On the Concurrent Composition of Quantum Zero-Knowledge

Prabhanjan Ananth* Kai-Min Chung† Rolando L. La Placa‡

Abstract

We study the notion of zero-knowledge secure against quantum polynomial-time verifiers (referred to as quantum zero-knowledge) in the concurrent composition setting. Despite being extensively studied in the classical setting, concurrent composition in the quantum setting has hardly been studied.

We initiate a formal study of concurrent quantum zero-knowledge. Our results are as follows:

- **Bounded Concurrent QZK for NP and QMA**: Assuming post-quantum one-way functions, there exists a quantum zero-knowledge proof system for NP in the bounded concurrent setting. In this setting, we fix a priori the number of verifiers that can simultaneously interact with the prover. Under the same assumption, we also show that there exists a quantum zero-knowledge proof system for QMA in the bounded concurrency setting.
- **Quantum Proofs of Knowledge**: Assuming quantum hardness of learning with errors (QLWE), there exists a bounded concurrent zero-knowledge proof system for NP satisfying quantum proof of knowledge property.

  Our extraction mechanism simultaneously allows for extraction probability to be negligibly close to acceptance probability (*extractability*) and also ensures that the prover's state after extraction is statistically close to the prover's state after interacting with the verifier (*simulatability*).

  The seminal work of [Unruh EUROCRYPT'12], and all its followups, satisfied a weaker version of extractability property and moreover, did not achieve simulatability. Our result yields a proof of *quantum knowledge* system for QMA with better parameters than prior works.

*UC Santa Barbara. Email: prabhanjan@cs.ucsb.edu

†Academia Sinica, Taiwan. Email: kmchung@iis.sinica.edu.tw

‡MIT. Email: rlaplaca@mit.edu

Contents

1 Introduction

Zero-knowledge [GMR85] is one of the foundational concepts in cryptography. A zero-knowledge system for NP is an interactive protocol between a prover P, who receives as input an instance x and a witness w, and a verifier V who receives as input an instance x. The (classical) zero-knowledge property roughly states that the view of the malicious probabilistic polynomial-time verifier V^* generated after interacting with the prover P can be simulated by a PPT simulator, who doesn't know the witness w.

Protocol Composition in the Quantum Setting. Typical zero-knowledge proof systems only focus on the case when the malicious verifier is classical. The potential threat of quantum computers forces us to revisit this definition. There are already many works [ARU14, BJSW16, BG19, BS20, ALP20, VZ20, ABG+20], starting with the work of Watrous [Wat09], that consider the definition of zero-knowledge against verifiers modeled as a quantum polynomial-time algorithm; henceforth this definition will be referred to as quantum zero-knowledge. However, most of these works study quantum zero-knowledge only in the standalone setting. These constructions work under the assumption that the designed protocols work in isolation. That is, a standalone protocol is one that only guarantees security if the parties participating in an execution of this protocol do not partake in any other protocol execution. This is an unrealistic assumption. Indeed, the standalone setting has been questioned in the classical cryptography literature by a large number of works [DS98, DCO99, Can01, CLOS02, CF01, RK99, BS05, DNS04, PRS02, Lin03, Pas04, PV08, PTV14, GJO+13, CLP15, FKP19] that have focussed on designing cryptographic protocols that still guarantee security even when composed with the other protocols.

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

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

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

We ask the following question in this work:

Do there exist quantum zero-knowledge proof systems for NP and QMA that are secure under concurrent composition?

1.1 Our Contributions

Bounded Concurrent QZK for NP. We initiate a formal study of concurrent composition in the quantum setting. We work in the (weaker) bounded concurrent setting: where the prover interacts only with a bounded number of verifiers where this bound is fixed at the time of protocol specification. This setting has been well studied in the classical concurrent setting [Lin03, PR03, Pas04, PTW09]. Moreover, we note that the only other existing work that constructs zero-knowledge against multiple verifiers albeit in the parallel composition setting, namely [ABG+20][1], also works in the bounded setting. We prove the following.

Theorem 1 (Informal)**.** *Assuming the existence of post-quantum one-way functions[2], there exists a bounded concurrent quantum zero-knowledge proof system for NP. Additionally, our protocol is a public coin proof system.*

Our construction satisfies quantum black-box zero-knowledge[3].

Quantum Proofs of Knowledge. Our construction, described above, only satisfies the standard soundness guarantee. A more desirable property is quantum proof of knowledge. Roughly speaking, proof of knowledge states the following: suppose a malicious (computationally unbounded) prover can convince a verifier to accept an instance x with probability ε. Let the state of the prover at the end of interaction with the verifier be $|\Psi\rangle$[4]. Then there exists an efficient extractor, with black-box access to the prover, that can output a witness w for x with probability δ. Additionally, it also outputs a quantum state $|\Phi\rangle$. Ideally, we require the following two conditions to hold: (i) $|\varepsilon - \delta|$ is negligible and, (ii) the states $|\Psi\rangle$ and $|\Phi\rangle$ are close in trace distance; this property is also referred to as simulatability property. Unruh [Unr12] presented a construction of quantum proofs of knowledge; their construction satisfies (i) but not (ii). Indeed, the prover's state, after it interacts with the extractor, could be completely destroyed. Condition (ii) is especially important if we were to use quantum proofs of knowledge protocols as a sub-routine inside larger protocols, for instance in secure multiparty computation protocols.

Since Unruh's work, there have been other works that present constructions that satisfy both the above conditions but they demonstrate extraction only against *computationally bounded* adversaries [HSS11, BS20, ALP20]. Thus, it has been an important open problem to design quantum proofs of knowledge satisfying both of the above conditions.

We show the following.

[1] They achieve bounded parallel ZK under the assumption of quantum learning with errors and circular security assumption in constant rounds. While the notion they consider is sufficient for achieving MPC, the parallel QZK constructed by [ABG+20] has the drawback that the simulator aborts even if one of the verifiers abort. Whereas the notion of bounded concurrent QZK we consider allows for the simulation to proceed even if one of the sessions abort. On the downside, our protocol runs in polynomially many rounds.

[2] That is, one-way functions secure against quantum polynomial-time algorithms.

[3] The simulator has oracle access to the unitary V and $V^\dagger$, where V is the verifier.

[4] We work in the purified picture and thus we can assume that the output of the prover is a pure state.

Theorem 2 (Informal)**.** *Assuming quantum hardness of learning with errors (QLWE), there exists a bounded concurrent zero-knowledge proof system for NP satisfying quantum proofs of knowledge property.*

Contrary to the belief that Watrous oblivious rewinding technique is insufficient for achieving quantum proofs of knowledge, we do in fact make black-box use of Watrous rewinding lemma in conjunction with novel cryptographic tools to prove the above theorem. On the downside, our protocol runs in polynomially many rounds, while Unruh's technique works for existing 3-message Σ protocols.

Bounded Concurrent QZK for QMA. We also show how to extend our result to achieve bounded concurrrent zero-knowledge proof system for QMA [KSVV02] (a quantum-analogue of MA).

We show the following.

Theorem 3 (Informal)**.** *Assuming post-quantum one-way functions, there exists a bounded concurrent quantum zero-knowledge proof system for QMA.*

This improves upon the existing QZK protocols for QMA [BJSW16, BG19, CVZ20, BS20] which only guarantee security in the standalone setting.

Our construction considers a simplified version of the framewok of [BJSW16][5] and instantiates the underlying primitives in their protocol with bounded concurrent secure constructions. Specifically, we use our bounded concurrent QZK construction for NP for the above result.

We could combine the recent work of Coladangelo et al. [CVZ20] with our quantum proof of knowledge system for NP to obtain a proof of *quantum* knowledge system for QMA. This result yields better parameters than the one guaranteed in prior works [CVZ20, BG19]. Specifically, if the malicious prover convinces the verifier with probability negligibly close to 1 then the extractor (in our result) can extract a state that is negligibly close to the witness state whereas the previous works did not have this guarantee.

1.2 Technical Overview

We highlight the main ideas behind the constructions of the bounded concurrent QZK and quantum proof of knowledge. We omit the overview for the construction of bounded concurrent QZK for QMA.

1.2.1 Bounded Concurrent QZK for NP

Black Box QZK via Watrous Rewinding. The traditional rewinding technique that has been used to prove powerful results on classical zero-knowledge cannot be easily ported to the quantum setting. The fundamental reason behind this difficulty is the fact that to carry out rewinding, it is necessary to clone the state of the verifier. While cloning comes for free in the classical setting, the no-cloning theorem of quantum mechanics prevents us from being able to clone arbitrary states. Nonetheless, the seminal work of Watrous [Wat09] demonstrates that there are rewinding techniques that are amenable to the quantum setting. Watrous used this technique

[5]For the reader familiar with [BJSW16], we consider a coin-flipping protocol secure against explainable adversaries as against malicious adversaries as considered in [BJSW16].

to present the first construction of quantum zero-knowledge for NP. This technique is so powerful that all quantum zero-knowledge protocols known so far (including the ones with non-black box simulation [BS20, ABG+20]!) either implicitly or explicitly use this technique.

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

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

  The oblivious rewinding condition requires that this distribution should be independent of the state of the verifier. That is, this distribution should remain the same irrespective of the state of the verifier[6].

- **No-recording**: Before rewinding any round, the simulator could record (or remember) the transcript generated so far. This recorded transcript along with the rewound transcript will be used for simulation. For instance, in Goldreich and Kahan [GK96], the simulator first commits to garbage values and then waits for the verifier to decommit its challenges. The simulator then records the decommitments before rewinding and then changing its own commitments based on the decommitted values.

  The no-recording condition requires the following to hold: in order for the simulator to rewind from point i to point j $(i > j)$, the simulator needs to forget the transcript generated from j^{th} round to the i^{th} round. Note that the simulator of [GK96] does not satisfy the no-recording condition.

Once such a classical simulator is identified, we can then simulate quantum verifiers as follows: run the classical simulator and the quantum verifier[7] in superposition and then at the end of each round, measure the appropriate register to figure out whether to rewind or not. The fact that the distribution associated with the decision bits are independent of the verifier's state is used to argue that the state, after measuring the decision register, is not disturbed. Using this fact, we can then reverse the computation and go back to an earlier round. Once the computation is reversed (or rewound to an earlier round), the simulator forgets all the messages exchanged from the point – to which its being rewound to – until the current round.

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

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

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

al. [PTW09], who construct a bounded concurrent ZK protocol whose simulator satisfies the no-recording condition but not the oblivious rewinding condition.

In more detail, at every round, the simulator (as described in [PTW09]) makes a decision to rewind based on the message it receives. This means that the probability of whether the simulator rewinds any given round depends on the scheduling of the messages of the verifiers. Unfortunately, the scheduling itself could be a function of the state of the verifier. The malicious verifier could look at the first bit of its auxiliary state. If it is 0, it will ask the first session verifier to send a message and if it is 1, it will ask the second session verifier to send a message and so on. This means that a simulator's decision to rewind could depend on the state of the verifier.

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

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

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

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

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

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

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

The simulator partitions the $\ell_{\mathsf{prot}} \cdot Q$ messages into many blocks with each block being of a fixed size (we discuss the parameters later). The simulator then runs the verifier till the end of first block. At this point, it checks if this block contains a slot. Note that the verifier can stagger the

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

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

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

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

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

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

Instead, we argue the following:

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

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

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

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

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

1.2.2 Standalone Quantum Proofs of Knowledge

Towards building a bounded-concurrent QZK system satisfying quantum proof of knowledge property, we first focus on the standalone QZK setting. The quantum proof of knowledge property roughly says the following: for every unbounded prover convincing a verifier to accept an instance x with probability p, there exists an extractor that outputs a witness w with probability negligibly close to p and it also outputs a state $|\Phi\rangle$ that is close (in trace distance) to the output state of the real prover.

Our approach is to design a novel extraction mechanism that uses oblivious transfer to extract a bit from a quantum adversary.

Main Tool: Statistical Receiver-Private Oblivious Transfer. Our starting point is an oblivious transfer (OT) protocol [Rab05]. This protocol is defined between two entities: a sender and a receiver. The sender has two bits (m_0, m_1) and the receiver has a single bit b. At the end of the protocol, the receiver receives the bit m_b. The security against malicious senders (receiver

privacy) states that the sender should not be able to distinguish (with non-negligible probability) whether the receiver's bit is 0 or 1.

The security against malicious receivers (also called sender privacy) states that there is a bit b' such that the receiver cannot distinguish (with non-negligible probability) the case when the sender's input is (m_0, m_1) versus the setting when the sender's input is $(m_{b'}, m_{b'})$. This bit b' is a function of the sender's as well as the adversary's randomness.

We require receiver privacy to hold against unbounded senders while we require sender privacy needs to hold against quantum polynomial-time receivers. The reason we require receiver privacy against unbounded senders is because our goal is to design extraction mechanism against computationally unbounded provers.

We postpone discussing the construction of statistical receiver-private oblivious transfer. We will now see how to use this to achieve extraction.

One-bit Extraction with $\left(\frac{1}{2} \pm \mathsf{negl}\right)$-error. We begin with a naive attempt to design the extraction mechanism for extracting a single secret bit, say s. The prover and the verifier execute the OT protocol; prover takes on the role of the OT sender and the verifier takes on the receiver's role. The prover picks bits b and α uniformly at random and then sets the OT sender's input to be (s, α) if $b = 0$, otherwise if $b = 1$, it sets the OT sender's input to be (α, s). The verifier sets the receiver's bit to be 0. After the protocol ends, the prover sends the bit b. Note that if the bit b picked by the prover was 0 then the verifier can successfully recover s, else it recovers α.

We first discuss the classical extraction process. The quantum extractor runs the classical extractor in superposition as we did in the case of quantum zero-knowledge. The extraction process proceeds as follows: the extractor picks a bit $\widetilde{b}$ uniformly at random and sets $\widetilde{b}$ to be the receiver's bit in the OT protocol. By the statistical receiver privacy property of OT, it follows that the probability that the extractor succeeds in recovering s is negligibly close to $\frac{1}{2}$. Moreover, the success probability is independent of the initial state of the prover. This means that we can apply the Watrous rewinding lemma and amplify the success probability.

Malicious Provers: However, we missed a subtle issue: the malicious prover could misbehave. For instance, the prover can set the OT sender's input to be (r, r) and thus, not use the secret bit s at all.

We resolve this issue by additionally requiring the prover to prove to the verifier that one of its inputs in the OT protocol is the secret bit s. This is realized by using a quantum zero-knowledge protocol, denoted by Π.

Error amplification. A malicious verifier can successfully recover the secret s with probability $\frac{1}{2}$. To reduce the verifier's success probability, we execute the above process (i.e., first executing the OT protocol and then executing the ZK protocol) λ number of times, where λ is the security parameter. First, the prover will additively secret share the bit s into secret shares $sh_1, \ldots, sh_\lambda$. It also samples the bits $b_1, \ldots, b_\lambda$ uniformly at random. In the i^{th} execution, it sets the OT sender's input to be (sh_i, α_i) if $b_i = 0$, otherwise it sets the OT sender's input to be (α_i, sh_i), where α_i is sampled uniformly at random. After all the OT protocols are executed, the prover is going to prove using a QZK protocol Π, as considered above, that the messages in the OT protocols were correctly computed.

We first argue that even in this protocol, the extraction still succeeds with overwhelming probability. In each OT execution, the extractor applies Watrous rewinding, as before, to extract all the shares $sh_1, \ldots, sh_\lambda$. From this, it can recover s. All is left is to argue that this template satisfies quantum zero-knowledge property. It turns out that arguing this is challenging.

Challenges in Proving QZK and Distinguisher-Dependent Hybrids. We first define the simulator as follows:

- The simulator uses (α_i, α_i) as the sender's input in the i^{th} OT execution, where α_i is sampled uniformly at random.
- It then simulates the protocol Π.

To prove that the output distribution of the simulated world is computationally indistinguishable from the real world, we adopt a hybrid argument. The first hybrid, Hyb_1, corresponds to the real world. In the second hybrid, Hyb_2, simulate the protocol Π. The indistinguishability of Hyb_1 and Hyb_2 follows from the QZK property of Π. Next, we define the third hybrid, Hyb_3, that executes the simulator. To prove the indistinguishability of Hyb_2 and Hyb_3, we consider a sequence of intermediate hybrids, denoted by $\{\mathsf{Hyb}_{2.j}\}_{j\in[\lambda]}$. Using this sequence of hybrids, we change the inputs in all the λ OT executions one at a time. Finally, we define the third hybrid, Hyb_3, that corresponds to the ideal world. Proving the indistinguishability of the consecutive hybrids, $\mathsf{Hyb}_{2.j}$ and $\mathsf{Hyb}_{2.j+1}$, in this sequence turns out to be challenging.

The main issue is the following: suppose we are in the j^{th} intermediate hybrid $\mathsf{Hyb}_{2.j}$, for $j \leq \lambda$. At this point, we have changed the inputs to the first j OT executions and we are about to change the input to the $(j+1)^{th}$ OT. But what exactly are the inputs we are using for the first j OT executions? It is unclear whether we use the input (sh_i, sh_i) or the input (α_i, α_i), for $i \leq j$, in the i^{th} OT execution. Note that the OT security states that we can either switch the real sender's inputs to either (sh_i, sh_i) or (α_i, α_i), based on the sender's and the distinguisher's randomness. And hence, we define an *inefficient* intermediate hybrid, which is a function (not necessarily computable), that determines for every i, where $i \leq j$, whether to use (sh_i, sh_i) or (α_i, α_i). Moreover, *this hybrid depends on the distinguisher*, that distinguishes the two intermediate hybrids.

The indistinguishability of the consecutive pair of inefficient hybrids, say $\mathsf{Hyb}_{2.j}$ and $\mathsf{Hyb}_{2.j+1}$, is proven by a non-uniform reduction that receives as input the advice corresponding to the first j executions of OT, where the sender's inputs are correctly switched to either (sh_i, sh_i) or (α_i, α_i), for $i \leq j$. This in turn depends on the distinguisher distinguishing these two hybrids. Then, the reduction uses the $(j+1)^{th}$ OT execution in the protocol to break the sender privacy property of OT. If the two hybrids can be distinguished with non-negligible probability then the reduction can succeed with the same probability.

In the hybrid $\mathsf{Hyb}_{2.\lambda-1}$, we additionally include an abort condition: if the inputs in the first $\lambda - 1$ OT executions are all switched to (sh_i, sh_i) then we abort. We show that the probability that $\mathsf{Hyb}_{2.\lambda-1}$ aborts is negligible. This is necessary to argue that the verifier does not receive all the shares of the secret.

Note that only the intermediate hybrids, namely $\{\mathsf{Hyb}_{2.j}\}_{j\in[\lambda]}$, are inefficient, and in particular, the final hybrid Hyb_3 is still efficient.

Extraction of Multiple Bits. To design a quantum proof of knowledge protocol, we need to be able to extract not just one bit, but multiple bits. To achieve this, we design the prover as follows:

on input the witness w, it sequentially executes the above extraction template for each bit of the witness. That is, for every $i \in [\ell_w]$, where ℓ_w is the length of w, it additively secret shares w_i into the shares $(sh_{i,1}, \ldots, sh_{i,\lambda})$. It then invokes $\ell_w \cdot \lambda$ number of OT executions, where in the $(i,j)^{th}$ execution, it chooses the input $(sh_{i,j}, \alpha_{i,j})$ if $b_{i,j} = 0$, or the input $(\alpha_{i,j}, sh_{i,j})$ if $b_{i,j} = 1$, where $\alpha_{i,j}, b_{i,j}$ are sampled uniformly at random. Finally, it uses a QZK protocol to prove that it behaved honestly in the earlier OT executions.

The proofs of quantum proof of knowledge and the QZK properties follow along the same lines as the single-bit extraction case.

1.2.3 Construction of Statistical Receiver-Private OT with Post-Quantum Security

All that is left is to construct an oblivious transfer protocol that guarantees statistical indistinguishability property against malicious senders and indistinguishability property against QPT malicious receivers. We denote the protocol that we intend to construct to be Π_{SROT}.

Towards this, we start with another oblivious transfer protocol, denoted by Π_{SSOT}, that has its properties flipped. That is, Π_{SSOT} satisfies statistical indistinguishability property against malicious *receivers* and indistinguishability property against QPT malicious *senders*. The reason we start with this protocol is that we do know how to achieve this; Brakerski-Döttling [BD18] constructed such a protocol from QLWE.

Our approach is inspired from previous works [KKS18, GJJM20] that show how to construct statistical receiver-private OT from statistical sender-private OT.

