal{H}_5$: If $c \neq \mathsf{Com}(1^\lambda, 0; r)$ we compute $|\psi\rangle = \mathsf{QOTP}(\mathsf{otk}, |\pi_1\rangle)$, where $|\pi_1\rangle$ is computed using the real witness for z. To the eyes of the distinguisher $|\psi\rangle$ is now maximally mixed and therefore this distribution is identical to that of the previous hybrid.
- Hybrid $\mathcal{H}_6$: We revert the change done in $\mathcal{H}_4$.
- Hybrid $\mathcal{H}_7$: We revert the change done in $\mathcal{H}_3$.
- Hybrid $\mathcal{H}_8$: We revert the change done in $\mathcal{H}_2$.

The proof is concluded by observing that $\mathcal{H}_8$ is identical to the output of the honest prover. □

Acknowledgements. G.M. wishes to thank James Bartusek and Dakshita Khurana for many insightful discussions and helpful comments at an early stage of this work. The authors are also thankful to Alex Lombardi, Fermi Ma, and the anonymous reviewers of TCC 2021 for pointing out a bug in the analysis of the statistically zero-knowledge sigma protocol in [BG20] and for their many insightful comments. The fix (Section 4.2) is included here with their permission.

References

[ACGH20] Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 153–180. Springer, Heidelberg, November 2020.

[ACP20] Prabhanjan Ananth, Kai-Min Chung, and Rolando L. La Placa. On the concurrent composition of quantum zero-knowledge. Cryptology ePrint Archive, Report 2020/1528, 2020. https://eprint.iacr.org/2020/1528.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th FOCS*, pages 166–175. IEEE Computer Society Press, October 2004.

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 119–135. Springer, Heidelberg, May 2001.

[AL20] Prabhanjan Ananth and Rolando L. La Placa. Secure quantum extraction protocols. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 123–152. Springer, Heidelberg, November 2020.

[AMTDW00] Andris Ambainis, Michele Mosca, Alain Tapp, and Ronald De Wolf. Private quantum channels. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 547–553. IEEE, 2000.

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

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private OT from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 370–390. Springer, Heidelberg, November 2018.

[BDGM19] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Leveraging linear decryption: Rate-1 fully-homomorphic encryption and time-lock puzzles. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 407–437. Springer, Heidelberg, December 2019.

[BFJ+20] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 642–667. Springer, Heidelberg, May 2020.

[BG20] Anne Broadbent and Alex B. Grilo. QMA-hardness of consistency of local density matrices with applications to quantum zero-knowledge. In *61st FOCS*, pages 196–205. IEEE Computer Society Press, November 2020.

[BGJ+16] Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles from randomized encodings. In Madhu Sudan, editor, *ITCS 2016*, pages 345–356. ACM, January 2016.

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM CCS 2012*, pages 784–796. ACM Press, October 2012.

[BJSW16] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. In Irit Dinur, editor, *57th FOCS*, pages 31–40. IEEE Computer Society Press, October 2016.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 671–684. ACM Press, June 2018.

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1091–1102. ACM Press, June 2019.

[BM21] James Bartusek and Giulio Malavolta. Candidate obfuscation of null quantum circuits and witness encryption for qma. Cryptology ePrint Archive, Report 2021/421, 2021. `https://eprint.iacr.org/2021/421`.

[BP19] Nir Bitansky and Omer Paneth. On round optimal statistical zero knowledge arguments. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 128–156. Springer, Heidelberg, August 2019.

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

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

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In Moni Naor, editor, *ITCS 2014*, pages 1–12. ACM, January 2014.

[CCLY21] Nai-Hui Chia, Kai-Min Chung, Qipeng Liu, and Takashi Yamakawa. On the impossibility of post-quantum black-box zero-knowledge in constant rounds. Cryptology ePrint Archive, Report 2021/376, 2021. `https://eprint.iacr.org/2021/376`.