Our first attempt to construct Π_{SROT} is the following:

- The sender of Π_{SSOT} samples a random bit $r \xleftarrow{\$} \{0,1\}$. It takes the role of the receiver in the underlying Π_{SROT}. It then sends the first message of Π_{SROT} with the receiver's message set to be r.
- The receiver of Π_{SSOT}, on input choice bit β, samples another random bit r'. It takes the role of the sender in the underlying protocol Π_{SSOT}. It then sends the sender's message in Π_{SSOT}, where the sender's input in Π_{SSOT} is set to be $(r', r' \oplus \beta)$.
- After the end of the execution of Π_{SSOT}, the sender on input (m_0, m_1), does the following: it recovers the message $\widetilde{r}$ from the underlying OT. It then sends $(\widetilde{r} \oplus m_0, \widetilde{r} \oplus r \oplus m_1)$ to the receiver.

If $\beta = 0$ then $\widetilde{r} = r'$ and so, the receiver can recover m_0. If $\beta = 1$ then $\widetilde{r} = r' \oplus r$ and so, the receiver can recover m_1.

The receiver privacy against computationally unbounded senders follows from the statistical sender privacy of the underlying two-round oblivious transfer protocol.

To prove sender privacy against QPT receivers, first let us make the previously described security notion more precise. The malicious receiver R^*, on input state $|\Psi\rangle$, interacts with the sender and produces an auxiliary state $|\widetilde{\Psi}\rangle$. During this interaction, the sender does not use (m_0, m_1). The sender uses (m_0, m_1) to compute the final round message. We define two games: in the first game, the adversary tries to distinguish (m_0, m_1) versus (m_0, m_0) and in the second game, the adversary tries to distinguish (m_0, m_1) versus (m_1, m_1). We say that oblivious transfer satisfies post-quantum computational sender privacy property if the malicious receiver cannot succeed in both the games with non-negligible advantage.

A natural approach to prove that the malicious receiver cannot win both the games is to extract the bit β from the malicious receiver; if $\beta = 0$ then the receiver will not be able to succeed in the second game if $\beta = 1$ then the receiver will not succeed in the first game. To ensure that we can extract the bit β from the receiver, we additionally introduce an extraction phase to the protocol.

Extraction Phase: To design the extraction phase, we use the same technique we introduced earlier. The main difference is that instead of using statistical receiver-private OT, we instead use a statistical sender-private OT for extraction.

In the extraction phase of Π_{SROT}, the sender and the receiver do the following:

- As before, the sender of Π_{SROT}, plays the role of the receiver of Π_{SSOT} and the receiver of Π_{SROT} plays the role of the sender of Π_{SSOT}.
- The sender does the following: it samples a bit b uniformly at random. It sets the Π_{SSOT}'s receiver's bit to be b.
- The receiver, on the other hand, samples $\alpha \xleftarrow{\$} \{0,1\}$ and sets the Π_{SSOT}'s sender's input to be (β, α) with probability $\frac{1}{2}$ and (α, β) with probability $\frac{1}{2}$.
- At the end of the execution of Π_{SSOT}, the receiver reveals the location of β – i.e., it sends 0 if (β, α) was used in Π_{SSOT} or it sends 1 if (α, β) was used.

Note that if the location matched with b then the sender can recover β, otherwise it cannot. With probability at most $\frac{1}{2}$, the sender can recover β. We can use the same error amplification technique (via secret sharing) introduced earlier to reduce the probability of success of the malicious sender to be negligible. On the other hand, we can design an extractor that uses Watrous rewinding, as mentioned earlier to recover the bit β with probability close to 1.

Template. Using the above ingredients, we now summarise the template to construct a statistical receiver-private oblivious transfer.

The sender, on input (m_0, m_1), and the receiver of Π_{SROT} on input β, do the following:

- The sender and the receiver execute the extraction phase described above. The receiver uses its bit β in the extraction phase.
- The sender and the receiver then execute Π_{SSOT}, where each party play the opposite role. The sender sets the input of the receiver in Π_{SSOT} to be r, where $r \xleftarrow{\$} \{0,1\}$ and the receiver sets the input of the sender in Π_{SSOT} to be $(r', r' \oplus \beta)$, where $r' \xleftarrow{\$} \{0,1\}$. After the end of the execution of Π_{SSOT}, the sender recovers $\widetilde{r}$.
- Of course, the receiver could have cheated and used a different β in both the extraction phase and in the execution of Π_{SSOT}. To ensure that the receiver does not cheat, we force the receiver to prove that it used β consistently. We use a computational argument system satisfying statistical zero-knowledge property for this step.
- Once the sender gets convinced that the receiver did not cheat, it sends $(\widetilde{r} \oplus m_0, \widetilde{r} \oplus r \oplus m_1)$ to the receiver.

Finally, we show how to implement computational argument system satisfying statistical zero-knowledge property from QLWE. The idea is to start with a statistical NIZK computational argument system in the CRS model and then generate the CRS using a coin flipping protocol.

1.3 Quantum Proof of Knowledge in the Bounded Concurrent Setting

Our construction of bounded concurrent quantum proof of knowledge is the same as the one described in Section 1.2.2, except that we instantiate Π using the bounded concurrent QZK protocol that we constructed in Section 1.2.1 [8].

However, proving the bounded concurrent QZK protocol turns out to be even more challenging than the standalone setting. To grasp the underlying difficulties, let us revisit the proof of QZK in Section 1.2.2. To prove the indistinguishability of the real and the ideal world, we first simulated the protocol Π. Since we are in the bounded concurrent setting, the simulator of Π is now simultaneously simulating multiple sessions of the verifier. Then using a sequence of intermediate hybrids, we changed the inputs used in the OT executions of all the sessions one at a time. However, in the bounded concurrent setting, the OT messages can be interleaved with QZK messages. This means that the simulator of QZK could be rewinding the OT messages along with the QZK messages. This makes it difficult to invoke the security of OT.

To reduce the indistinguishability of hybrids to breaking OT, we will carefully design the security reduction such that it does not rewind the blocks (the definition of a block is the same as the one described in Section 1.2.1) containing the messages of the OT protocol. This ensures that we can embed the messages exchanged with the external challenger without the fear of being rewound. Of course, we need to be cautious: the decision to not rewind a specific block could leak information about the private state of the verifier. To overcome this issue, for a block containing the OT messages, we perform a dummy rewind where the transcript of conversation in this block does not change. Thus, we can still interact with the external challenger using the messages in this block. Another issue that arises is that we might end up not rewinding as many blocks as the round complexity of the underlying OT protocol, which is polynomially many rounds. We show that the simulator of the bounded concurrent QZK we constructed in Section 1.2.1 can successfully simulate all the sessions even if polynomially many blocks are ignored.

1.4 Organization

- We present the necessary preliminaries – including the notation used in the paper, basics on quantum computing, definitions of commitments and Watrous rewinding lemma – in Section 2.
- We present the definitions of concurrent quantum zero-knowledge for both NP and QMA in Section 3. We also present the definition of quantum proof of knowledge in the same section.
- Then, we provide the construction of bounded concurrent QZK for NP in Section 4.
- We then focus on construct a quantum proof of knowledge in the bounded concurrent QZK setting. The main tool used in this construction is an oblivious transfer protocol; we present the definition and the construction of this oblivious transfer protocol in Section 5.

  In Section 6, we present the construction of quantum proof of knowledge in two steps. First we present a construction in the standalone setting. We then extend this construction to the bounded concurrent setting.
- Finally, we present a construction of bounded concurent QZK for QMA in Section 7.

[8] We emphasize that we use the specific bounded concurrent QZK protocol that we constructed earlier and we do not know how to provide a generic transformation.

2 Preliminaries

We denote the security parameter by λ. We assume basic familiarity of cryptographic concepts.

We denote (classical) computational indistiguishability of two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ by $\mathcal{D}_0 \approx_{c,\varepsilon} \mathcal{D}_1$. In the case when ε is negligible, we drop ε from this notation. We denote the process of an algorithm A being executed on input a sample from a distribution $\mathcal{D}$ by the notation $A(\mathcal{D})$.

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

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

2.1 Notation and General Definitions

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

Quantum states and channels. Let $\mathcal{H}$ be any finite Hilbert space, and let $L(\mathcal{H}) := \{\mathcal{E} : \mathcal{H} \to \mathcal{H}\}$ be the set of all linear operators from $\mathcal{H}$ to itself (or endomorphism). Quantum states over $\mathcal{H}$ are the positive semidefinite operators in $L(\mathcal{H})$ that have unit trace.

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

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

Quantum Computational Indistinguishability. We define computational indistinguishability; we borrow the following definition from [Wat09]. Roughly, the below definition states that two collections of quantum states $\{\rho_x\}$ and $\{\sigma_x\}$ are computationally indistinguishable if any quantum distinguisher, running in time polynomial in $|x|$, cannot distinguish ρ_x from σ_x, where x is sampled from some distribution. Moreover, the computational indistinguishability should hold even if the distinguisher has quantum advice (that might be entangled with ρ_x and σ_x).

Definition 4 (Computational Indistinguishability of Quantum States)**.** *Let I be an infinite subset $I \subset \{0,1\}^*$, let $p : \mathbb{N} \to \mathbb{N}$ be a polynomially bounded function, and let ρ_x and σ_x be $p(|x|)$-qubit states. We say that $\{\rho_x\}_{x\in I}$ and $\{\sigma_x\}_{x\in I}$ are* ***quantum computationally indistinguishable***

collections of quantum states *if for every QPT $\mathcal{E}$ that outputs a single bit, any polynomially bounded $q : \mathbb{N} \to \mathbb{N}$, and any auxiliary collection of $q(|x|)$-qubits states $\{\nu_x\}_{x\in I}$, and for all (but finitely many) $x \in I$, we have that*

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

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

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

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

Interactive Models. We model an interactive protocol between a prover, P, and a verifier, V, as follows. There are 2 registers R_P and R_V corresponding to the prover's and the verifier's private registers, as well as a message register, R_{M}, which is used by both P and V to send messages. In other words, both prover and verifier have access to the message register. We denote the size of a register R by $|\mathsf{R}|$ – this is the number of bits or qubits that the register can store. There are 3 different notions of interactive computation.

1. **Classical protocol:** An interactive protocol is classical if R_P, R_V, and R_{M} are classical, and P and V can only perform classical computation.
2. **Quantum protocol with classical messages:** An interactive protocol is quantum with classical messages if either one of R_P or R_V is a quantum register, and R_{M} is classical. P and V can perform quantum computations if their respective private register is quantum, but they can only send classical messages.
3. **Quantum protocol:** R_P, R_V, and R_{M} are all quantum registers. The prover performs quantum operations on $\mathrm{R}_P \otimes \mathrm{R}_{\mathsf{M}}$ and the verifier performs quantum operations on $\mathrm{R}_V \otimes \mathrm{R}_{\mathsf{M}}$.

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

Notation. We use the following notation in the rest of the paper.

- $\langle P, V\rangle$ denotes the interactive protocol between the QPT algorithms P and V. We denote the $\langle P(y_1), V(y_2)\rangle$ to be (z_1, z_2), where z_1 is the prover's output and z_2 is the verifier's output. Sometimes we omit the prover's output and write this as $z \leftarrow \langle P(y_1), V(y_2)\rangle$ to indicate the output of the verifier to be z.
- $\mathsf{View}_V(\langle P(y_1), V(y_2)\rangle)$ denotes the view of the QPT algorithm V in the protocol Π, where y_1 is the input of P and y_2 is the input of V. In the classical case, the view includes the output of V and the transcript of the conversation. In a quantum protocol, the view is the output on registers $\mathrm{R}_{\mathsf{M}} \otimes \mathrm{R}_V$. Similarly, we can define the view of P to be $\mathsf{View}_P(\langle P(y_1), V(y_2)\rangle)$ that includes the output on the registers $\mathrm{R}_P \otimes \mathrm{R}_{\mathsf{M}}$.

2.2 Statistically Binding and Quantum-Concealing Commitments

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

Definition 5 (Statistically Binding)**.** *A two-message commitment scheme between a committer (*Comm*) and a receiver (*R*), both running in probabilistic polynomial time, is said to satisfy statistical binding property if the following holds for any adversary* $\mathcal{A}$*:*

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

for some negligible function negl*.*

Definition 6 (Quantum-Concealing)**.** *A commitment scheme* Comm *is said to be quantum concealing if the following holds. Suppose* $\mathcal{A}$ *be a non-uniform QPT algorithm and let* $\mathbf{r}$ *be the message generated by* $\mathcal{A}(1^\lambda)$*. We require that* $\mathcal{A}$ *cannot distinguish the two distributions,* $\{\mathsf{Comm}(1^\lambda,\mathbf{r},x_1)\}$ *and* $\{\mathsf{Comm}(1^\lambda,\mathbf{r},x_2)\}$*, for any two inputs* x_1,x_2*.*

Remark 7. *We only considered two message protocols in the above definition for simplicity.*

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

2.3 Watrous Rewinding Lemma

We first state the following lemma due to Watrous [Wat09].

Lemma 8 (Watrous Rewinding Lemma)**.** *Suppose* Q *be a quantum circuit acting on* $n+k$ *qubits such that for every* n*-qubit state* $|\psi\rangle$*, the following holds:*

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

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

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

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

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

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

3 Concurrent Quantum ZK Proof Systems: Definitions

In Section 3.1, we define the notion of bounded concurrent QZK for NP. In Section 3.2, we define the notion of bounded concurrent ZK for QMA. We present the definition of quantum proof of knowledge in Section 3.3.

3.1 Bounded Concurrent QZK for NP

We start by recalling the definitions of the completeness and soundness properties of a classical interactive proof system.

Definition 9 (Proof System)**.** *Let* Π *be an interactive protocol between a classical PPT prover* P *and a classical PPT verifier* V*. Let* $\mathcal{R}(\mathcal{L})$ *be the NP relation associated with* Π*.*

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

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

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

for some negligible function negl*.*

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

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

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

for some negligible function negl*.*

Remark 10. *In Section 6, we define a stronger property called proof of knowledge property that subsumes the soundness property.*

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

Definition 11 (Q-session Quantum Adversary)**.** *Let* $Q \in \mathbb{N}$*. Let* Π *be an interactive protocol between a (classical) PPT prover and a (classical) PPT verifier* V *for the relation* $\mathcal{R}(\mathcal{L})$*. Let* $(x, w) \in \mathcal{R}(\mathcal{L})$*. We say that an adversarial non-uniform QPT verifier* V^* *is a* Q**-session adversary** *if it invokes* Q *sessions with the prover* $P(x, w)$*.*

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

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

P_i responds to msg_i. If $\mathsf{msg}_i = \mathsf{N/A}$ then it sets $\mathsf{msg}'_i = \mathsf{N/A}$. If V_i^* has sent the messages in the correct order[9]*, then P_i applies the next message function on its own private state and msg_i to obtain z' and sets $\mathsf{msg}'_i = (t+1, z')$. Otherwise, it sets $\mathsf{msg}'_i = (\bot, \bot)$. Finally, V^* receives $\Big((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q)\Big)$. In total, V^* exchanges $\ell_{\mathsf{prot}} \cdot Q$ number of messages, ℓ_{prot} is the number of the messages in the protocol.*

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

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

Definition 12 (Concurrent Quantum ZK for NP)**.** *An interactive protocol Π between a (classical) PPT prover P and a (classical) PPT verifier V for a language $\mathcal{L} \in \mathsf{NP}$ is said to be a* **concurrent quantum zero-knowledge (QZK) proof system** *if it satisfies completeness, unconditional soundness and the following property:*

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

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

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

Definition 13 (Bounded Concurrent Quantum ZK for NP)**.** *Let $Q \in \mathbb{N}$. An interactive protocol between a (classical) probabilistic polynomial time (in Q) prover P and a (classical) probabilistic polynomial time (in Q) verifier V for a language $\mathcal{L} \in \mathsf{NP}$ is said to be a* **bounded concurrent quantum zero-knowledge (QZK) proof system** *if it satisfies completeness, unconditional soundness and the following property:*

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

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

3.2 Bounded Concurrent QZK for QMA

We start by recalling the definitions of completeness and soundness properties of a quantum interactive proof system for promise problems.

Definition 14 (Interactive Quantum Proof System for QMA)**.** *Π is an interactive proof system between a QPT prover P and a QPT verifier V, associated with a promise problem $A = A_{yes} \cup A_{no} \in \mathsf{QMA}$, if the following two conditions are satisfied.*

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

- **Completeness**: *For all $x \in A_{yes}$, there exists a* $\mathrm{poly}(|x|)$*-qubit state* $|\psi\rangle$ *such that the following holds:*

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

for some negligible function negl*.*

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

- **Soundness**: *For every prover P^* (possibly computationally unbounded), every $x \in A_{no}$, the following holds:*

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

for some negligible function negl*.*

To define bounded concurrent QZK for QMA, we first the notion of Q-sesssion adversaries.

Definition 15 (Q-session adversary for QMA)**.** *Let $Q \in \mathbb{N}_{\geq 1}$. Let Π be a quantum interactive protocol between a QPT prover and a QPT verifier V for a* QMA *promise problem $A = A_{yes} \cup A_{no}$. We say that an adversarial non-uniform QPT verifier V^* is a Q-session adversary if it invokes Q sessions with the prover $P(x, |\psi\rangle)$.*

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

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

P_i responds to msg_i. If $\mathsf{msg}_i = \mathsf{N/A}$ then it sets $\mathsf{msg}_i' = \mathsf{N/A}$. If V_i^* has sent the messages in the correct order, P_i applies the next message function (modeled as a quantum circuit) on $|\Psi_{t,i}\rangle$ and its private quantum state to obtain ρ' and sets $\mathsf{msg}_i' = (t+1, \rho')$. Otherwise, it sets $\mathsf{msg}_i' = (\perp, \perp)$. Finally, V^* receives $\left((1, \mathsf{msg}_1'), \ldots, (Q, \mathsf{msg}_Q')\right)$. In total, V^* exchanges $\ell_{\mathsf{prot}} \cdot Q$ number of messages, where ℓ_{prot} is the number of the messages in the protocol.

Remark 16. *To invoke Q different sessions, we assume that the prover has Q copies of the witness state.*

Remark 17. *We assume, without loss of generality, the prover will measure the appropriate registers to figure out the round number for each verifier. This is because the malicious verifier can always send the superposition of the ordering of messages.*

We define quantum ZK for QMA in the bounded concurrent setting below.

Definition 18 (Bounded Concurrent QZK for QMA)**.** *Let $Q \in \mathbb{N}$. An interactive protocol Π between a QPT prover P (running in time polynomial in Q) and a QPT verifier V (running in time polynomial in Q) for a* QMA *promise problem $A = A_{yes} \cup A_{no}$ if it satisfies completeness, unconditional soundness and the following property:*

- **Bounded Concurrent Quantum Zero-Knowledge:** *For every sufficiently large $\lambda \in \mathbb{N}$, for every Q-session QPT adversary V^*, there exists a QPT simulator* Sim *such that for every $x \in A_{yes}$ and any witness $|\psi\rangle$,* $\mathrm{poly}(\lambda)$*-qubit advice ρ, the following holds:*

$$\{\mathsf{View}_{V^*}\langle P(x, |\psi\rangle), V^*(x, \rho)\rangle\} \approx_{\mathcal{Q}} \{\mathsf{Sim}(x, \rho)\}$$

3.3 Quantum Proofs of Knowledge

We present the definition of quantum proof of knowledge; this is the traditional notion of proof of knowledge, except that the unbounded prover could be a quantum algorithm and specifically, its intermediate states could be quantum states.

Definition 19 (Quantum Proof of Knowledge)**.** *We say that an interactive proof system (P, V) for a NP relation $\mathcal{R}$ satisfies (ε, δ)-proof of knowledge property if the following holds: suppose there exists a malicious (possibly computationally unbounded prover) P^* such that for every x, quantum state ρ such that:*

$$\mathsf{Pr}\left[(\widetilde{\rho}, \mathsf{decision}) \leftarrow \langle P^*(x,\rho), V(x)\rangle \bigwedge \mathsf{decision} = \mathsf{accept}\right] = \varepsilon$$

Then there exists a quantum polynomial-time extractor Ext*, such that:*

$$\mathsf{Pr}\left[(\widetilde{\rho}, w) \leftarrow \mathsf{Ext}\,(x,\rho) \bigwedge \mathsf{decision} = \mathsf{accept}\right] = \delta$$

Moreover, we require $T(\rho, \widetilde{\rho}) = \mathsf{negl}(|x|)$, where $T(\cdot,\cdot)$ is trace distance and negl *is a negligible function.*

We drop (ε, δ) from the notation if $|\delta - \varepsilon| \leq \mathsf{negl}(|x|)$, for a negligible function negl*.*

Remark 20 (Comparison with Unruh's Proof of Knowledge [Unr12])**.** *Our definition is a special case of Unruh's quantum proof of knowledge definition. Any proof system satisfying our definition is a quantum proof of knowledge system (according to Unruh's definition) with knowledge error κ for any κ. Moreover, in Unruh's definition, the extraction probability is allowed to be polynomially related to the acceptance probability whereas in our case, the extraction probability needs to be negligibly close to the acceptance probability.*

Definition 21 (Concurrent Quantum ZK PoK)**.** *We say that a concurrent (resp., bounded) quantum ZK is a concurrent (resp., bounded) QZKPoK if it satisfies proof of knowledge property.*

3.4 Intermediate Tool: Quantum Witness-Indistinguishable Proofs for NP

For our construction, we use a proof system that satisfies a property called witness indistinguishability. We recall this notion below.

Definition 22 (Quantum Witness-Indistinguishability)**.** *An interactive protocol between a (classical) PPT prover P and a (classical) PPT verifier V for a language $L \in \mathsf{NP}$ is said to be a* ***quantum witness-indistinguishable proof system*** *if in addition to completeness, unconditional soundness, the following holds:*

- **Quantum Witness-Indistinguishability***: For every $x \in \mathcal{L}$ and w_1, w_2 such that $(x, w_1) \in \mathcal{R}(\mathcal{L})$ and $(x, w_2) \in \mathcal{R}(\mathcal{L})$, for every QPT verifier V^* with $\mathrm{poly}(\lambda)$-qubit advice ρ, the following holds:*

$$\{\mathsf{View}_{V^*}\left(\langle P(x,w_1), V^*(x,\rho)\rangle\right)\} \approx_{\mathcal{Q}} \{\mathsf{View}_{V^*}\left(\langle P(x,w_2), V^*(x,\rho)\rangle\right)\}$$

Instantiation. By suitably instantiating the constant round WI argument system of Blum [Blu86] with statistically binding commitments (which in turn can be based on post-quantum one-way functions [Nao91]), we achieve a 4 round quantum WI proof system for NP. Moreover, this proof system is a public-coin proof system; that is, the verifier's messages are sampled uniformly at random.

4 Bounded Concurrent QZK for NP

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

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

- Statistically-binding and quantum-concealing commitment protocol (see Section 2.2), denoted by $(\mathsf{Comm}, \mathsf{R})$.
- Four round quantum witness-indistinguishable proof system Π_{WI} (Definition 22). The relation associated with Π_{WI}, denoted by $\mathcal{R}_{\mathsf{WI}}$, is defined as follows:

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

4.1 Construction

We describe the construction in Figure 1.

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

Theorem 23. *Assuming the security of* $(\mathsf{Comm}, \mathsf{R})$ *and* Π_{WI}*, the construction in Figure 1 is a bounded concurrent QZK proof system.*

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

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

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

Definition 24 (Matched Slot)**.** *We say that a slot is matched if the bit committed by P equals V's response.*

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

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

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

// We refer to the P's and V's message in one of the executions as a slot.

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

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

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

Figure 1: Construction of classical bounded concurrent ZK for NP.

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

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

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

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

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

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

with probability negligibly close to 1.

4.2 Quantum Zero-Knowledge

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

Parameters.

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

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

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

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

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

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

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

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

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

   - If the t^{th} message is a Stage 1 message from the prover responding to the first session message of a slot, apply the following operation in superposition over the receiver's message[10]:

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

     while leaving all the other registers intact. Note that we can prepare this state efficiently by first applying $H^{\otimes(\lambda+1)}$ to the $\mathbf{R}_t$ register followed by applying Comm in superposition and storing the output in the $\mathbf{Sim}_t$ register.
   - If the t^{th} message is a verifier's message, apply V^* on the registers corresponding to the transcript of the protocol until the t^{th} message (i.e. registers $\{(\mathbf{Sim}_i)\}_{i\le t}, \{\mathbf{Ver}_i\}_{i<t}$, $\mathbf{Aux}$) and on $\mathbf{Aux}$ register that corresponds to the verifier's private state, and output in the register $\mathbf{Ver}_t$.
   - If the t^{th} message is a Stage 2 message from the prover responding to the i^{th} WI initiated by the verifier (this just means that so far, $(i-1)$ WIs from $(i-1)$ sessions have already been initiated in the transcript), let w be the string in the register $\mathbf{W}$. Let c_i be the bit in register $\mathbf{B}_i$. If $c_i = 1$, use w as the witness to the WI proof. If $c_i = 0$, check if at least $\frac{\ell_{\mathsf{slot}}}{2} + Q^4\lambda$ matched slots corresponding to the session whose WI message is being computed. If so, compute the WI of Stage 2 using these matched slots. Otherwise, abort and output $\perp$ on register $\mathbf{Sim}_t$[11].

   (b) Let T contain the transcript of messages sent in block B_j along with the input and randomness used by the simulator to create these messages (i.e. the string stored in the registers $\{(\mathbf{R}_t, \mathbf{Sim}_t, \mathbf{Ver}_t)\}_{i\in B_j}$), and let $\mu(T)$ denote the set of all slots that are inside B_j in the transcript T. In superposition, perform the unitary U' defined below. Let I be a register containing a subset of qubits in $\mathbf{X}$. We omit the subscripts of the registers associated with the transcript T.

$$\begin{aligned} & U'|T\rangle|0\rangle_{\mathbf{M}_j}|0\rangle_{\mathbf{Dec}}|0^{\otimes|I|}\rangle_I \\ \approx\ & |T\rangle \otimes \left(\frac{1}{\sqrt{|\mu(T)|}} \sum_{(\mathbf{c},b')\in\mu(T)} |\mathbf{c}, b'\rangle_{\mathbf{M}_j}|1 \oplus \mathsf{Match}(T, \mathbf{c}, b'))\rangle_{\mathbf{Dec}}|\phi_{\mathbf{c},b'}\rangle_I\right) \text{ if } \mu(T) \neq \emptyset \\ =\ & |T\rangle|0\rangle_{\mathbf{M}_j}|+\rangle_{\mathsf{Dec}}|0^{\otimes|I|}\rangle_I \text{ if } \mu(T) = \emptyset \end{aligned}$$

[10] We assume without loss of generality that the length of the sender's randomness in the commitment scheme is λ.

[11] It may not be clear why we need this register. However, having this register would help us in the presentation of the hybrids.

where $\mathsf{Match}(T, \mathbf{c}, b') = 1$ if $\mathbf{c}$ is a commitment to b' and 0 otherwise. $|\phi_{\mathbf{c},b'}\rangle$ is some auxiliary state. Note that T, in addition to containing the transcript of messages exchanged in B_j, also contains the input and the randomness used by the simulator to create these messages.

By $\approx$, we mean the following: we say $|\phi_0\rangle \approx |\phi_1\rangle$ if both the states $|\phi_0\rangle$ and $|\phi_1\rangle$ are exponentially close (in trace distance) to each other. To see how we can obtain the above state, the unitary U' creates uniform superpositions over $[1], [2], \ldots, [|T|]$. Then, U' determines $\mu(T)$ and uses the uniform superposition over $[|\mu(T)|]$ to create a uniform superposition over $|\mathbf{c}, b'\rangle$.

Let $W_j = \mathsf{Amplifier}\left(U_j^{V^*}\right)$; where $\mathsf{Amplifier}$ is the circuit guaranteed by Lemma 8. Simulator computes $|\Psi_{j,\perp}\rangle = W_j|\Psi_{j-1,\perp}\rangle$.

3. For all $t \in \{1, ..., \ell_{\mathsf{prot}} \cdot Q\}$, measure all the $\mathbf{Sim}_t$ and $\mathbf{Ver}_t$ registers in the computational basis, and output the measurement outcomes along with the resulting state in the $\mathbf{Aux}$ register. In other words, let Y be the measurement outcome after measuring the registers corresponding to the protocol's transcript. Then, output Y along with

$$\widetilde{\rho} = \frac{\mathsf{Tr}_{\overline{\mathsf{Aux}}}\left[\Pi_Y |\Psi_{L,\perp}\rangle\langle\Psi_{L,\perp}| \Pi_Y\right]}{\mathsf{Tr}\left[\Pi_Y |\Psi_{L,\perp}\rangle\langle\Psi_{L,\perp}| \Pi_Y\right]}$$

where Π_Y projects the registers $(\mathbf{Sim}_1, \mathbf{Ver}_1, \ldots, \mathbf{Sim}_{\ell_{\mathsf{prot}} \cdot Q}, \mathbf{Ver}_{\ell_{\mathsf{prot}} \cdot Q})$ onto Y. By $\mathsf{Tr}_{\overline{\mathsf{Aux}}}[\cdot]$, we mean the operation of tracing out all the registers except Aux.

Remark 25. *Using the description of the unitaries $U_i^{V^*}$ as above, note that for any $(x, w) \in \mathcal{R}(\mathcal{L})$, if the prover and the verifier ran their protocol in superposition (and never measured), their combined output would be $U_L^{V^*} \cdots U_1^{V^*} \left(I \otimes X^{\otimes_{j\in[Q]} \mathbf{B}_j}\right) |\Psi_{0,w}\rangle$, where $X^{\otimes_{j\in[Q]} \mathbf{B}_j}$ is Pauli X's applied to the $\{\mathbf{B}_i\}_{i\in[Q]}$ registers and I is the identity operator applied on the rest of the registers. On the other hand, the state obtained by the simulator just before the final partial measurement is $W_L \cdots W_1 |\Psi_{0,\perp}\rangle$.*

We will show that for any verifier's auxiliary state $|\Psi\rangle$, the output of this simulator is indistinguishable from the output of the verifier when interacting with the honest prover.

Lemma 26. *For any $(x, w) \in \mathcal{R}(\mathcal{L})$, and for any auxiliary $\mathrm{poly}(\lambda)$-qubits state[12] $|\Psi\rangle$, the output of $\mathsf{Sim}^{V^*}(1^\lambda, x, |\Psi\rangle)$ is computationally indistinguishable from $\mathsf{View}_{V^*}\langle P(x, w), V^*(x, |\Psi\rangle)\rangle$.*

Proof. We will proceed with a series of hybrids.

$\underline{\mathsf{Hyb}_0}$: The output of this hybrid is the output of the verifier when interacting with the honest prover.

$\underline{\mathsf{Hyb}_1}$: Define a hybrid simulator $\mathsf{Hyb}_1.\mathsf{Sim}^{V^*}(x, w, |\Psi\rangle)$ that behaves like the honest prover, but performs the execution of the prover and the verifier in all the sessions in superposition. This simulator first prepares the state $U_L^{V^*} \ldots U_1^{V^*} \left(I \otimes X^{\otimes_{j\in[Q]} \mathbf{B}_j}\right) |\Psi_{0,w}\rangle$, then, it measures the registers corresponding to the transcript (that is, $\{(\mathbf{Sim}_t, \mathbf{Ver}_t)\}_{t\in[\ell_{\mathsf{prot}}]]}$) and outputs the measurement

[12] We can assume without of generality, via the process of purification, that the input state of the verifier is a pure state.

outcome along with the resulting verifier's private state.

The distribution of outputs in Hyb_0 and Hyb_1 are identical, since measurements can be deferred to the end by the *principle of deferred measurement*.

$\mathsf{Hyb}_{2.i}$, for $i = 1$ to L: Consider the following sequence of hybrid simulators, $\mathsf{Hyb}_{2.i}.\mathsf{Sim}^{V^*}(x, w)$, that behaves like $\mathsf{Hyb}_1.\mathsf{Sim}^{V^*}(x, w)$, but perform Watrous' rewinding on blocks $B_1, \ldots, B_i$. In other words, instead of performing the unitary $U_i^{V^*}$, it performs $W_i = \mathsf{Amplifier}\left(U_i^{V^*}\right)$. This means that $\mathsf{Hyb}_{2.i}.\mathsf{Sim}^{V^*}(x, w, |\Psi\rangle)$ computes:

$$U_L^{V^*} \cdots U_{i+1}^{V^*} W_i \cdots W_1 \left(I \otimes X^{\otimes_{j\in[Q]}\mathbf{B}_j}\right) |\Psi_{0,w}\rangle$$

The final partial measurement is performed as in the previous hybrid.

We defer the proof of the following claim to Section 4.2.1.

Claim 27. *Assuming that* Comm *satisfies hiding against quantum polynomial-time adversaries, the output distributions of the verifier in* $\mathsf{Hyb}_{2.i}$ *is computationally indistinguishable from the output distribution of the verifier in* $\mathsf{Hyb}_{2.i+1}$.

$\mathsf{Hyb}_{3.i}$ for $i \in [Q]$: Define a hybrid simulator $\mathsf{Hyb}_{3.i}.\mathsf{Sim}^{V^*}$ that behaves like $\mathsf{Hyb}_{2.L}$ except that it does not applies the initial bit flip X on registers $\mathbf{B}_k$ for all $k \leq i$. Formally, hybrid $\mathsf{Hyb}_{3.i}$ computes:

$$W_L W_{L-1} \ldots W_1 \left(I \otimes X^{\otimes_{j>i}\mathbf{B}_j}\right) |\Psi_{0,w}\rangle.$$

This change means that in Stage 2 of the protocol, for the sessions that initiate the first i WI protocols, the hybrid simulator $\mathsf{Hyb}_{3.i}.\mathsf{Sim}$ will use matched slots instead of the actual witness to compute the WI proof. For the rest of the sessions, the hybrid simulator still uses the witness w to produce the WI proof.

We defer the proof of the following claim to Section 4.2.2.

Claim 28. *Assuming the witness-indistinguishability property of* Π_{WI}*, the output distributions of the hybrids* $\mathsf{Hyb}_{3.i}$ *and* $\mathsf{Hyb}_{3.i+1}$ *are computationally indistinguishable.*

Hyb_4: The output of this hybrid is the output of the simulator.

The output distributions of $\mathsf{Hyb}_{3.L}$ and Hyb_4 are identical. □

□

4.2.1 Proof of Claim 27

We prove this in the following steps:

1. First, we reduce proving the indistinguishability of $\mathsf{Hyb}_{2.i}$ and $\mathsf{Hyb}_{2.i-1}$ to proving the following statement: the following two distributions are computationally indistinguishable.

   - $\mathcal{D}_1$: Measure the $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t\leq i}$ registers at the end of execution of the block B_i in $\mathsf{Hyb}_{2.i-1}$ and output the measurement outcome along with the residual state in the register $\mathbf{Aux}$.

- $\mathcal{D}_2$: Measure the $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t\leq i}$ registers at the end of execution of the block B_i in $\mathsf{Hyb}_{2.i}$ and output the measurement outcome along with the residual state in the register $\mathbf{Aux}$.

2. Next, we show the indistinguishability of $\mathcal{D}_1$ and $\mathcal{D}_2$ by using Watrous rewinding and quantum-concealing property of the commitments.

Bullet 1 follows from the fact that the registers $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t\leq i}$ are never written upon after the execution of Block B_i and hence measurement operators applied on these registers in the end commute with the unitaries applied after the execution of B_i.

For Bullet 2, we first make some observations on the state obtained in $\mathsf{Hyb}_{2.i}$ after applying Watrous rewinding.

Applying Watrous Rewinding. Let $|\Psi^{i-1}_{0,w}\rangle = W_{i-1}\dots W_1\left(I\otimes X^{\otimes_{j\in[Q]}\mathbf{B}_j}\right)|\Psi_{0,w}\rangle$. Without loss of generality, we can write $U_i^{V^*}|\Psi^{i-1}_{0,w}\rangle$ the following way:

$$U_i^{V^*}|\Psi^{i-1}_{0,w}\rangle \quad = \quad \sqrt{q}|\Phi_{i,\mathsf{noslot}}\rangle|+\rangle_{\mathbf{Dec}} + \sqrt{(1-q)}|\Phi_{i,\mathsf{slot}}\rangle$$

where:

- $|\Phi_{i,\mathsf{noslot}}\rangle$ is a superposition of all the transcripts containing no slot in the i^{th} block B_i. This is defined on all the registers except the $\mathbf{Dec}$ register.
- $|\Phi_{i,\mathsf{slot}}\rangle$ is a superposition of all the transcripts containing at least one slot in the i^{th} block B_i. This is defined on all the registers.

Furthermore, $|\Phi_{i,\mathsf{slot}}\rangle$ can be written as $\sqrt{p(\Phi_{i,\mathsf{slot}})}|\Phi_{\mathsf{yes}}\rangle|0\rangle_{\mathbf{Dec}}+\sqrt{1-p(\Phi_{i,\mathsf{slot}})}|\Phi_{\mathsf{no}}\rangle|1\rangle_{\mathbf{Dec}}$, for some states $|\Phi_{\mathsf{yes}}\rangle$, $|\Phi_{\mathsf{no}}\rangle$ and some function $p(\cdot)$. We first claim the following.

Claim 29. *Assuming quantum concealing property of* $(\mathsf{Comm},\mathsf{R})$*, the following holds:*

$$\left|p(\Phi_{i,\mathsf{slot}}) - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$$

Proof. By the quantum-concealing property of Comm, any QPT adversary $\mathcal{A}$, with auxiliary state $|\Phi\rangle$, can win the following game with probability at most negligibly close to $\frac{1}{2}$: given a commitment $c = \mathsf{Comm}(b;r)$, where $b \xleftarrow{\$} \{0,1\}$ and $r \xleftarrow{\$} \{0,1\}^\lambda$, we say that $\mathcal{A}$ wins if it outputs $b' = b$.

We execute the above experiment in superposition:

- $\mathcal{A}$ sends the first commitment message, $\mathbf{r}$.
- Challenger prepares the following state (omitting the register containing $\mathbf{r}$):

$$\frac{1}{\sqrt{2^{\lambda+1}}}\sum_{b\in\{0,1\},r\in\{0,1\}^\lambda}|b,r\rangle_X|\mathsf{Comm}(1^\lambda,b,\mathbf{r};r)\rangle_Y|0\rangle_Z|\Phi\rangle_{\mathbf{Aux}}|0\rangle_{\mathbf{Dec}}$$

- $\mathcal{A}$ is computed (over the registers $Y, Z, \mathbf{Aux}$) in superposition:

$$\frac{1}{\sqrt{2^{\lambda+1}}} \sum_{b\in\{0,1\},r\in\{0,1\}^\lambda} |b,r\rangle_X |\mathsf{Comm}(1^\lambda,\mathbf{r},b;r)\rangle_Y |\mathcal{A}(\mathsf{Comm}(b;r))\rangle_Z |\Phi'\rangle_{\mathbf{Aux}} |0\rangle_{\mathbf{Dec}}$$

- The challenger computes the following:

$$\frac{1}{\sqrt{2^{\lambda+1}}} \sum_{b\in\{0,1\},r\in\{0,1\}^\lambda} |b,r\rangle_X |\mathsf{Comm}(1^\lambda,\mathbf{r},b;r)\rangle_Y |\mathcal{A}(\mathsf{Comm}(1^\lambda,\mathbf{r},b;r))\rangle_Z |\Phi'\rangle_{\mathbf{Aux}} |b\oplus\mathcal{A}(\mathsf{Comm}(b;r))\rangle_{\mathbf{Dec}}$$

We can rewrite the above state as follows:

$$\sqrt{p'}|\phi_0\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p'}|\phi_1\rangle|0\rangle_{\mathbf{Dec}}$$

From the above game, it follows that p' is negligibly close to $\frac{1}{2}$. Moreover, if we suitably instantiate $\mathcal{A}$ (using the verifier) and $|\Phi\rangle$, it follows that $|\Phi_{\mathsf{yes}}\rangle = |\phi_0\rangle$ and $|\Phi_{\mathsf{no}}\rangle = |\phi_1\rangle$. Thus, we have $p(\Phi_{i,\mathsf{slot}})$ to be negligibly close to $\frac{1}{2}$.

□

Using above, we write $U_i^{V^*}|\Psi_{0,w}^{i-1}\rangle$ as follows:

$$U_i^{V^*}|\Psi_{0,w}^{i-1}\rangle \quad = \quad \sqrt{p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Bad}}\rangle|1\rangle_{\mathbf{Dec}},$$

where $|\Psi_{i,\mathsf{Good}}\rangle$ is a superposition over transcripts such that either one of the following two conditions are satisfied: (i) the slot chosen in the i^{th} block is matched or, (ii) the verifier aborts and the simulator decides to not rewind. Similarly, we can define $|\Psi_{i,\mathsf{Bad}}\rangle$. Define $p_1 = \frac{1}{2}$ and $p_0 = 0.49$. We note that the following holds:

- $|p(\Phi_{i,\mathsf{slot}}) - p_1| \leq \varepsilon$, where $\varepsilon = \nu(\lambda)$, for some negligible function $\nu(\cdot)$ and,
- $p_0(1-p_0) \leq p_1(1-p_1)$ and,
- $p_0 \leq p(\Phi_{i,\mathsf{slot}})$.

Thus, from the Watrous rewinding lemma (Lemma 8), $\mathsf{Amplifier}\left(U_i^{V^*}\right)$ outputs a circuit W_i, of polynomial size, such that W_i on input the state $|\Psi_{0,w}^{i-1}\rangle$, outputs a state $|\Psi_{0,w}^{i}\rangle$ that is exponentially (in λ) close in trace distance to the state $|\Psi_{i,\mathsf{Good}}\rangle$. This means that, in hybrid $\mathsf{Hyb}_{2.i+1}$, the state obtained after the execution of block B_i is exponentially close in trace distance to the state $|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}}$.