[CCY20] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 181–206. Springer, Heidelberg, November 2020.

[CDM20] Orestis Chardouvelis, Nico Döttling, and Giulio Malavolta. Rate-1 secure function evaluation for bqp. Cryptology ePrint Archive, Report 2020/1454, 2020. `https://eprint.iacr.org/2020/1454`.

[CFGS18] Alessandro Chiesa, Michael A. Forbes, Tom Gur, and Nicholas Spooner. Spatial isolation implies zero knowledge even in a quantum world. In Mikkel Thorup, editor, *59th FOCS*, pages 755–765. IEEE Computer Society Press, October 2018.

[CVZ20] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive zero-knowledge arguments for QMA, with preprocessing. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 799–828. Springer, Heidelberg, August 2020.

[DGI+19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 3–32. Springer, Heidelberg, August 2019.

[DS02] Cynthia Dwork and Larry J. Stockmeyer. 2-round zero knowledge and proof auditors. In *34th ACM STOC*, pages 322–331. ACM Press, May 2002.

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

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

[GJJM20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 668–699. Springer, Heidelberg, May 2020.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, June 1996.

[GKVW19] Rishab Goyal, Venkata Koppula, Satyanarayana Vusirikala, and Brent Waters. On perfect correctness in (lockable) obfuscation. Cryptology ePrint Archive, Report 2019/1010, 2019. `https://eprint.iacr.org/2019/1010`.

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

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

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th FOCS*, pages 174–187. IEEE Computer Society Press, October 1986.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.

[GSY19] Alex Bredariol Grilo, William Slofstra, and Henry Yuen. Perfect zero knowledge for quantum multiprover interactive proofs. In David Zuckerman, editor, *60th FOCS*, pages 611–635. IEEE Computer Society Press, November 2019.

[HSS11] Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 411–428. Springer, Heidelberg, August 2011.

[HW18] Susan Hohenberger and Brent Waters. Synchronized aggregate signatures from the RSA assumption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 197–229. Springer, Heidelberg, April / May 2018.

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st FOCS*, pages 294–304. IEEE Computer Society Press, November 2000.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 158–189. Springer, Heidelberg, August 2017.

[JMR20] Samuel Jaques, Hart Montgomery, and Arnab Roy. Time-release cryptography from minimal circuit assumptions. Cryptology ePrint Archive, Report 2020/755, 2020. https://eprint.iacr.org/2020/755.

[KKS18] Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 34–65. Springer, Heidelberg, April / May 2018.

[KNYY21] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Round-optimal blind signatures in the plain model from classical and quantum standard assumptions. Cryptology ePrint Archive, Report 2021/306, 2021. https://eprint.iacr.org/2021/306.

[KS17] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. In Chris Umans, editor, *58th FOCS*, pages 564–575. IEEE Computer Society Press, October 2017.

[LN11] Carolin Lunemann and Jesper Buus Nielsen. Fully simulatable quantum-secure coin-flipping and applications. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11*, volume 6737 of *LNCS*, pages 21–40. Springer, Heidelberg, July 2011.

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. Cryptology ePrint Archive, Report 2019/279, 2019. https://eprint.iacr.org/2019/279.

[LVW20] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Statistical ZAPR arguments from bilinear maps. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 620–641. Springer, Heidelberg, May 2020.

[OPP14] Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private FHE. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 536–553. Springer, Heidelberg, August 2014.

[Pas03a] Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 316–337. Springer, Heidelberg, August 2003.

[Pas03b] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 160–176. Springer, Heidelberg, May 2003.

[PRS17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 461–473. ACM Press, June 2017.

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

[RSW96] Ronald L. Rivest, Adi Shamir, and David A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, 1996.

[Shm20] Omri Shmueli. Multi-theorem (malicious) designated-verifier NIZK for QMA. Cryptology ePrint Archive, Report 2020/928, 2020. `https://eprint.iacr.org/2020/928`.

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

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

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

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.