Indistinguishability of $\mathcal{D}_1$ and $\mathcal{D}_2$. We just argued above that the intermediate state obtained in $\mathsf{Hyb}_{2.i}$ is $|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}}$. On the other hand, the intermediate state obtained in $\mathsf{Hyb}_{2.i-1}$ is $|\Psi_{0,w}^{i-1}\rangle$ is $U_i^{V^*}|\Psi_{0,w}^{i-1}\rangle = \sqrt{p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Bad}}\rangle|1\rangle_{\mathbf{Dec}}$. We need to argue that the distribution of measurements of the registers $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}$ along with the residual state $\mathbf{Aux}$ register in both the cases are computationally indistinguishable.

Note that for any ρ_0, ρ_1 such that $\rho_0 \approx_c \rho_1$[13], then for any $p \geq 0$ we have that $\rho_0 = p \cdot \rho_0 + (1-p)\rho \approx_c p \cdot \rho_0 + (1-p) \cdot \rho_1$. In our case we have, ρ_0 is the post-measurement state on the registers $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t \leq i}, \mathbf{Aux}$ after measuring the $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t \leq i}$ registers of the state $|\Psi_{\mathsf{Good}}\rangle$. Similarly, we define ρ_1 with respect to $|\Psi_{\mathsf{Bad}}\rangle$. In $\mathsf{Hyb}_{2.i-1}$, the intermediate state is ρ_0 and in $\mathsf{Hyb}_{2.i}$, the intermediate state is ρ_1.

Thus, it suffices to show that with probability neglibly close to 1, the post-measurement states ρ_0 and ρ_1 are computationally indistinguishable. This follows from the quantum-concealing property of commitment schemes and is similar to the proof of Claim 29; if the verifier can distinguish a matched slot versus an unmatched slot then this verifier is violating the quantum-concealing property of the commitment scheme.

This proves that hybrids $\mathsf{Hyb}_{2.i}$ and $\mathsf{Hyb}_{2.i+1}$ are computationally indistinguishable.

4.2.2 Proof of Claim 28

Before we prove Claim 28, we first give an auxiliary definition and some claims.

Auxiliary Definition and Claims.

Definition 30 (Partitioning)**.** *We define a partitioning of a protocol transcript (consisting of messages from all the sessions) $\mathcal{S}$ to be $\{B_1, \ldots, B_L\}$ associated with parameter ℓ_B as follows: B_1 consists of the first ℓ_B messages of S, B_2 consists of the second ℓ_B messages of S and so on. If $|S| - \ell_B \cdot (L-1) < \ell_B$ then the last block B_L will just contain the remaining $|S| - \ell_B \cdot (L-1)$ messages.*

The following claim lower bounds the number of blocks that will contain a full slot for any given verifier. In particular, with our chosen parameters, we can show that the number of such blocks is at least $6Q^5\lambda$. This will turn out to be enough number of blocks for the simulator to be able to obtain more than $60Q^7\lambda + Q^4\lambda$ matched commitments, with probability neglibly close to 1, for every verifier before starting Stage 2.

Claim 31. *For any transcript $\mathcal{S}$ of Q verifiers $V_1, \ldots, V_Q$ with partitioning $\{B_1, \ldots, B_L\}$, for every verifier V_i, we have $\mathsf{N}_i \geq 6Q^5\lambda$; that is, there are at least $6Q^5\lambda$ number of blocks containing at least one slot of V_i.*

Proof. Fix a verifier V_i. Note that the number of blocks containing at least 4 messages of V_i lower bounds N_i. Denote μ_i be the number of blocks containing at least 4 messages of V_i.

Let $b_1, \ldots, b_{\mu_i}$ be the number of messages of V_i in each of these μ_i blocks. Let the number of messages in the remaining $L - \mu_i$ blocks be denoted by $a_1, \ldots, a_{L-\mu_i}$.

The following holds: $\sum_{i=1}^{\mu_i} b_i + \sum_{i=1}^{L-\mu_i} a_i = \frac{2(\ell_{\mathsf{prot}}-1)}{3}$. Since $\sum_{i=1}^{\mu_i} b_i \leq \ell_B \mu_i$, $\sum_{i=1}^{L-\mu_i} a_i \leq 3(L-\mu_i)$ and $\ell_B = \frac{\ell_{\mathsf{prot}} \cdot Q}{L}$, we have:

$$\mu_i \ell_B + 3(L - \mu_i) \geq \frac{2(\ell_{\mathsf{prot}} - 1)}{3} \geq \frac{\ell_{\mathsf{prot}}}{2}$$

[13] By $\rho_0 \approx_c \rho_1$, we mean that the state sampled according to ρ_0 is computationally indistinguishable from the state sampled according to ρ_1.

From this, we can determine μ_i to be at least $\frac{\frac{\ell_{\mathsf{prot}}}{2}-3L}{\ell_B-3}$. We can now lower bound the number of blocks containing at least 4 messages as follows.

$$\begin{aligned}
\mathsf{N}_i \geq \mu_i &\geq \left(\frac{\frac{\ell_{\mathsf{prot}}}{2}-3L}{\frac{\ell_{\mathsf{prot}}Q}{L}-3}\right) \\
&\geq \frac{\frac{\ell_{\mathsf{prot}}}{2}-3L}{\ell_{\mathsf{prot}}} \cdot \frac{L}{Q} \\
&\geq \left[1-\frac{6L}{\ell_{\mathsf{prot}}}\right]\frac{L}{2Q} \\
&\geq \left[1-\frac{6L}{3\ell_{\mathsf{slot}}}\right]\frac{L}{2Q} \\
&\geq \left[1-\frac{2}{5Q}\right]\frac{L}{2Q} \\
&\geq \left(1-\frac{1}{2}\right)12\lambda Q^5 \qquad (\because L = 24Q^6\lambda,\ \ell_{\mathsf{slot}} = 120Q^7\lambda) \\
&\geq 6\lambda Q^5
\end{aligned}$$

□

The following claim lower bounds the expected number of slots that will be *rigged* by the simulator (i.e., these are the slots the simulator matches by rewinding) for any given verifier before starting Stage 2. Specifically, it bounds the number of slots that it will be able to match thanks to block rewinding.

Claim 32 (Matching by Rigging)**.** *Let $\mathcal{S}$ be a scheduling of Q verifiers $V_1, \ldots, V_Q$. Let $\{B_1, \ldots, B_L\}$ be the partitioning associated with $\mathcal{S}$.*

Consider the following process: for $i = 1, \ldots, L$,

- *Let T_i be such that all the verifiers $\{V_j\}_{j \in T_i}$ have a slot in B_i.*
- *Pick $j^* \xleftarrow{\$} T$.*
- *Finally, pick a slot of V_{j^*} in block B_i uniformly at random.*

Let $\mathsf{X}_{i,j}$ be a random variable defined to be 1 if in the j^{th} block, a slot of V_i is picked. Then, for any $i \in [Q]$, $\mathbb{E}[\sum_{j\in[L]} \mathsf{X}_{i,j}] \geq 6\lambda Q^4$. Furthermore, we have that

$$\mathsf{Pr}\left[\exists i \in [Q], \sum_{j\in[L]} \mathsf{X}_{i,j} \leq 3\lambda Q^4\right] \leq \mathsf{negl}(\lambda)$$

.

Proof. Let $b_{i,j}$ be such that $b_{i,j} = 1$ if the i^{th} verifier has a slot in the j^{th} block, else its set to 0. Then, we have $\mathbb{E}[\sum_{j\in[L]} \mathsf{X}_{i,j}] \geq \sum_{j\in[L]} b_{i,j} \cdot \frac{1}{Q}$. Note that $|\{j : b_{i,j} \neq 0\}| = \mathsf{N}_i$. Thus, we have

$\mathbb{E}[\sum_{j\in[L]} \mathsf{X}_{i,j}] \geq \frac{1}{Q} \cdot \mathsf{N}_i$. Further applying Claim 31, we have $\mathbb{E}[\sum_{j\in[L]} \mathsf{X}_{i,j}] \geq 6\lambda Q^4$. To finish the proof of the claim, first notice that by Chernoff bound, we have that for any $i \in [Q]$,

$$\mathsf{Pr}\left[\sum_{j\in[L]} \mathsf{X}_{i,j} \leq 3\lambda Q^4\right] \leq e^{-\frac{3}{4}Q^4\lambda}.$$

By the union bound, we obtain that

$$\mathsf{Pr}\left[\exists i \in [Q], \sum_{j\in[L]} \mathsf{X}_{i,j} \leq 3\lambda Q^4\right] \leq Qe^{-\frac{3}{4}Q^4\lambda}$$

□

While the above claim provides a lower bound on the number of rigged slots, the following claim lower bounds the number of slots matched by luck. Combining the above and the below claim, it follows that with overwhelming probability, the number of matchd slots is at least $60Q^7\lambda + Q^4\lambda$.

Claim 33 (Matching by Luck)**.** *Let $\mathcal{S}$ be a transcript of the Q verifiers $V_1, \ldots, V_Q$. For any $i \in [Q]$ let $Z_{i,1}, \ldots Z_{i,120Q^7\lambda}$ be binary random variables such that $Z_{i,j} = 1$ iff $\mathsf{Comm}(b'_j; r_j) = \mathbf{c}_j$ where b'_j is the j^{th} response of the i^{th} verifier to commitment $\mathbf{c}_j$ by the prover. Let $X_{i,j}$ be as defined in Claim 32. The following holds:*

$$\mathsf{Pr}\left[T_i \subseteq [L],\ (\forall j \in T_i, X_{i,j} = 1) \bigwedge \left(\sum_{j\in[L]\setminus T_i} Z_{i,j} \geq 60Q^7\lambda - 2Q^4\lambda\right)\right] \geq 1 - \nu(\lambda),$$

for some negligible function $\nu(\cdot)$.

Proof. By the previous Claim, we have that with probability neglible close to 1, for all $i \in [Q]$, there exists T_i satisfying the desired properties, what is left is to show that

$$\sum_{j\in[L]\setminus T_i} Z_{i,j} \geq 60Q^7\lambda - 2Q^4\lambda$$

for all $i \in [Q]$.

For any $i \in [Q]$, we have that $\mathbb{E}[\sum_{j\in[L]\setminus T_i} Z_{i,j}] = 60Q^7\lambda - \frac{3}{2}Q^4\lambda$, and by Chernoff bound:

$$\begin{aligned}
\mathsf{Pr}\left[\sum_{j\in[L]\setminus T_i} Z_{i,j} \leq 60Q^7\lambda - 2Q^4\lambda\right] &= \mathsf{Pr}\left[\sum_{j\in[L]\setminus T_i} Z_{i,j} \leq \left(60Q^7\lambda - \frac{3}{2}Q^4\lambda\right) - \frac{1}{2}Q^4\lambda\right] \\
&\leq \exp\left(-\frac{(\frac{1}{2}Q^4\lambda)^2}{2(60Q^7\lambda - \frac{3}{2}Q^4\lambda)}\right) \\
&\leq \exp\left(-\frac{(\frac{1}{2}Q^4\lambda)^2}{2(60Q^7\lambda)}\right) \\
&= e^{-\frac{Q\lambda}{480}}.
\end{aligned}$$

Again, by union bound, we have that

$$\mathsf{Pr}\left[\exists i \in [Q], \sum_{j \in [L] \setminus T_i} Z_{i,j} \leq 60Q^7\lambda - 2Q^4\lambda\right] \leq Qe^{-\frac{Q\lambda}{480}}.$$

□

Combining these last two claims we conclude that the probability that there is a session V_i^* for which the simulator does not have more than $60Q^7\lambda + Q^4\lambda$ matched commitments is negligibly small in λ.

Finishing Proof of Claim 28. We use the auxiliary claims from the previous section to complete the proof. We prove this via the following hybrid argument.

$\underline{\mathsf{Hyb}_{3.i}^{(1)}}$: This is identical to the hybrid $\mathsf{Hyb}_{3.i}$.

$\underline{\mathsf{Hyb}_{3.i}^{(2)}}$: This is the same as the previous hybrid except that the simulator sets it responses, to the i^{th} session, as $\perp$ if the number of matched slots for the i^{th} session is $< 60Q^7\lambda + Q^4\lambda$.

From Claim 33, we have that the probability that this hybrid aborts is negligible in λ. Conditioned on this hybrid not aborting, the output distributions of $\mathsf{Hyb}_{3.i}^{(1)}$ and $\mathsf{Hyb}_{3.i}^{(2)}$ are identical.

$\underline{\mathsf{Hyb}_{3.i}^{(3)}}$: This is identical to the hybrid $\mathsf{Hyb}_{3.i+1}$.

To argue that $\mathsf{Hyb}_{3.i}^{(3)}$ and $\mathsf{Hyb}_{3.i}^{(2)}$ are computationally indistinguishable we will use the quantum witness indistinguishable property of Π_{WI}. Suppose that there is an adversary $\mathcal{A}$ that distinguishes the output distributions of these two hybrids. We define the following QPT $\mathcal{B}_i$ that breaks the security of Π_{WI}. That is, $\mathcal{B}_i$ is a QPT verifier, in the WI experiment, that can distinguish whether the prover used one witness versus another. $\mathcal{B}_i$ is given as auxiliary advice a transcript (and verifier's private state) of $\mathsf{Hyb}_{3.i}^{(2)}$ executed until the verifier's first message of the i^{th} WI execution in the transcript. In particular, conditioned on not aborting, this transcript has enough number of matching slots corresponding to the i^{th} execution (in the order of arrival of messages) of WI. Then, $\mathcal{B}_i$ interacts with the verifier V^* as in the protocol with P from then on, but forwards the verifier's messages corresponding to the i^{th} WI execution to the prover of WI. The output of $\mathcal{B}_i$ is the same as the output of the verifier V^*.

Firstly, from the security of WI, the output distribution of $\mathcal{B}_i$ when the prover uses w is computationally indistinguishable from the output distribution of $\mathcal{B}_i$ when the prover uses the other witness, i.e., decommitments of matched slots.

If the prover used the witness w, then the output distribution of $\mathcal{B}_i$ is computationally indistinguishable from the output of $\mathsf{Hyb}_{3.i}^{(2)}$. To see why, note that the only difference between $\mathcal{B}_i$ and $\mathsf{Hyb}_{3.i}^{(2)}$ is that in $\mathcal{B}_i$, all the blocks starting from the i^{th} WI are not rewound. But we already showed, assuming security of commitments, that V^* cannot distinguish the case when the block is being rewound versus the case when it is not.

Furthermore, similarly, when the prover is using the decommitments of matched slots, the output distribution of $\mathcal{B}_i$ is computationally indistinguishable from the output of $\mathsf{Hyb}_{3.i}^{(3)}$.

Thus, the output distributions of $\mathsf{Hyb}_{3.i}^{(2)}$ and $\mathsf{Hyb}_{3.i}^{(3)}$ are computationally indistinguishable.

5 Post-Quantum Statistical Receiver Oblivious Transfer

We begin by presenting the definition of statistical receiver oblivious transfer with post-quantum security. We consider a natural adaption of the definition of [GJJM20] (see also [DGH+20]), who originally defined in the classical setting.

The definition we provide is written this way to make it compatible with the 3-round OT definition from [GJJM20]. The main difference is that we allow for an interactive phase instead of the sender's first message in [GJJM20].

5.1 Definition

Definition 34 (Post-Quantum Statistical Receiver-Private Oblivious Transfer)**.** *An oblivious transfer protocol, Π_{OT}, is an interactive protocol between a PPT sender and a PPT receiver (S,R), and a triplet of algorithms $(\mathsf{OT}_2,\mathsf{OT}_3,\mathsf{OT}_4)$ such that*

Interactive Phase. S *and* P *interact for* $\mathrm{poly}(\lambda)$ *rounds. The receiver's input is* λ *and a bit* $\beta\in\{0,1\}$*. The sender's input is* λ*. Let* ot_1 *be the transcript generated in this round, and let* st_S *and* st_R *be the private state of the sender and receiver (respectively) at the end of the round.*

Receiver's Final Message. *The receiver* R *computes* $(\mathsf{ot}_2,\mathsf{st}'_R)\leftarrow\mathsf{OT}_2(1^\lambda,\mathsf{ot}_1,\beta,\mathsf{st}_R)$

Sender's Final Message. S *with input* $(m_0,m_1)\in\{0,1\}^2$ *computes* $(\mathsf{ot}_3,\mathsf{st}'_S)\leftarrow\mathsf{OT}_3(1^\lambda,\mathsf{ot}_2,\mathsf{st}_S,m_0,m_1)$*, and it sends* ot_3 *to* R*.*

Reconstruction. *The receiver computes* $m'\leftarrow\mathsf{OT}_4(1^\lambda,\mathsf{ot}_3,\mathsf{st}'_R)$*. Output* m'*.*

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

$$\mathsf{Pr}\left[\begin{array}{c}(\mathsf{ot}_1,\mathsf{st}_S,\mathsf{st}_R)\leftarrow\langle\mathsf{S}(1^\lambda),\mathsf{R}(1^\lambda,\beta)\rangle\\(\mathsf{ot}_2,\mathsf{st}'_R)\leftarrow\mathsf{OT}_2(1^\lambda,\mathsf{ot}_1,\beta,\mathsf{st}_R)\\(\mathsf{ot}_3,\mathsf{st}'_S)\leftarrow\mathsf{OT}_3(1^\lambda,\mathsf{ot}_2,\mathsf{st}_S,m_0,m_1)\\m'\leftarrow\mathsf{OT}_4(1^\lambda,\mathsf{ot}_3,\mathsf{st}'_R)\end{array}\;:\;m'=m_\beta\right]=1$$

Statistical Receiver-Privacy. *For any sender* S^**, denote* $(\mathsf{ot}_1^{(0)},\mathsf{st}_R^{(0)})\leftarrow\langle S^*(1^\lambda),R(1^\lambda,0)\rangle$ *and* $(\mathsf{ot}_1^{(1)},\mathsf{st}_R^{(1)})\leftarrow\langle S^*(1^\lambda),R(1^\lambda,1)\rangle$*. Furthermore, let* $(\mathsf{ot}_2^{(0)},(\mathsf{st}_R^{(0)})')\leftarrow\mathsf{OT}_2(1^\lambda,\mathsf{ot}_1^{(0)},0,\mathsf{st}_R^{(0)})$ *and let* $(\mathsf{ot}_2^{(1)},(\mathsf{st}_R^{(1)})')\leftarrow\mathsf{OT}_2(1^\lambda,\mathsf{ot}_1^{(1)},1,\mathsf{st}_R^{(1)})$*.*

Then the statistical distance between the marginal distributons $\{(\mathsf{ot}_1^{(0)},\mathsf{ot}_2^{(0)})\}$ *and* $\{(\mathsf{ot}_1^{(1)},\mathsf{ot}_2^{(1)})\}$ *is a negligible function in* λ*.*

Post-Quantum Sender-Privacy. *For any non-uniform QPT distinguisher* $\mathcal{A}$ *and any malicious receiver* R^**, which receives as input state that is possibly entangled with the input state of* $\mathcal{A}$*, we define the following games.*

Interact with R^*. *The challenger plays the role of an honest sender in the interactive phase with* R^**. Then the receiver* R^* *outputs a state in a register* $\mathbf{B}$ *and a message* z*. The message* z *is sent to the challenger. The register* $\mathbf{B}$ *is given to* $\mathcal{A}$*.*

Game $G_0(m_0, m_1)$: *The challenger samples* $b_0 \leftarrow \{0,1\}$ *at random and computes* $ot_3 \leftarrow \mathsf{OT}_3(1^\lambda, z, \mathsf{st}_S, m_{b_0}, m_1)$. *Then,* ot_3 *is sent to* $\mathcal{A}$. *Finally,* $\mathcal{A}$ *outputs two bits* b'_0 *and* b'_1. *If* $b_0 = b'_0$ *then we say that* $\mathcal{A}$ *wins the game* G_0.

Game $G_1(m_0, m_1)$: *The challenger samples* $b_1 \xleftarrow{\$} \{0,1\}$ *at random, and then computes* $ot_3 \leftarrow \mathsf{OT}_3(1^\lambda, z, \mathsf{st}_S, m_0, m_{b_1})$. *Then,* ot_3 *is sent to* $\mathcal{A}$. *Finally,* $\mathcal{A}$ *outputs two bits* b'_0 *and* b'_1. *If* $b_1 = b'_1$ *then we say that* $\mathcal{A}$ *wins the game* G_1.

We define the advantage as follows:

$$\mathsf{Adv}(\mathcal{A}, \mathsf{R}^*, m_0, m_1) = \mathbb{E}_{\mathsf{View}_{R^*}} \left[\min\{p_0, p_1\}\right],$$

where:

- $p_0 = \left|\mathsf{Pr}[\mathcal{A} \text{ wins } \mathsf{G}_0(m_0, m_1)] - \frac{1}{2}\right|$
- $p_1 = \left|\mathsf{Pr}[\mathcal{A} \text{ wins } \mathsf{G}_1(m_0, m_1)] - \frac{1}{2}\right|$

We say that the oblivious transfer scheme is computational sender-secure if for every $m_0, m_1 \in \{0,1\}$, *we have* $\mathsf{Adv}(\mathcal{A}, \mathsf{R}^*, m_0, m_1)$ *to be negligible in* λ.

Remark 35. *The definition of advantage we state above is weaker than what is stated in [GJJM20]. In particular, from the max-min inequality, the advantage in our definition is upper bounded by the advantage in their definition. This means that any protocol which has negligible advantage according to their notion will also have negligible advantage according to our definition.*

5.2 Main Tools

To construct a statistical receiver-private oblivious transfer, we use two tools: (i) a statistical zero-knowledge argument system and, (ii) statistical sender-private oblivious transfer.

5.2.1 Statistical Zero-Knowledge Quantum Argument System

Definition 36 (Statistical ZK Quantum Argument System)**.** *Let* Π *be an interactive protocol between a classical PPT prover* P *and a classical PPT verifier* V. *Let* $\mathcal{R}(\mathcal{L})$ *be the NP relation associated with* Π.

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

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

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

for some negligible function negl.

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

- **(Quantum Computational) Soundness**: *For every QPT adversary* P^*, *every* $x \notin \mathcal{R}(\mathcal{L})$, *every* $\mathrm{poly}(\lambda)$*-qubit advice* ρ,

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

for some negligible function negl.

Π *is said to satisfy* **statistical zero-knowledge** *if the following holds:*

- **Statistical Zero-Knowledge***:For every sufficiently large $\lambda \in \mathbb{N}$, every computationally unbounded adversary V^*, there exists a QPT simulator* Sim *such that for every $(x, w) \in \mathcal{R}(\mathcal{L})$, the state output by V^* is close in trace distance to the state output by* Sim*.*

To construct a statistical ZK quantum argument system, we will use a non-interactive (statistical) ZK protocol for NP (NIZK). We define NIZK below.

Definition 37 (Non-interactive statistical ZK argument system)**.** *A NIZK argument system, Π_{nizk}, for an NP relation $\mathcal{R}(\mathcal{L})$ in the common random string model is a triplet of PPT algorithms:*

- $\mathsf{Gen}(1^\lambda)$*: outputs a public random string* CRS*. The output distribution of* CRS *is generated according to the uniform distribution.*
- $P(\mathsf{CRS}, x, w)$*: On instance $(x; w) \in \mathcal{R}(\mathcal{L})$ outputs a proof π.*
- $V(\mathsf{CRS}, x, \pi)$*: Outputs 1 if it accepts the proof, and 0 if it rejects.*

Π_{nizk} *is said to satisfy* ***completeness*** *if the following holds:*

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

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

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

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

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

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

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

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

5.2.2 Construction

In order to construct a statistical ZK quantum argument system for an NP relation $\mathcal{R}(\mathcal{L})$, we use the following ingredients:

- A quantum zero-knowledge protocol Π_{zk} for the NP relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$. We described the relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$, parametrized by security parameter λ, described below:

$$\mathcal{R}(\mathcal{L}_{\mathsf{zk}}) = \{((crs, \mathbf{c}, b); (a, \mathbf{r})) : \overset{crs=a\oplus b}{\underset{\mathbf{c}=\mathsf{Comm}(1^\lambda,a;\mathbf{r})}{\bigwedge}}\}$$

- A perfectly binding and quantum computationally hiding commitment scheme, Comm, where the length of randomness is λ.
- A non-interactive statistical zero-knowledge argument system Π_{nizk} for $\mathcal{R}(\mathcal{L})$, where the length of the CRS is $q(\lambda)$.

We present a construction in Figure 2.

Instance: x.
Witness: w.

- V samples $\mathbf{a} \xleftarrow{\$} \{0,1\}^\lambda$.
- For every $i \in [q(\lambda)]$, P and V perform the following operations:
  - $V \to P$: V computes $\mathbf{c}_i \leftarrow \mathsf{Comm}(1^\lambda, a_i; \mathbf{r}_i)$, where $\mathbf{r}_i \xleftarrow{\$} \{0,1\}^\lambda$ and a_i is the i^{th} bit of $\mathbf{a}$. V sends $\mathbf{c}_i$ to P.
  - $P \to V$: P sends b_i to V, where $b_i \xleftarrow{\$} \{0,1\}$.
  - $V \to P$: V sends $crs_i = a_i \oplus b_i$ to P.
  - $V \leftrightarrow P$: P and V will execute Π_{zk}, with P playing the role of verifier in Π_{zk} and V plays the role of the prover. The instance is $(crs_i, \mathbf{c}_i, b_i)$ and the witness is $(a_i, \mathbf{r}_i)$.
- Set $\mathsf{CRS} = \left(crs_1, \ldots, crs_{q(\lambda)}\right)$.
- $P \to V$: P computes a proof π on input common random string CRS, instance x and witness w using Π_{nizk}. It sends π to V.
- V computes the verifier of Π_{nizk} on input (CRS, x, π). It outputs the decision bit of the verifier of Π_{nizk}.

Figure 2: Statistical ZK Quantum Argument System

Completeness. The completeness follows from the completeness of Π_{zk} and Π_{nizk}.

Quantum Computational Soundness. Let $x \notin \mathcal{L}$. Suppose P^* be a QPT prover that on input (x, ρ), for some poly(λ)-qubit advice ρ, convinces the verifier V^* to accept x with probability ε. We prove that ε is negligible using a hybrid argument.

$\underline{\mathsf{Hyb}_1}$: This corresponds to the execution of P^* and V. The probability that V accepts is ε.

$\underline{\mathsf{Hyb}_{2.i} \text{ for } i \in [q(\lambda)]}$: We consider a hybrid verifier $\mathsf{Hyb}_{2.i}.V$ that executes the simulator Sim in the i^{th} execution of Π_{zk}, instead of running the real prover. Except this change, the hybrid verifier $\mathsf{Hyb}_{2.i}.V$ behaves the same as $\mathsf{Hyb}_{2.i-1}.V$ if $i > 1$ or as V if $i = 1$.

From the (computational) quantum zero-knowledge property of Π_{zk}, the probability that $\mathsf{Hyb}_{2.i}.V$ accepts is negligibly close to ε.

$\underline{\mathsf{Hyb}_{3.i}, \text{ for } i \in [q(\lambda)]}$: We consider a hybrid verifier $\mathsf{Hyb}_{3.i}.V$ that computes the i^{th} commitment $\mathbf{c}_i$ as follows: $\mathbf{c}_i \leftarrow \mathsf{Comm}(1^\lambda, 0)$. Except this change, the hybrid verifier $\mathsf{Hyb}_{3.i}.V$ behaves the same as $\mathsf{Hyb}_{3.i-1}.V$ if $i > 1$ or as $\mathsf{Hyb}_{2.q(\lambda)}.V$ if $i = 1$.

From the quantum-concealing property of Comm, the probability that $\mathsf{Hyb}_{3.i}.V$ accepts is negligibly close to ε.

$\underline{\mathsf{Hyb}_4}$: We consider a hybrid verifier $\mathsf{Hyb}_4.V$, which is essentially the same as $\mathsf{Hyb}_{3.q(\lambda)}.V$, except that it generates $\mathsf{CRS} \xleftarrow{\$} \{0,1\}^{q(\lambda)}$ and sends CRS to P.

Since the hybrids $\mathsf{Hyb}_{3.q(\lambda)}.V$ and $\mathsf{Hyb}_4.V$ are identical, the probability that $\mathsf{Hyb}_4.V$ accepts is negligibly close to ε.

From the computational soundness of Π_{nizk}, the probability that $\mathsf{Hyb}_4.V$ accepts is negligible. Thus, ε is negligible.

Statistical Zero-Knowledge. Let V^* be a computationally unbounded verifier and let $|\Psi\rangle$ be the initial state of V^*. Before we describe the simulator we first define the following registers. For $i = 1, \ldots, q(\lambda)$:

- $\mathbf{B}_i$: it contains the bit sent by the simulator in the i^{th} iteration.
- $\mathbf{R}_i$: it contains the receiver's commitment and the i^{th} bit of CRS sent during the i^{th} iteration.
- $\mathbf{IZ}_i$: it contains the messages of zero-knowledge exchanged during the i^{th} iteration.
- **Dec**: it contains the decision bit.
- **Aux**: it contains the auxiliary state of the verifier.
- **NZ**: it contains the final NIZK proof sent by the simulator.
- **C**: it contains the common reference string.
- **X**: this is a poly(λ)-qubit ancillary register.

Description of Simulator:

- The simulator prepares the following state:

$$|\Psi_1\rangle = \left(\bigotimes_{i=1}^{q(\lambda)} |0\rangle_{\mathbf{R}_i}|0\rangle_{\mathbf{B}_i}|0\rangle_{\mathbf{IZ}_i}\right) \otimes |0\rangle_{\mathbf{NZ}}|0\rangle_{\mathbf{X}}|0\rangle_{\mathbf{C}}|\Psi\rangle_{\mathbf{Aux}}|0\rangle_{\mathsf{Dec}}$$

- It runs the NIZK simulator, $(\widehat{\mathsf{CRS}}, \widehat{\pi}) \leftarrow \Pi_{\mathsf{nizk}}.\mathsf{Sim}(1^\lambda, x)$, to compute $\widehat{\mathsf{CRS}}$. It stores $\widehat{CRS}$ in the register $\mathbf{C}$, and it stores $\widehat{\pi}$ in $\mathbf{NZ}$.

- For all $i = 1, \ldots, q(\lambda)$, let U_i be the unitary that performs the following operations in superposition.

  - It first applies V^* on the registers $\{\mathbf{B}_j, \mathbf{R}_j, \mathbf{IZ}_j, \mathbf{Aux}\}_{j<i}, \mathbf{R}_i$.
  - It then maps $|0\rangle_{\mathbf{B}_i}$ to $|+\rangle_{\mathbf{B}_i}$.
  - It then applies V^* on the registers $\{\mathbf{B}_j, \mathbf{R}_j, \mathbf{IZ}_j, \mathbf{Aux}\}_{j<i}, \mathbf{R}_i, \mathbf{B}_i$.
  - It then performs the i^{th} iteration of Π_{zk} with V^* in superposition. The transcript is stored in $\mathbf{IZ}_i$.
  - It then applies the following unitary $\widehat{U}$:

$$\widehat{U}|b_i\rangle_{\mathbf{B}_i}|\mathbf{c}_i\ crs_i\rangle_{\mathbf{R}_i}|\tau_i\rangle_{\mathbf{IZ}_i}|\widehat{\mathsf{CRS}}\rangle_{\mathbf{C}}|0\rangle_{\mathbf{Dec}} = \begin{cases} |b_i\rangle_{\mathbf{B}_i}|\mathbf{c}_i\ crs_i\rangle_{\mathbf{R}_i}|\tau_i\rangle_{\mathbf{IZ}_i}|\widehat{\mathsf{CRS}}\rangle_{\mathbf{C}}|1 \oplus \theta_i\rangle_{\mathbf{Dec}}, & \text{if } \tau_i \text{ is valid,} \\ |b_i\rangle_{\mathbf{B}_i}|\mathbf{c}_i\ crs_i\rangle_{\mathbf{R}_i}|\tau_i\rangle_{\mathbf{IZ}_i}|\widehat{\mathsf{CRS}}\rangle_{\mathbf{C}}|+\rangle_{\mathbf{Dec}}, & \text{othewise} \end{cases}$$

  We define $\theta_i = 1$ if the i^{th} bit of $\widehat{\mathsf{CRS}}$ is the same as crs_i, where crs_i is the i^{th} bit of CRS computed by V^* in the register $\mathbf{R}_i$. Furthermore, we define τ_i to be valid if the verifier in the i^{th} execution of Π_{zk} accepts.

- Let $W_i = \mathsf{Amplifier}(U_i)$; where $\mathsf{Amplifier}$ is the circuit guaranteed by Lemma 8. Simulator computes $|\Psi_i\rangle = W_i|\Psi_{i-1}\rangle$.

- Finally, it uses $\widehat{\pi}$ stored in $\mathbf{NZ}$ as the proof for the NIZK step.

- Measure all the registers except for $\mathbf{Aux}$.

We now prove the statistical indistinguishability of the real and the ideal world using a hybrid argument. Consider the following hybrids.

$\underline{\mathsf{Hyb}_1}$: This corresponds to the real execution between P and V^*.

$\underline{\mathsf{Hyb}_{2.i} \text{ for } i \in [q(\lambda)]}$: We define a hybrid prover as follows: sample $\mathsf{CRS} \leftarrow \mathsf{Gen}(1^\lambda)$. Note that CRS is generated according to the uniform distribution. Prepare the state $|\Psi_1\rangle$ as given in the description of the simulator. Apply $W_i \cdots W_1|\Psi_1\rangle$ to obtain $|\Psi_i\rangle$. That is, perform Watrous rewinding for the first i iterations of the OT protocol, similarly to the way the simulator does, but using CRS, instead of $\widehat{\mathsf{CRS}}$. Then, the hybrid prover uses the real prover to interact with V^*, that receives as input $|\Psi_i\rangle$, to perform the operations for the rest of the protocol.

We now show that the output distributions of the hybrids $\mathsf{Hyb}_{2.i-1}$ and $\mathsf{Hyb}_{2.i}$ are computationally indistinguishable. In order to show this, we use a similar argument that was used in the proof of Claim 27. It suffices to argue that the following distributions are statistically close:

- $\mathcal{D}_1$: Measure the registers $\{\mathbf{R}_i, \mathbf{IZ}_i\}_{i\in[q(\lambda)]}, \mathbf{NZ}$ after the i^{th} iteration in $\mathsf{Hyb}_{2.i-1}$ and output the measurement outcome along with the residual state in **Aux**.
- $\mathcal{D}_2$: Measure the registers $\{\mathbf{R}_i, \mathbf{IZ}_i\}_{i\in[q(\lambda)]}, \mathbf{NZ}$ after the i^{th} iteration in $\mathsf{Hyb}_{2.i}$ and output the measurement outcome along with the residual state in **Aux**.

We prove this in two steps: first, we apply Watrous rewinding and analyze the state obtained after the i^{th} iteration in $\mathsf{Hyb}_{2.i}$. In the next step, we use this to argue the indistinguishability of $\mathcal{D}_1$ and $\mathcal{D}_2$.

Applying Watrous Rewinding. Suppose $|\Psi_{i-1}\rangle = W_{i-1}\cdots W_1|\Psi_1\rangle$. Consider the following:

$$\begin{aligned} U_i|\Psi_{i-1}\rangle &= \sqrt{q}\left(\sqrt{p}|\phi_{\mathsf{good}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p}|\phi_{\mathsf{bad}}\rangle|1\rangle_{\mathbf{Dec}}\right) + \sqrt{1-q}|\phi_{\mathsf{invalid}}\rangle|+\rangle_{\mathbf{Dec}}, \\ &= \sqrt{p}\left(\sqrt{q}|\phi_{\mathsf{good}}\rangle + \sqrt{1-q}|\phi_{\mathsf{invalid}}\rangle\right)|0\rangle_{\mathbf{Dec}} + \sqrt{1-p}\left(\sqrt{q}|\phi_{\mathsf{bad}}\rangle + \sqrt{1-q}|\phi_{\mathsf{invalid}}\rangle\right)|1\rangle_{\mathbf{Dec}}, \end{aligned}$$

where:

- q is the probability with which V^* convinces P in the i^{th} execution of Π_{zk},
- $\left|p - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$: this follows from a similar argument as in the proof of Claim 29,
- $|\phi_{\mathsf{invalid}}\rangle$ (defined on all the registers except the **Dec** register) is a superposition over the messages containing the i^{th} iteration Π_{zk} transcripts that are not accepted by the verifier of Π_{zk},
- $|\phi_{\mathsf{good}}\rangle$ (defined on all the registers except the **Dec** register) is a superposition over the messages of the i^{th} iteration containing $crs_i = \mathsf{CRS}_i$ and,
- $|\phi_{\mathsf{bad}}\rangle$ (defined on all the registers except the **Dec** register) is a superposition over the messages of the i^{th} iteration containing $crs_i \neq \mathsf{CRS}_i$.

Once we apply Theorem 8, the resulting state will be $W_i|\Psi_{i-1}\rangle = \left(\sqrt{q}|\phi_{\mathsf{good}}\rangle + \sqrt{1-q}|\phi_{\mathsf{invalid}}\rangle\right)|0\rangle_{\mathbf{Dec}}$ with probability negligibly close to 1.

Indistinguishability of $\mathcal{D}_1$ and $\mathcal{D}_2$. As in the proof of Claim 27, it suffices to argue that the distribution of measurements of $\{\mathbf{R}_i, \mathbf{IZ}_i\}_{i\in[q(\lambda)]}, \mathbf{NZ}$ in $|\phi_{\mathsf{good}}\rangle$ along with the residual state in **Aux** is computationally indistinguishable from the distribution of measurements of $\{\mathbf{R}_i, \mathbf{IZ}_i\}_{i\in[q(\lambda)]}, \mathbf{NZ}$ in $|\phi_{\mathsf{bad}}\rangle$ along with the residual state in **Aux**. This follows from the perfect binding property of Comm and the statistical soundness property of Π_{zk} using a similar argument used in Claim 29: if the verifier is not computed in superposition then the verifier cannot distinguish whether $crs_i = \mathsf{CRS}_i$ or whether $crs_i \neq \mathsf{CRS}_i$. Moreover, this is true even if the verifier is computed in superposition and measuring the transcript registers in the end.

Hyb_3: Execute the simulator on input the state $|\Psi\rangle$.

From the statistical zero-knowledge property of Π_{nizk}, it follows that the state output by V^* in the hybrid $\mathsf{Hyb}_{2.q(\lambda)}$ is close in trace distance to the state output by V^* in the hybrid Hyb_3.

5.2.3 Post-Quantum Statistical Sender-Private OT

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

Definition 38 (Post-Quantum Statistical Sender-Private OT)**.** *A two-round oblivious transfer is a tuple of algorithms* $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ *which specifies the following protocol.*

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

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

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

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

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

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

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

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

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

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

5.3 Construction

- A 2-round post-quantum statistical sender-private OT, $\Pi_{OT} = (OT_1, OT_2, OT_3)$. Without loss of generality, we assume that the length of the randomness is λ.

  We say that a transcript τ, consisting of messages $(\mathsf{msg}_1, \mathsf{msg}_2)$, is valid with respect to sender's randomness r and its input bits (m_0, m_1) if the following holds: $(\mathsf{msg}_2, \mathsf{st}) \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{msg}_1, m_0, m_1; r)$.

- A statistical zero-knowledge quantum argument system, Π_{zk}, for the NP relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$. We described the relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$, parametrized by security parameter λ, described below:

$$\mathcal{R}\left(\mathcal{L}_{\mathsf{zk}}\right) = \left\{ \left(\left(\tau^*_{\mathsf{OT}}, \{\tau^{(i,j)}_{\mathsf{OT}}, b_{i,j}\}_{i\in[\lambda+2], j\in[\lambda]}\right); \left(r', \beta, r^*_{\mathsf{OT}}, \left\{r^{(i,j)}_{\mathsf{OT}}, sh_{i,j}, \alpha_{i,j}\right\}_{i\in[\lambda+2], j\in[\lambda]}\right)\right) \;:\right.$$

$$\left\{\left(\begin{array}{c}\forall i\in[\lambda+2],j\in[\lambda],\\ \tau_{\mathsf{OT}}^{(i,j)} \text{ is valid w.r.t}\\ r_{\mathsf{OT}}^{(i,j)} \text{ and } (((1-b_{i,j})sh_{i,j}+b_{i,j}\cdot\alpha_{i,j}),\ (b_{i,j}sh_{i,j}+(1-b_{i,j})\cdot\alpha_{i,j}))\\ \bigwedge\\ \tau_{\mathsf{OT}}^{*} \text{ is valid w.r.t } r_{\mathsf{OT}}^{*} \text{ and } (r',r'\oplus\beta)\end{array}\right)\bigwedge\left(\begin{array}{c}\forall i\in[\lambda+2],\\ \oplus_{j=1}^{\lambda}sh_{i,j}=w_i\end{array}\right)\bigwedge w=(r',\beta,r_{\mathsf{OT}}^{*})\right\}$$

We show that the construction in Figure 3 is a post-quantum statistical receiver oblivious transfer protocol.

Correctness. The correctness of our protocol follows from the correctness of Π_{OT} and the completeness of Π_{zk}.

Statistical Receiver Privacy. Let S^* be a computationally unbounded sender. Instead of proving that the sender cannot distinguish receiver's bit to be 0 versus receiver's bit to be 1 with non-negligible probability, we instead prove the following: suppose receiver chooses its bit uniformly at random then the probability that the malicious sender can output β with probability negligibly close to $\frac{1}{2}$. We prove this via a hybrid argument. In the first hybrid, the receiver behaves honestly and uses the receiver's bit to be β, where β is chosen uniformly at random. We define a sequence of hybrids and show computational indistinguishability of every pair of consecutive hybrids. In the final hybrid, the receiver's bit will be information-theoretically hidden in the messages exchanged with S^*, which will prove the statistical receiver privacy property.

$\underline{\mathsf{Hyb}_1}$: In this hybrid, the receiver uses the bit β.

Let ε be the probability that S^* outputs β.

$\underline{\mathsf{Hyb}_2}$: Let $\mathsf{Sim}_{\mathsf{zk}}$ be the simulator associated with Π_{zk}. Instead of R playing the role of the prover in Π_{zk}, it executes $\mathsf{Sim}_{\mathsf{zk}}$.

From the statistical zero-knowledge property of Π_{zk}, the output distributions of S^* in the hybrids Hyb_1 and Hyb_2 are statistically close. The probability that S^* outputs β is negligibly close to ε in this hybrid.

$\underline{\mathsf{Hyb}_{3.(i,j)}}$, for $i \in [\lambda+2]$, $j \in [\lambda]$: In the $(i,j)^{th}$ execution of Π_{OT}, perform inefficient extraction to extract $b'_{i,j}$ from R'_{OT}. Recall that S^* plays the role of R'_{OT}. If $b'_{i,j} = b_{i,j}$ then set the input of the sender S'_{OT} to be $(sh_{i,j}, sh_{i,j})$ and if $b'_{i,j} \neq b_{i,j}$ then set the input of the sender S'_{OT} to be $(\alpha_{i,j}, \alpha_{i,j})$.

The statistical indistinguishability of this hybrid and the previous hybrid follows from the statistical sender-privacy property of Π_{OT}. The probability that S^* outputs β is negligibly close to ε in this hybrid.

$\underline{\mathsf{Hyb}_4}$: If there exists $i \in [\lambda+2]$, such that for every $j \in [\lambda]$, $b_{i,j} = b'_{i,j}$ then abort.

The probability that this hybrid aborts is at most $\frac{\lambda+2}{2^\lambda}$. Conditioned on this hybrid not aborting, this hybrid is identical to the previous one. The probability that S^* outputs β is negligibly close to ε in this hybrid.

$\underline{\mathsf{Hyb}_5}$: In the main execution of Π_{OT}, perform inefficient extraction to extract r from Π_{OT}. If $r = 0$, then set the input of the sender S'_{OT} to be (r', r') and if $r = 1$, then set the input of the sender to be $(r' \oplus \beta, r' \oplus \beta)$.

Input of sender S: (m_0, m_1).
Input of receiver R: β.

- R generates $r' \xleftarrow{\$} \{0,1\}$ uniformly at random. R samples $r^*_{OT} \xleftarrow{\$} \{0,1\}^\lambda$.
- Let $w = (r', \beta, r^*_{OT})$. For every $i \in [\lambda + 2]$, R generates shares $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random conditioned on $\oplus_{j=1}^{\lambda} sh_{i,j} = w_i$.
- For $i \in [\lambda + 2]$, R also generates bits $\alpha_{i,1}, \ldots, \alpha_{i,\lambda}$ uniformly at random.
- For $i \in [\lambda + 2], j \in [\lambda]$, do the following:
  - $S \leftrightarrow R$: S and R execute Π_{OT} with S playing the role of the receiver, denoted by R'_{OT}, in Π_{OT} and R playing the role of the sender, denoted by S'_{OT}, in Π_{OT}. The input of the receiver R'_{OT} in this protocol is 0, while the input of the sender S'_{OT} is set to be $(sh_{i,j}, \alpha_{i,j})$ if $b_{i,j} = 0$, otherwise it is set to be $(\alpha_{i,j}, sh_{i,j})$ if $b_{i,j} = 1$, where the bit $b_{i,j}$ is sampled uniformly at random by S'_{OT}. We call this execution as $(i,j)^{th}$ execution of Π_{OT}.
    Call the resulting transcript of the protocol to be $\tau^{(i,j)}_{OT}$ and let $r^{(i,j)}_{OT}$ be the randomness used by the sender S'_{OT} in Π_{OT}.
  - $R \to S$: R sends $b_{i,j}$ to S.
- S samples $r \xleftarrow{\$} \{0,1\}$.
- $S \leftrightarrow R$: S and R execute Π_{OT} with S playing the role of the receiver, denoted by R'_{OT}, in Π_{OT} and R playing the role of the sender, denoted by S'_{OT}, in Π_{OT}. The input of the receiver R'_{OT} is r and the input of the sender is $(r', r' \oplus \beta)$. Let $\widetilde{r}$ be the bit recovered by R'_{OT} at the end of the protocol.
  We call this execution the main execution of Π_{OT}. Call the resulting transcript of the protocol to be τ^*_{OT} and let r^*_{OT} be the randomness used by the sender S'_{OT} in Π_{OT}.
- $S \leftrightarrow R$: S and R execute Π_{zk} with R playing the role of the prover P of Π_{zk} and S playing the role of the verifier V of Π_{zk}. The instance is $\left(\tau^*_{OT}, \left\{\tau^{(i,j)}_{OT}, b_{i,j}\right\}_{i\in[\ell_w], j\in[\lambda]}\right)$ and the witness is $\left(r', \beta, r^*_{OT}, \left\{r^{(i,j)}_{OT}, sh_{i,j}, \alpha_{i,j}\right\}_{i\in[\ell_w], j\in[\lambda]}\right)$. If the verifier in Π_{zk} rejects, then S aborts.
- S sends $(\widetilde{r} \oplus m_0, \widetilde{r} \oplus r \oplus m_1)$.

Figure 3:

The statistical indistinguishability of Hyb_4 and Hyb_5 follows from the statistical sender-privacy property of Π_{OT}. The probability that S^* outputs β is negligibly close to ε in this hybrid.

Note that in Hyb_5, the receiver's bit β is information-theoretically hidden in the messages exchanged with S^*. Thus, the probability that S^* guesses β in Hyb_5 is $\frac{1}{2}$. This proves that the probability that the receiver outputs β in Hyb_1 is negligibly close to $\frac{1}{2}$.

Post-Quantum Sender Privacy. Let R^* be a QPT receiver and $\mathcal{A}$ be a QPT adversary such that the following holds for some $m_0 \in \{0,1\}, m_1 \in \{0,1\}$,

$$\mathbb{E}_{\mathsf{View}_{R^*}} \left[\min \{p_0, p_1\}\right] \geq \nu(\lambda),$$

where:

- View_{R^*} is the view of R^*.
- $p_0 = \left|\mathsf{Pr}[\mathcal{A} \text{ wins } \mathsf{G}_0(m_0, m_1)] - \frac{1}{2}\right|$
- $p_1 = \left|\mathsf{Pr}[\mathcal{A} \text{ wins } \mathsf{G}_1(m_0, m_1)] - \frac{1}{2}\right|$

for some non-negligible fucntion $\nu(\lambda)$, where G_0, G_1 are defined with respect to R^* and $\mathcal{A}$ as in Section 5.1. We define $p_0^{(\mathbf{i})}$ to be the absolute difference of the probability that $\mathcal{A}$ wins in the game G_0 and $\frac{1}{2}$ in the hybrid $\mathsf{Hyb}_{\mathbf{i}}$. Similarly, we define $p_1^{(\mathbf{i})}$.

Consider the following hybrids.

$\underline{\mathsf{Hyb}_1}$: This hybrid corresponds to the real execution of the protocol.
By our initial assumption, we have $\mathbb{E}_{\mathsf{View}_{R^*}} \left[\min \{p_0, p_1\}\right] \geq \nu(\lambda)$.

$\underline{\mathsf{Hyb}_2}$: In this hybrid, defer the measurements of the receiver until the end.
The output distributions of Hyb_1 and Hyb_2 are identical. Thus, $\mathbb{E}_{\mathsf{View}_{R^*}} \left[\min \left\{p_0^{(2)}, p_1^{(2)}\right\}\right] \geq \nu(\lambda)$.

$\underline{\mathsf{Hyb}_{3.(i,j)} \text{ for every } i \in [\lambda+2], j \in [\lambda]}$: Instead of computing S, perform the following hybrid extractor as follows.
We first give a description of the registers used in the system.

- $\mathbf{R}_{i,j}$ for $i \in [\lambda+2], j \in [\lambda]$: this consists of the sender S – recall that S is taking the role of the receiver R' of the $(i,j)^{th}$ execution of the OT – randomness used by the extractor in the $(i,j)^{th}$ executions of Π_{OT}.
- $\mathbf{B}_{i,j}$, for $i \in [\lambda+2], j \in [\lambda]$: this is a single-qubit register that contains a bit that is used by the extractor in the $(i,j)^{th}$ execution of the OT protocol.
- **Dec**: it contains the decision register that indicates whether to rewind or not.
- **Aux**: this is initialized with the auxiliary state of the receiver.
- $\mathbf{T}_{i,j}$, for $i \in [\lambda+2], j \in [\lambda]$: it contains the transcripts of the $(i,j)^{th}$ executions of the OT protocol.

- $\mathbf{T}^*$: it contains the transcript of the protocol Π_{zk}.
- $\mathbf{X}$: this is a poly(λ)-qubit ancillary register.

Description of $\mathsf{Hyb}_{3.(i,j)}.\mathsf{Ext}(x, |\Psi\rangle)$: The state of the extractor is initialized as follows:

$$\left(\bigotimes_{i\in[\ell_w],j\in[\lambda]} |0\rangle_{\mathbf{B}_{i,j}}|0\rangle_{\mathbf{R}_{i,j}}|0\rangle_{\mathbf{T}_{i,j}}\right) \otimes |0\rangle_{\mathbf{T}^*} \otimes |\Psi\rangle_{\mathbf{Aux}} \otimes |0\rangle_{\mathbf{Dec}} \otimes |0^{\otimes \mathrm{poly}(\lambda)}\rangle_{\mathbf{X}}$$

- For $i' \in [\lambda+2], j' \in [\lambda]$ such that $(i',j') \geq (i,j)$, perform the following operations in superposition:
  - Let $|\widetilde{\Psi}\rangle$ be the state of the system at the beginning of the $(i,j)^{th}$ execution.
  - Prepare the following state[14]:

$$|0\rangle_{\mathbf{B}_{i,j}}|0\rangle_{\mathbf{R}_{i,j}} \to \frac{1}{\sqrt{2^{\lambda+1}}} \sum_{\beta_{i,j}\in\{0,1\}, s_{\mathsf{OT}}^{(i,j)}\in\{0,1\}^\lambda} |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}$$

  - It then performs the $(i,j)^{th}$ execution of Π_{OT} along with the R^*'s message immediately after the $(i,j)^{th}$ execution of Π_{OT} in superposition. The resulting transcript is stored in the register $\mathbf{T}_{i,j}$. We denote the unitary that performs this step to be $U_{i,j}^{(1)}$.
  - After R^* sends the message immediately after the $(i,j)^{th}$ execution of Π_{OT}, apply the unitary $U_{i,j}^{(2)}$ defined as follows:

$$\begin{aligned} & U_{i,j}^{(2)}|\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|0\rangle_{\mathbf{Dec}} \\ = & \begin{cases} |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|\mathsf{Match}_{i,j}\rangle_{\mathbf{Dec}} \text{ if } \mathsf{acc}_{i,j}=1, \\ |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\rangle_{\mathbf{T}_{i,j}}|+\rangle_{\mathbf{Dec}}, \quad \text{otherwise} \end{cases} \end{aligned}$$

    Here, $\mathsf{Match}_{i,j} = 0$ if and only if $\beta_{i,j} = b_{i,j}$, where $b_{i,j}$ is the bit sent by R^* after the $(i,j)^{th}$ execution of the OT protocol. Moreover, $\mathsf{acc}_{i,j} = 1$ only if R^* has not aborted in $(i,j)^{th}$ OT execution.

    Let $W_{i,j} = \mathsf{Amplifier}\left(U_{i,j}^{(2)}U_{i,j}^{(1)}\right)$, where $\mathsf{Amplifier}$ is defined in Lemma 8. Perform $W_{i,j}|\widetilde{\Psi}\rangle$ to obtain $|\Psi_{i,j}\rangle$.

- For $i' \in [\lambda+2], j' \in [\lambda]$, such that $(i',j') < (i,j)$ perform the $(i',j')^{th}$ execution of Π_{OT} as in the previous hybrid.
- Perform the main execution of Π_{OT} in superposition.
- Perform the execution of Π_{zk} in superposition.

[14] We will assume, without loss of generality, that the length of the random strings used in the OT protocol be λ.

- Measure all the registers except **Aux**. Perform the OT reconstruction on input the measured transcript $\tau_{\mathsf{OT}}^{i,j}$, for $i \in [\lambda+2], j \in [\lambda]$, measured randomness $s_{\mathsf{OT}}^{i,j}$ and receiver's bit $b_{i,j}$. Call the resulting reconstruction output to be $\widetilde{sh_{i,j}}$. Let $\widetilde{u}_i = \oplus_{j=1}^{\ell_w} \widetilde{sh_{i,j}}$. Let (r', β, r^*_{OT}) be the concatenation of all the $\widetilde{u}_i$ bits. If either the S'_{OT}'s inputs to the main execution of Π_{OT} is not $(r', r' \oplus \beta)$ or if S'_{OT}'s randomness is not r^*_{OT} then abort. Otherwise output the state in **Aux** along with w.

From the post-quantum computational receiver privacy of Π_{OT}, it holds that $\mathsf{Hyb}_{3.(i,j)}$ and the previous hybrid are computationally indistinguishable. Thus, the following holds:

$\mathbb{E}_{\mathsf{View}_{R^*}}\left[\min\left\{p_0^{(3.(i,j))}, p_1^{(3.(i,j))}\right\}\right] \geq \nu_{3.(i,j)}(\lambda)$, where $\nu_{3.(i,j)}$ is a non-negligible function.

$\underline{\mathsf{Hyb}_4}$: In the main execution of Π_{OT}, the input of R'_{OT} is set to be 0. Recall that in the previous hybrids, the input of R'_{OT} was r.

From the post-quantum computational receiver privacy of Π_{OT}, it holds that the hybrids Hyb_4 and $\mathsf{Hyb}_{3.(\lambda+2,\lambda)}$ are computationally indistinguishable. Thus, the following holds:

$\mathbb{E}_{\mathsf{View}_{R^*}}\left[\min\left\{p_0^{(4)}, p_1^{(4)}\right\}\right] \geq \nu_4(\lambda)$, where ν_4 is a non-negligible function.

$\underline{\mathsf{Hyb}_5}$: Sample $u \xleftarrow{\$} \{0,1\}$. If $\beta = 1$, set the last message to be $(u, r' \oplus m_1)$. Else if $\beta = 0$, set the last message to be $(r' \oplus m_0, u)$.

From the computational soundness of Π_{zk}, it follows that β extracted from all the $(i,j)^{th}$, for $i \in [\lambda+2], j \in [\lambda]$, executions of Π_{OT} is the same as the β used by the receiver in the main execution of Π_{OT} with probability negligibly close to 1. This further implies that the bit reconstructed by R'_{OT} in the main execution is $\widetilde{r} = r' \oplus (r \cdot \beta)$. Thus, the last message sent by S can be rewritten as follows: $(\widetilde{r} \oplus m_0, \widetilde{r} \oplus r \oplus m_1) = (r' \oplus (r \cdot \beta) \oplus m_0, r' \oplus (r \cdot \beta) \oplus r \oplus m_0)$. If $\beta = 1$, we have the message sent by S to be $(r' \oplus r \oplus m_0, r' \oplus m_1)$. If $\beta = 0$, we have the message sent by S to be $(r' \oplus m_0, r' \oplus r \oplus m_1)$. We now use the fact that r is information-theoretically hidden from the receiver R^* to show that the hybrids Hyb_4 and Hyb_5 are computationally indistinguishable. Thus, the following holds:

$\mathbb{E}_{\mathsf{View}_{R^*}}\left[\min\left\{p_0^{(5)}, p_1^{(5)}\right\}\right] \geq \nu_5(\lambda)$, where ν_5 is a non-negligible function.

But one of the two sender's inputs are information-theoretically hidden from the malicious receiver; in one of the two games G_0 or G_1, the adversary can win only with negligible probability. This contradicts the fact that $\mathbb{E}_{\mathsf{View}_{R^*}}\left[\min\left\{p_0^{(5)}, p_1^{(5)}\right\}\right]$ is non-negligible. Thus, our construction satisfies post-quantum sender privacy.

6 Quantum Proofs of Knowledge for Bounded Concurrent QZK

In this section, we construct a bounded concurrent QZK satisfying quantum proof of knowledge property assuming post-quantum statistical receiver-private oblivious transfer.

We first start with a simpler case: we present a construction of quantum proof of knowledge for a standalone quantum ZK proof system for NP. We then show how to extend this construction to the bounded concurrent QZK setting.

6.1 Construction of (Standalone) QZKPoK

We construct a (standalone) QZKPoK (P, V) for an NP relation $\mathcal{R}(\mathcal{L})$. The following tools are used in our construction:

- A post-quantum statistical receiver-private oblivious transfer protocol, Π_{OT} with associated sender and receiver (S, R) (Section 5) with perfect correctness.

  We say that a transcript τ is valid with respect to sender's randomness r and its input bits (m_0, m_1) if τ can be generated with a sender that uses r as randomness for the protocol and uses (m_0, m_1) as inputs.

- A (standalone) QZK proof system Π_{zk} for $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$. We describe the relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$, parameterized by security parameter λ, below.

$$\mathcal{R}\left(\mathcal{L}_{\mathsf{zk}}\right) = \left\{ \left(\left(x, \{\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\}_{i\in[\ell_w], j\in[\lambda]} \right) ; \left(w, \left\{ r_{\mathsf{OT}}^{(i,j)}, sh_{i,j}, \alpha_{i,j} \right\}_{i\in[\ell_w], j\in[\lambda]} \right) \right) : \right.$$
$$\left. \left(\begin{array}{c} \forall i\in[\ell_w], j\in[\lambda], \\ \tau_{\mathsf{OT}}^{(i,j)} \text{ is valid w.r.t} \\ r_{\mathsf{OT}}^{(i,j)} \text{ and } (((1-b_{i,j})sh_{i,j}+b_{i,j}\cdot\alpha_{i,j}),\ (b_{i,j}sh_{i,j}+(1-b_{i,j})\cdot\alpha_{i,j})) \end{array} \right) \bigwedge \left(\begin{array}{c} \forall i\in[\ell_w], \\ \oplus_{j=1}^{\lambda} sh_{i,j} = w_i \end{array} \right) \bigwedge (x, w) \in \mathcal{R}(\mathcal{L}) \right\}$$

  In other words, the relation checks if the shares $\{sh_{i,j}\}$ used in all the OT executions so far are defined to be such that the XOR of the shares $sh_{i,1}, \ldots, sh_{i,\lambda}$ yields the bit w_i. Moreover $w_1 \cdots w_{\ell_w}$ is the witness to the instance x.

We describe the construction in Figure 4.

Completeness. The completeness follows by the completeness of Π_{zk}.

Quantum Proof of Knowledge. Let P^* be a malicious prover, that on input (x, ρ), can convince V to accept x with non-negligible probability ε. Before we construct a QPT extractor Ext, we first give a description of the registers used in the system.

- $\mathbf{R}_{i,j}$ for $i \in [\ell_w], j \in [\lambda]$: this consists of the receiver randomness used by the extractor in the $(i,j)^{th}$ executions of Π_{OT}.
- $\mathbf{B}_{i,j}$, for $i \in [\ell_w], j \in [\lambda]$: this is a single-qubit register that contains a bit that is used by the extractor in the $(i,j)^{th}$ execution of the OT protocol.
- **Dec**: it contains the decision register that indicates whether to rewind or not.
- **Aux**: this is initialized with the auxiliary state of the prover.
- $\mathbf{T}_{i,j}$, for $i \in [\ell_w], j \in [\lambda]$: it contains the transcripts of the $(i,j)^{th}$ executions of the OT protocol.
- $\mathbf{T}^*$: it contains the transcript of the protocol Π_{zk}.
- $\mathbf{X}$: this is a poly(λ)-qubit ancillary register.

Input of P: Instance $x \in \mathcal{L}$ along with witness w. The length of w is denoted to be ℓ_w.
Input of V: Instance $x \in \mathcal{L}$.

- For every $i \in [\ell_w]$, P samples the shares $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random conditioned on $\oplus_{j=1}^{\lambda} sh_{i,j} = w_i$, where w_i is the i^{th} bit of w.
- For every $i \in [\ell_w]$, P samples the bits $\alpha_{i,1}, \ldots, \alpha_{i,\lambda}$ uniformly at random.
- For $i \in [\ell_w], j \in [\lambda]$, do the following:
  - $P \leftrightarrow V$: P and V execute Π_{OT} with V playing the role of the receiver in Π_{OT} and P playing the role of the sender in Π_{OT}. The input of the receiver in this protocol is 0, while the input of the sender is set to be $(sh_{i,j}, \alpha_{i,j})$ if $b_{i,j} = 0$, otherwise it is set to be $(\alpha_{i,j}, sh_{i,j})$ if $b_{i,j} = 1$, where the bit $b_{i,j}$ is sampled uniformly at random.

    Call the resulting transcript of the protocol to be $\tau_{\mathsf{OT}}^{(i,j)}$ and let $r_{\mathsf{OT}}^{(i,j)}$ be the randomness used by the sender in OT.
  - $P \rightarrow V$: P sends $b_{i,j}$ to V.
- $P \leftrightarrow V$: P and V execute Π_{zk} with P playing the role of the prover of Π_{zk} and V playing the role of the verifier of Π_{zk}. The instance is $\left(x, \left\{\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\right\}_{i\in[\ell_w], j\in[\lambda]}\right)$ and the witness is $\left(w, \left\{r_{\mathsf{OT}}^{(i,j)}, sh_{i,j}, \alpha_{i,j}\right\}_{i\in[\ell_w], j\in[\lambda]}\right)$. If the verifier in Π_{zk} rejects, then V rejects.

Figure 4: Construction of (standalone) QZKPoK for NP.

Description of $\mathsf{Ext}(x, |\Psi\rangle)$: The state of the extractor is initialized as follows:

$$\left(\bigotimes_{i\in[\ell_w], j\in[\lambda]} |0\rangle_{\mathbf{B}_{i,j}} |0\rangle_{\mathbf{R}_{i,j}} |0\rangle_{\mathbf{T}_{i,j}}\right) \otimes |0\rangle_{\mathbf{T}^*} \otimes |\Psi\rangle_{\mathbf{Aux}} \otimes |0\rangle_{\mathbf{Dec}} \otimes |0^{\otimes \mathrm{poly}(\lambda)}\rangle_{\mathbf{X}}$$

- For all $i \in [\ell_w], j \in [\lambda]$, perform the following operations in superposition:
  - Let $|\widetilde{\Psi}\rangle$ be the state of the system at the beginning of the $(i,j)^{th}$ execution.
  - Prepare the following state[15]:

$$|0\rangle_{\mathbf{B}_{i,j}} |0\rangle_{\mathbf{R}_{i,j}} \rightarrow \frac{1}{\sqrt{2^{\lambda+1}}} \sum_{\beta_{i,j}\in\{0,1\}, s_{i,j}^{\mathsf{OT}}\in\{0,1\}^{\lambda}} |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}} |s_{i,j}^{\mathsf{OT}}\rangle_{\mathbf{R}_{i,j}}$$

  - It then performs the $(i,j)^{th}$ execution of Π_{OT} along with the P^*'s message immediately after the $(i,j)^{th}$ execution of Π_{OT} in superposition. The resulting transcript is stored in the register $\mathbf{T}_{i,j}$. We denote the unitary that performs this step to be $U_{i,j}^{(1)}$.

[15] We will assume, without loss of generality, that the length of the random strings used in the OT protocol be λ.

– After P^* sends the message immediately after the $(i,j)^{th}$ execution of Π_{OT}, apply the unitary $U_{i,j}^{(2)}$ defined as follows:

$$U_{i,j}^{(2)}|\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)},b_{i,j}\rangle_{\mathbf{T}_{i,j}}|0\rangle_{\mathbf{Dec}}$$
$$= \begin{cases} |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)},b_{i,j}\rangle_{\mathbf{T}_{i,j}}|\mathsf{Match}_{i,j}\rangle_{\mathbf{Dec}} \text{ if } \mathsf{acc}_{i,j}=1, \\ |\beta_{i,j}\rangle_{\mathbf{B}_{i,j}}|s_{\mathsf{OT}}^{(i,j)}\rangle_{\mathbf{R}_{i,j}}|\tau_{\mathsf{OT}}^{(i,j)},b_{i,j}\rangle_{\mathbf{T}_{i,j}}|+\rangle_{\mathbf{Dec}}, \quad \text{otherwise} \end{cases}$$

Here, $\mathsf{Match}_{i,j}=0$ if and only if $\beta_{i,j}=b_{i,j}$, where $b_{i,j}$ is the bit sent by P^* after the $(i,j)^{th}$ execution of the OT protocol. Moreover, $\mathsf{acc}_{i,j}=1$ only if P^* has not aborted in $(i,j)^{th}$ OT execution.

Let $W_{i,j}=\mathsf{Amplifier}\left(U_{i,j}^{(2)}U_{i,j}^{(1)}\right)$, where $\mathsf{Amplifier}$ is defined in Lemma 8. Perform $W_{i,j}|\widetilde{\Psi}\rangle$ to obtain $|\Psi_{i,j}\rangle$.

- Perform the execution of Π_{zk} in superposition.
- Measure all the registers except **Aux**. Perform the OT reconstruction on input the measured transcript $\tau_{i,j}^{\mathsf{OT}}$, measured randomness $s_{i,j}^{\mathsf{OT}}$ and receiver's bit $b_{i,j}$. Call the resulting reconstruction output to be $\widetilde{sh_{i,j}}$. Let $\widetilde{w}_i=\oplus_{j=1}^{\ell_w}\widetilde{sh_{i,j}}$. Let w be the concatenation of the bits $\widetilde{w}_1,\ldots,\widetilde{w}_{\ell_w}$. If w is not a witness for x, abort. Otherwise output the state in **Aux** along with w.

We now argue that our protocol satisfies the proof of knowledge property. We assume that there is some total ordering defined on (i,j), for $i\in[\ell_w]$ and $j\in[\lambda]$. Without loss of generality, we assume that $(0,0)$ is the least element in this total ordering.

$\underline{\mathsf{Hyb}_1}$: In this hybrid, P^* interacts with the honest verifier V. The verifier V accepts the proof with probability ε.

$\underline{\mathsf{Hyb}_{2.(i,j)}}$, for $i\in[\ell_w], j\in[\lambda]$: We define a hybrid verifier $\mathsf{Hyb}.V_{i,j}$ as follows. Let $|\Phi\rangle$ be the initial state of the system. Compute $|\Psi_{i,j}\rangle=\prod_{(i',j')\leq(i,j)}W_{i',j'}|\Phi\rangle$. From here on, the rest of the iterations of Π_{OT} are computed by interacting with P^* interacting honestly as specified in the real execution. The protocol Π_{zk} is computed by interacting with P^* honestly as in the real execution. Finally, $\mathsf{Hyb}.V_{i,j}$ outputs its decision.

The probability that $\mathsf{Hyb}.V_{i,j}$ accepts is negligibly close to ε. Moreover, from the statistical security against senders, it follows that the state output by P^* in this hybrid is close in trace distance to the state output by P^* in the previous hybrid. We omit the proof since it essentially follows the same line of argument used in Section 4.2.1.

$\underline{\mathsf{Hyb}_3}$: We define a hybrid verifier $\mathsf{Hyb}.V_3$ as follows. Let $|\Phi\rangle$ be the initial state of the system. Compute $|\Psi_{i,j}\rangle=\prod_{(i\in[\ell_w],j\in[\lambda])}W_{i,j}|\Phi\rangle$. The protocol Π_{zk} is computed by interacting with P^* honestly as in the real execution. Finally, $\mathsf{Hyb}.V_3$ outputs its decision.

The probability that $\mathsf{Hyb}.V_3$ accepts is negligibly close to ε. This follows from the fact that $\mathsf{Hyb}.V_3$ is identical to $\mathsf{Hyb}.V_{i^*,j^*}$, where (i^*,j^*) is the highest element in the total ordering.

Moreover, the state output by P^* in this hybrid is the same as the state output by P^* in the previous hybrid.

$\underline{\mathsf{Hyb}_4}$: Define a hybrid verifier $\mathsf{Hyb}.V_4$ as follows: it executes the hybrid verifier $\mathsf{Hyb}.V_3$ until the step just before it outputs its decision. Let $\widetilde{sh_{i,j}}$ be the share output by the reconstruction algorithm of the receiver of Π_{OT}. Let $\widetilde{w}_i = \oplus_{j=1}^{\ell_w} \widetilde{sh_{i,j}}$. Let w be the concatenation of the bits $\widetilde{w}_1, \ldots, \widetilde{w}_{\ell_w}$. If w is not a witness for x, abort. Otherwise, output the decision of $\mathsf{Hyb}.V_3$.

The probability that $\mathsf{Hyb}.V_4$ accepts is negligibly close to ε. To see this, note that it is sufficient to argue that $|p_3 - p_4| \leq \mathsf{negl}(\lambda)$, where p_3 is the probability with which $\mathsf{Hyb}.V_3$ aborts and p_4 is the probability with which $\mathsf{Hyb}.V_4$ aborts. This fact follows from the soundness of Π_{zk}. Moreover, the output state of the prover in Hyb_3 is the same as the output state of the prover in Hyb_4.

Note that the probability that the extractor Ext outputs a valid witness w is the same as the probability that the hybrid verifier $\mathsf{Hyb}.V_4$ accepts. Moreover, the state output by P^* when interacting with Ext is exactly the same as the state output by P^* in Hyb_4.

6.1.1 (Standalone) Quantum Zero-Knowledge

Suppose $(x, w) \in \mathcal{R}(\mathcal{L})$. Suppose V^* is a QPT verifier, that on input $(x, |\Psi\rangle)$, interacts with the honest prover $P(x, w)$. We construct a simulator Sim that takes as input $(x, |\Psi\rangle)$ such that the output distribution of the simulator is computationally indistinguishable from the output distribution of V^*.

$\underline{\text{Description of } \mathsf{Sim}(x, |\Psi\rangle)}$:

- For every $i \in [\ell_w]$, Sim samples $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random.
- For $i \in [\ell_w], j \in [\lambda]$, do the following:
  - Sim and V^* execute Π_{OT}. The verifier V^* takes the role of the receiver of Π_{OT} and Sim takes the role of the sender. The input of the sender is $(sh_{i,j}, sh_{i,j})$.
  - Sim samples a random bit $b_{i,j}$ and sends to V^*.
- Let the state of the verifier, at this point of this protocol, be $\widetilde{|\Psi\rangle}$. Let $\mathsf{Sim}_{\mathsf{zk}}$ be the Π_{zk} simulator associated with the Π_{zk} verifier $\widetilde{V^*}$, where $\widetilde{V^*}$ is the code used by V^* in the execution of the protocol Π_{zk}. Compute $\mathsf{Sim}_{\mathsf{zk}}$ on input the state $\widetilde{|\Psi\rangle}$ and the instance $\left(x, \left\{\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\right\}\right)$.
- Output the transcript of the protocol along with the private state of the verifier V^*.

We now prove that the state output by V^* when interacting with the honest prover $P(x, w)$ is computationally indistinguishable from the state output by $\mathsf{Sim}(x, |\Psi\rangle)$. Consider the following hybrids. As before we consider a total ordering on (i, j), for $i \in [\ell_w], j \in [\lambda]$.

$\underline{\mathsf{Hyb}_1}$: In this hybrid, P and V^* interact with each other. The output of this hybrid is the output of V^*.

$\underline{\mathsf{Hyb}_2}$: We define another hybrid prover $\mathsf{Hyb}_2.P$ that behaves as follows: it simulates the protocol Π_{zk} using the simulator $\mathsf{Sim}_{\mathsf{zk}}$ as given in the description of Sim. The rest of the protocol is the same as in the hybrid Hyb_1.

The computational indistinguishability of Hyb_1 and Hyb_2 follows from the quantum zero-knowledge property of Π_{zk}.

$\underline{\mathsf{Hyb}_3}$: The output of this hybrid is the output of $\mathsf{Sim}(x, |\Psi\rangle)$.

Claim 39. *Assuming the post-quantum sender-privacy of Π_{OT}, the output of Hyb_2 is computationally indistinguishable from the output of Hyb_3.*

Proof. Let $\mathcal{A}$ be the distinguisher distinguishing Hyb_2 and Hyb_3. We are going to prove that $\mathcal{A}$ can only distinguish with negligible probability. Consider the intermediate hybrids.

$\underline{\mathsf{Hyb}_{2.1}}$: This is identical to Hyb_2.

We now consider a series of hybrids that are defined with respect to $\mathcal{A}$.

$\underline{\mathsf{Hyb}^{\mathcal{A}}_{2.2.(i^*,j^*)} \text{ for } i^* \in [\ell_w], j^* \in [\lambda - 1]}$: We say that a hybrid prover $\mathsf{Hyb}_{2.2.(i^*,j^*)}.P$, uses $\left(\left\{\widehat{b}_{i,j}\right\}_{(i,j)\leq(i^*,j^*)}\right)$, if the following holds: it executes the prover as in $\mathsf{Hyb}_{2.1}$, except, for $(i,j) \leq (i^*,j^*)$, it uses the input $(sh_{i,j}, sh_{i,j})$ if $\widehat{b}_{i,j} \neq b_{i,j}$ or uses the input $(\alpha_{i,j}, \alpha_{i,j})$ if $\widehat{b}_{i,j} = b_{i,j}$.

Now, execute the above hybrid prover $\mathsf{Hyb}_{2.2.(i^*,j^*)}.P$ by adaptively choosing $\left(\left\{\widehat{b}_{i,j}\right\}_{(i,j)\leq(i^*,j^*)}\right)$ (as a function of the current state of the verifier and $\mathcal{A}$) such that the output distributions of $\mathsf{Hyb}_{2.2.(i^*,j^*)}.P$ and $\mathsf{Hyb}_{2.2.(i^*,j^*)-1}.P$ cannot be distinguished by $\mathcal{A}$. If such a set of bits cannot be adaptively chosen then abort. Otherwise, this hybrid prover interacts with the verifier and the output of this hybrid is set to be the output of the verifier.

Claim 40. *The hybrid $\mathsf{Hyb}^{\mathcal{A}}_{2.2.(i^*,j^*)}$ aborts with negligible probability.*

Proof. We prove this by induction.

Base Case: $(i^*,j^*) = (1,1)$. We prove that $\mathsf{Hyb}^{\mathcal{A}}_{2.2.(1.1)}$ aborts with negligible probability. From the post-quantum sender privacy property of Π_{OT} (Definition 34), it follows that upon fixing the view of the verifier until the last message of execution of $(1,1)^{th}$ OT protocol, there exists a bit $\widehat{b}$, with probability negligibly close to 1, such that the adversary cannot win the Game $G_{\widehat{b}}\left(m_0^{(1,1)}, m_1^{(1,1)}\right)$ (specified in Definition 34) where, $\left(m_0^{(1,1)}, m_1^{(1,1)}\right) = (sh_{1,1}, \alpha_{1,1})$ is $b_{1,1} = 0$ and $\left(m_0^{(1,1)}, m_1^{(1,1)}\right) = (\alpha_{1,1}, sh_{1,1})$ is $b_{1,1} = 1$.

Induction Hypothesis. Suppose this statement is true for all $(i,j) < (i^*,j^*)$. We prove this statement to be true even for (i^*,j^*) using proof by contradiction.

Suppose $\mathsf{Hyb}_{2.2.(i^*,j^*)}$ aborts with non-negligible probability then we design a QPT adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$, that receives as input non-uniform quantum advice, and breaks the post-quantum sender privacy property of Π_{OT}.

We first define the non-uniform advice as follows: it computes the interaction between the hybrid prover $\mathsf{Hyb}_{2.2.(i^*,j^*)-1}.P$ and the verifier V^*, until the $((i^*,j^*)-1)^{th}$ execution of OT. It

outputs the private state of $\mathsf{Hyb}_{2.2.(i^*,j^*)-1}.P$ and the private state of V^*. Call this state $|\Psi_{adv}\rangle$.

$\mathcal{B}_1$, upon receiving $|\Psi_{adv}\rangle$, takes the role of the receiver and interacts with the external challenger until the receiver's last message of the $(i^*,j^*)^{th}$ execution of Π_{OT}. $\mathcal{B}_1$ uses the code of V^* to interact with the external challenger. The external challenger on the other receives as input $m_0^{(i^*,j^*)} = sh_{i^*,j^*}$ and $m_1^{(i^*,j^*)} = \alpha_{i^*,j^*}$ if $b_{i^*,j^*} = 0$ or $m_1^{(i^*,j^*)} = sh_{i^*,j^*}$ and $m_0^{(i^*,j^*)} = \alpha_{i^*,j^*}$ if $b_{i^*,j^*} = 1$, from $\mathcal{B}_1$, where $sh_{i^*,j^*}, \alpha_{i^*,j^*}, b_{i^*,j^*}$ are generated as in $\mathsf{Hyb}_{2.1}$. It then outputs the state $|\Psi_1\rangle$ of V^* obtained after the receiver sends the last message in the $(i^*,j^*)^{th}$ execution of Π_{OT}.

$\mathcal{B}_2$, upon receiving $|\Psi_1\rangle$, computes the rest of the executions of Π_{OT} and Π_{zk} by emulating the interaction between the hybrid prover $\mathsf{Hyb}_{2.2.(i^*,j^*)}.P$ and the verifier V^*. It then inputs the final state of V^* to $\mathcal{A}$. The output of $\mathcal{B}_2$ is set to be the output of $\mathcal{A}$.

Our initial assumption was that the $\mathsf{Hyb}_{2.2.(i^*,j^*)}$ aborts with non-negligible probability. This means that the adversary $\mathcal{A}$ can distinguish with non-negligible probability (over the view of the verifier until the $(i^*,j^*)^{th}$ OT execution) both the games – that is, distinguishing $(m_0^{(i^*,j^*)}, m_1^{(i^*,j^*)})$ from $(m_1^{(i^*,j^*)}, m_1^{(i^*,j^*)})$ (Game 0) as well as distinguishing $(m_0^{(i^*,j^*)}, m_1^{(i^*,j^*)})$ from $(m_0^{(i^*,j^*)}, m_0^{(i^*,j^*)})$ (Game 1) – with probability significantly greater than $\frac{1}{2}$. This in turn means that $\mathcal{B}$ can break the post-quantum sender privacy property of Π_{OT} with non-negligible probability. Thus, we arrived at a contradiction. □

$\underline{\mathsf{Hyb}^{\mathcal{A}}_{2.3}}$: This hybrid is the same as $\mathsf{Hyb}_{2.2.(\ell_w,\lambda-1)}$, except that the hybrid prover will abort if for all $i \in [\ell_w]$ and all $j \in [\lambda-1]$, it holds that $b_{i,j} \neq \widehat{b_{i,j}}$.

Claim 41. *The hybrids $\mathsf{Hyb}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ and $\mathsf{Hyb}^{\mathcal{A}}_{2.3}$ can be distinguished by $\mathcal{A}$ with only negligible probability.*

Proof. To prove this, we consider an alternate hybrid prover in $\mathsf{Hyb}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ which samples, for any i, $b_{i,j} \xleftarrow{\$} \{0,1\}$ at the end of first $(\lambda-1)$ iterations of Π_{OT}. It then sets the input to the λ^{th} iteration of Π_{OT} to be $\left(\oplus_{j=1}^{\lambda-1} m_{b_{i,j}}^{(i,j)}, u\right)$ with probability $\frac{1}{2}$ or $\left(u, \oplus_{j=1}^{\lambda-1} m_{b_{i,j}}^{(i,j)}\right)$ with probability $\frac{1}{2}$, where $u \xleftarrow{\$} \{0,1\}$ and $\{m_0^{(i,j)}, m_1^{(i,j)}\}_{j\in[\lambda-1]}$ are the inputs used in the first $\lambda-1$ iterations of Π_{OT}. Note that the output distribution of $\mathsf{Hyb}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ remains the same even with this change.

Since the $b_{i,j}$'s, for $j \leq \lambda-1$, are sampled after the $\widehat{b}_{i,j}$'s are decided, the probability that $\widehat{b}_{i,j} \neq b_{i,j}$ is $\frac{1}{2}$ for any $i \in [\ell_w], j \in [\lambda-1]$. Thus, the probability that $\left(\forall i \in [\ell_w], j \in [\lambda-1], b_{i,j} \neq \widehat{b}_{i,j}\right)$ is $\leq \frac{\ell_w}{2^{\lambda-1}}$. Conditioned on this bad event, the output distributions of $\mathsf{Hyb}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ and $\mathsf{Hyb}_{2.3}$ are identical. Thus, $\mathcal{A}$ cannot distinguish the hybrids $\mathsf{Hyb}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ and $\mathsf{Hyb}^{\mathcal{A}}_{2.3}$.

□

$\underline{\mathsf{Hyb}^{\mathcal{A}}_{2.4.i^* \text{ for all } i\in[\ell_w]}}$: This hybrid is the same as $\mathsf{Hyb}^{\mathcal{A}}_{2.3}$ except that the hybrid prover $\mathsf{Hyb}_{2.4.i^*}.P$ is additionally parameterized by $\left(\left\{\widehat{b}_{i,\lambda}\right\}_{i\leq i^*}\right)$. The only change from the previous hybrid is that the hybrid prover, for $i \leq i^*$, use the input $(sh_{i,\lambda}, sh_{i,\lambda})$ if $\widehat{b}_{i,\lambda} \neq b_{i,\lambda}$ or use $(\alpha_{i,\lambda}, \alpha_{i,\lambda})$ if $\widehat{b}_{i,\lambda} = b_{i,\lambda}$.

Now, consider a hybrid prover $\mathsf{Hyb}_{2.4.i^*}.P$, parameterized by $\left(\left\{\widehat{b}_{i,j}\right\}_{(i\leq i^*)\vee(j\leq\lambda-1)}\right)$, where $\left(\left\{\widehat{b}_{i,j}\right\}_{(i,j)\leq(i^*,j^*)}\right)$, is defined to be such that the output distributions of $\mathsf{Hyb}_{2.i^*}.P$ and $\mathsf{Hyb}_{2.4.i^*-1}.P$ cannot be distinguished by $\mathcal{A}$. If such a hybrid prover does not exist, then abort. Otherwise, this hybrid prover interacts with the verifier and the output of this hybrid is set to be the output of the verifier.

Claim 42. *The hybrid* $\mathsf{Hyb}_{2.4.i^*}$ *aborts with negligible probability.*

We omit the proof of the above claim since it uses the same inductive argument as the proof of Claim 44.

$\underline{\mathsf{Hyb}_{2.5}}$: This hybrid is the same as Hyb_3, i.e. the output of the simulator.

Conditioned on $\mathsf{Hyb}_{2.4.\ell_w}$ not aborting, the output distributions of $\mathsf{Hyb}_{2.4.\ell_w}$ and $\mathsf{Hyb}_{2.5}$ are the same. This follows from the fact that if $\mathsf{Hyb}_{2.4.\ell_w}$ does not abort then the distribution of the inputs used in all the OT executions in the hybrids $\mathsf{Hyb}_{2.4.\ell_w}$ and $\mathsf{Hyb}_{2.5}$ are the same. Thus, $\mathcal{A}$ can disitnguish $\mathsf{Hyb}_{2.4.\ell_w}$ and $\mathsf{Hyb}_{2.5}$ only with negligible probability.

From the above hybrids, it follows that $\mathcal{A}$ can distinguish the hybrids $\mathsf{Hyb}_{2.1}$ and $\mathsf{Hyb}_{2.5}$ with only negligible probability.

□

6.2 Extending to Bounded Concurrent QZK Setting

We show how to adopt the construction in Section 6.1 to the bounded concurrent setting.

The construction of bounded concurrent quantum proof of knowledge system is the same as Figure 4, except that we instantiate Π_{zk} with a modified version of the bounded concurrent QZK for NP construction in Section 4.

Modified Bounded Concurrent QZK for NP Construction. We modify the construction in Section 4 as follows: Let M be the round complexity of the statistical receiver private OT protocol and let $M = \lambda^c$ for some constant c, where λ is the security parameter used in the OT protocol. Let λ' denote a different security parameter used in Π_{zk} such that $\lambda' - M \geq \lambda$. We set the threshold of matched slots needed in the WI protocol from Section 4, to instead be, $60Q^7\lambda' + Q^4\lambda' - M$, provided we set $\lambda' \gg M$.

The completeness and soundness proofs of this modified construction are the same as the ones in Section 4. Even the quantum zero-knowledge property is the same as before. However, we will need the simulator to satisfy a stronger property defined next.

Strong QZK Simulator. We explain the differences between the strong QZK simulator and the simulator Sim defined in Section 4. The strong simulator proceeds as follows:

1. It simulates block-by-block similarly to Sim, but instead of using $|+\rangle_{\mathbf{Dec}}$ only in the decision bit of the registers that aborted, it can choose to use $|+\rangle_{\mathbf{Dec}}$ on other transcripts as well. This decision is based on an efficiently computable function f. For example, on a transcript t, it can set $\mathbf{Dec}$ to $|+\rangle_{\mathbf{Dec}}$ conditioned on $f(t) = 1$.

2. After rewinding a block, it measures the transcript of that block instead of waiting until the end to measure. Furthermore, it keeps tracks of the total number of blocks on which the measurement outcomes correspond to a transcript in which it used $|+\rangle_{\mathbf{Dec}}$.

3. If at any point, the number of block measurement outcomes that correspond to $|+\rangle_{\mathbf{Dec}}$ transcripts is greater than M, it aborts.

Conditioned on the strong simulator not aborting in Step 3, its output is computationally indistinguishable from the output of Sim.

Arguing Bounded Concurrent Quantum Zero-Knowledge for Figure 4. In the proof of bounded concurrent quantum zero-knowledge, we now need to handle Q-session verifiers, where Q is the number of sessions associated with the protocol.

The description of the simulator is the same as in the description of the simulator in Section 6.1.1 except that we now execute the bounded concurrent strong simulator described above for Π_{zk} instead of the standalone ZK simulator.

We describe the hybrids below. Our description of hybrids and the proofs of indistinguishability between the hybrids closely follows the structure of the proof in Section 6.1.1 and hence we only highlight the main differences.

$\underline{\mathsf{Hyb}_1}$: Same as Hyb_1 in Section 6.1.1.

$\underline{\mathsf{Hyb}_2}$: We define another hybrid prover $\mathsf{Hyb}_2.P$ that behaves as follows: it simulates the protocol $\overline{\Pi}_{\mathsf{zk}}$ using the bounded concurrent simulator $\mathsf{Sim}_{\mathsf{zk}}$ as given in the description of Sim. The rest of the protocol is the same as in the hybrid Hyb_1.

The computational indistinguishability of Hyb_1 and Hyb_2 follows from the bounded concurrent quantum zero-knowledge property of Π_{zk}.

$\underline{\mathsf{Hyb}_3}$: The output of this hybrid is the output of the simulator.

Claim 43. *Assuming the post-quantum sender-privacy of Π_{OT}, the output of Hyb_2 is computationally indistinguishable from the output of Hyb_3.*

Proof. Let $\mathcal{A}$ be the distinguisher distinguishing Hyb_2 and Hyb_3. We are going to prove that $\mathcal{A}$ can only distinguish with negligible probability. Consider the intermediate hybrids.

$\underline{\mathsf{Hyb}_{2.1}}$: This is identical to Hyb_2.

We now consider a sequence of hybrids. Each hybrid in this sequence is parameterized by the number of OT executions and the number of sessions. We first replace the inputs of all the OTs corresponding to one session before we move on to the next session. That is, each hybrid is of the form $\mathsf{Hyb}_{2.2.i.j.k}$. We first start with $i = 1, j = 1, k = 1$. We then iterate over $j = 1, \ldots, \lambda - 1$, and then we increment i. We keep doing this, until we reach the hybrid $\mathsf{Hyb}_{2.2.\ell_w.\lambda-1.k}$. The next hybrid is $\mathsf{Hyb}_{2.3.k}$. After this, we have the hybrid, $\mathsf{Hyb}_{2.4.i.k}$, where $i = 1$ and $k = 1$. We then iterate over $i = 1, \ldots, \ell_w$. Immediately after the hybrid $\mathsf{Hyb}_{2.4.\ell.k}$, we have the hybrid $\mathsf{Hyb}_{2.5.k}$. At this point, all the OTs corresponding to the first session have been replced. Immediately after this hybrid, we then move on to the hybrid $\mathsf{Hyb}_{2.2.i.j.k}$, where $i = 1, j = 1, k = 2$. We then continue as above, until

we reach the hybrid $\mathsf{Hyb}_{2.5.Q}$. The hybrid that follows $\mathsf{Hyb}_{2.5.Q}$ is the hybrid $\mathsf{Hyb}_{2.6}$.

$\underline{\mathsf{Hyb}^{\mathcal{A}}_{2.2.i.j.k} \text{ for } i \in [\ell_w], j \in [\lambda - 1], k \in [Q]}$: We say that a prover, uses $\left(\left\{\widehat{b}_j\right\}_{j \leq i}\right)$ in a particular transcript, if the following holds: in superposition, execute the prover as in $\mathsf{Hyb}_{2.1}$, except that in the first $j \leq i$ OT executions to end in the transcript being generated, it uses the input (sh_j, sh_j) if $\widehat{b}_j \neq b_j$ or uses the input (α_j, α_j) if $\widehat{b}_j = b_j$.

We define a hybrid prover $\mathsf{Hyb}_{2.2.i.j.k}.P$ as follows:

- For $k' < k$, it chooses the input to the $(i,j)^{th}$ execution to be $(\alpha_{i,j}, \alpha_{i,j})$, where $\alpha_{i,j}$ is sampled uniformly at random.
- For $k' > k$, it chooses the inputs to the OT executions as done by the prover in $\mathsf{Hyb}_{2.1}$.
- For $k' = k$, the hybrid prover, *in superposition*, adaptively uses $\left(\left\{\widehat{b}_{(i',j')}\right\}_{(i',j') \leq (i,j)}\right)$ such that the output distributions of $\mathsf{Hyb}_{2.2.(i,j).k}.P$ and $\mathsf{Hyb}_{2.2.(i,j)-1.k}.P$ (if $(i,j) = (1,1)$ then the hybrid $\mathsf{Hyb}_{2.2.(i,j).k}.P$ is defined to be $\mathsf{Hyb}_{2.4.k-1}.P$) cannot be distinguished by $\mathcal{A}$. That is, since the whole protocol is being executed in superposition, as a function of each term in the superposition, the bits $\left(\left\{\widehat{b}_{(i',j')}\right\}_{(i',j') \leq (i,j)}\right)$ are adaptively determined and stored in a separate register to be used by the hybrid prover. If the entire sequence of bits cannot be determined, then store $\perp$ in the same register. At the end of the protocol, we measure this register. If the outcome is $\perp$ then abort, otherwise, measure the registers storing the transcript, trace out all the registers except the register containing the auxiliary state of the verifier and output the measured transcript along with the residual auxiliary state of the verifier.

Claim 44. *The hybrid $\mathsf{Hyb}^{\mathcal{A}}_{2.2.i.j.k}$ aborts with negligible probability.*

Proof. We prove this by induction.

Base Case: $(i,j) = (1,1)$. We prove that $\mathsf{Hyb}^{\mathcal{A}}_{2.2.1.1.k}$ aborts with negligible probability. Suppose not. We demonstrate a reduction that breaks the sender privacy property of OT with non-negligible probability. The goal of the reduction is to win in both the games with non-negligible probability: in the first game, it needs to distinguish the case when the challenger uses the input (m_0, m_1), where $(m_0, m_1) = (sh_{1,1}, \alpha_{1,1})$ with probability $\frac{1}{2}$ and $(m_0, m_1) = (\alpha_{1,1}, sh_{1,1})$ or when it uses the input $(sh_{1,1}, sh_{1,1})$. In the second game, it needs to distinguish the case when the challenger uses the input (m_0, m_1), where (m_0, m_1) is defined as above, versus the case when it uses the input $(\alpha_{1,1}, \alpha_{1,1})$.

We describe a reduction that does the following: just like the simulator of the bounded concurrent QZK, it divides the entire protocol transcript into blocks $B_1, \ldots, B_L$, where L is as defined in Π_{zk}. For every block B_i, it does the following: it executes the simulator in superposition. If it encounters a message of $(1,1)^{th}$ OT, it computes stops simulating the rest of the block. It then puts $|+\rangle$ state in the decision register. Otherwise, it continues the simulation until the end of the block. It then performs Watrous rewinding. At the end, it measures the trascript. There are two cases:

- If the block B_i has completed it's execution and if no message in the $(1,1)^{th}$ OT execution has been encountered so far, then continue to the block B_{i+1}.
- If a message in the $(1,1)^{th}$ OT execution has been encountered then forward to the challenger of the OT protocol. Use the response by the challenger to continue the execution of B_i, albeit by interacting V^*, rather running V^* in superposition. Once this is completed, move on to the block B_{i+1}.

Finally, after all the blocks are executed, the transcript along with the final private state of the verifier is input to $\mathcal{A}$.

If the challenger uses the input (m_0, m_1) then it corresponds to the hybrid $\mathsf{Hyb}_{2.4.k-1}$ (if $k = 1$, then $\mathsf{Hyb}_{2.4.k-1}$ is the hybrid $\mathsf{Hyb}_{2.1}$) and if the challenger uses the input $(sh_{1,1}, sh_{1,1})$ or the input $(\alpha_{1,1}, \alpha_{1,1})$ then it corresponds to the hybrid $\mathsf{Hyb}_{2.2.1.1.k}$.

If $\mathcal{A}$ can distinguish the hybrids $\mathsf{Hyb}_{2.2.1.1.k}$ and $\mathsf{Hyb}_{2.4.k-1}$ with non-negligible probability then even the reduction can break the sender privacy property with non-negligible probability.

Induction Hypothesis. Suppose this statement is true for all $(i', j') < (i, j)$. We then show this to be true even for (i, j).

Suppose this is not true. We then design a reduction that violates the sender privacy of OT with non-negligible probability. The reduction essentially is defined along the same lines as the base case, except that the first $(i, j) - 1$ OT executions of the k^{th} verifier are generated as non-uniform advice. That is, the advice generation algorithm executes the protocol in superposition and in each term of the superposition, it halts after the final $((i, j) - 1)^{th}$ execution of the k^{th}. Until this point, the inputs to the $(i', j')^{th}$ OT execution, for $(i', j') < (i, j)$, is set to be either $(sh_{(i',j')}, sh_{(i',j')})$ or $(\alpha_{(i',j')}, \alpha_{(i',j')})$, depending on the distinguishing probability of $\mathcal{A}$. Finally, the advice generation measures the transcript and outputs the transcript along with the residual state.

The reduction then performs block-by-block execution of the protocol and interacts with the challenger as in the base case. The final state of the verifier along with the transcript of the entire protocol is input to $\mathcal{A}$.

As in the base case, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible probability then the reduction can also violate the sender privacy property with non-negligible probability.

□

$\underline{\mathsf{Hyb}^{\mathcal{A}}_{2.3.k} \text{ for } k \in [Q]}$: This hybrid is the same as $\mathsf{Hyb}_{2.2.\ell_w,\lambda-1.k}$, except that the hybrid prover will abort if for all $i \in [\ell_w]$ and all $j \in [\lambda - 1]$, it holds that $b_{i,j} \neq \widehat{b_{i,j}}$.

Claim 45. *The hybrids $\mathsf{Hyb}^{\mathcal{A}}_{2.2.\ell_w,\lambda-1.k}$ and $\mathsf{Hyb}^{\mathcal{A}}_{2.3.k}$ can be distinguished by $\mathcal{A}$ with only negligible probability.*

Proof. To prove this, we consider an alternate hybrid prover in $\mathsf{Hyb}^{\mathcal{A}}_{2.2.\ell_w,\lambda-1.k}$ which samples, for any i, $b_{i,j} \xleftarrow{\$} \{0,1\}$ at the end of first $(\lambda - 1)$ iterations of Π_{OT}. It then sets the input to the λ^{th} iteration of Π_{OT} to be $\left(\oplus_{j=1}^{\lambda-1} m^{(i,j)}_{b_{i,j}}, u\right)$ with probability $\frac{1}{2}$ or $\left(u, \oplus_{j=1}^{\lambda-1} m^{(i,j)}_{b_{i,j}}\right)$ with probability $\frac{1}{2}$, where $u \xleftarrow{\$} \{0,1\}$ and $\{m^{(i,j)}_0, m^{(i,j)}_1\}_{j\in[\lambda-1]}$ are the inputs used in the first $\lambda - 1$ iterations of Π_{OT}. Note that the output distribution of $\mathsf{Hyb}^{\mathcal{A}}_{2.2.(\ell_w,\lambda-1)}$ remains the same even with this change.

Since the $b_{i,j}$'s, for $j \leq \lambda - 1$, are sampled after the $\widehat{b}_{i,j}$'s are decided, the probability that $\widehat{b}_{i,j} \neq b_{i,j}$ is $\frac{1}{2}$ for any $i \in [\ell_w], j \in [\lambda - 1]$. Thus, the probability that $\left(\forall i \in [\ell_w], j \in [\lambda - 1], b_{i,j} \neq \widehat{b}_{i,j}\right)$ is $\leq \frac{\ell_w}{2^{\lambda-1}}$. Conditioned on this bad event, the output distributions of $\mathsf{Hyb}^{\mathcal{A}}_{2.2.\ell_w.\lambda-1.k}$ and $\mathsf{Hyb}_{2.3}$ are identical. Thus, $\mathcal{A}$ cannot distinguish the hybrids $\mathsf{Hyb}^{\mathcal{A}}_{2.2.\ell_w.\lambda-1.k}$ and $\mathsf{Hyb}^{\mathcal{A}}_{2.3.k}$.

□

$\mathsf{Hyb}^{\mathcal{A}}_{2.4.i^*.k}$ for all $i^* \in [\ell_w], k \in [Q]$: This hybrid is the same as $\mathsf{Hyb}^{\mathcal{A}}_{2.3}$ except that the hybrid prover $\mathsf{Hyb}_{2.4.i^*}.P$ is additionally parameterized by $\left(\left\{\widehat{b}_{i,\lambda}\right\}_{i \leq i^*}\right)$. The only change from the previous hybrid is that the hybrid prover, for $i \leq i^*$, use the input $(sh_{i,\lambda}, sh_{i,\lambda})$ if $\widehat{b}_{i,\lambda} \neq b_{i,\lambda}$ or use $(\alpha_{i,\lambda}, \alpha_{i,\lambda})$ if $\widehat{b}_{i,\lambda} = b_{i,\lambda}$.

Now, consider a hybrid prover $\mathsf{Hyb}_{2.4.i^*}.P$, parameterized by $\left(\left\{\widehat{b}_{i,j}\right\}_{(i \leq i^*) \vee (j \leq \lambda-1)}\right)$, where $\left(\left\{\widehat{b}_{i,j}\right\}_{(i,j) \leq (i^*,j^*)}\right)$, is defined to be such that the output distributions of $\mathsf{Hyb}_{2.4.i^*.k}.P$ and $\mathsf{Hyb}_{2.4.i^*-1.k}.P$ cannot be distinguished by $\mathcal{A}$. If such a hybrid prover does not exist, then abort. Otherwise, this hybrid prover interacts with the verifier and the output of this hybrid is set to be the output of the verifier.

Claim 46. *The hybrid* $\mathsf{Hyb}_{2.4.i^*.k}$ *aborts with negligible probability.*

We omit the proof of the above claim since it uses the same inductive argument as the proof of Claim 44.

$\mathsf{Hyb}_{2.5.k}$ for $k \in [Q]$: We define a hybrid prover that does the following:

- For $k' \leq k$, it chooses the input to the $(i,j)^{th}$ execution to be $(\alpha_{i,j}, \alpha_{i,j})$, where $\alpha_{i,j}$ is sampled uniformly at random.
- For $k' > k$, it chooses the inputs to the OT executions as done by the prover in $\mathsf{Hyb}_{2.1}$.

Conditioned on $\mathsf{Hyb}_{2.4.\ell_w.k}$ not aborting, the output distributions of $\mathsf{Hyb}_{2.4.\ell_w.k}$ and $\mathsf{Hyb}_{2.5.k}$ are the same. This follows from the fact that if $\mathsf{Hyb}_{2.4.\ell_w}$ does not abort then the distribution of the inputs used in all the OT executions in the hybrids $\mathsf{Hyb}_{2.4.\ell_w.k}$ and $\mathsf{Hyb}_{2.5.k}$ are the same. Thus, $\mathcal{A}$ can disitnguish $\mathsf{Hyb}_{2.4.\ell_w.k}$ and $\mathsf{Hyb}_{2.5.k}$ only with negligible probability.

$\mathsf{Hyb}_{2.6}$: This hybrid is the same as Hyb_3, i.e. the output of the simulator.

The output distributions of $\mathsf{Hyb}_{2.5.Q}$ and $\mathsf{Hyb}_{2.6}$ are identical.

From the above hybrids, it follows that $\mathcal{A}$ can distinguish the hybrids $\mathsf{Hyb}_{2.1}$ and $\mathsf{Hyb}_{2.6}$ with only negligible probability.

□

7 Bounded Concurrent QZK for QMA

We show a construction of bounded concurrent QZK for QMA. Our starting point is the QZK protocol for QMA from [BJSW16], which constructs QZK for QMA from QZK for NP, commitments and a coin-flipping protocol. We make the following observation about the proof of quantum zero-knowledge in [BJSW16]. Only two ingredients in their construction require rewinding by the QZK simulator: the coin-flipping protocol and the QZK for NP protocol. The rest of their simulation is straightline. Furthermore, we can combine the coin-flipping step with the QZK for NP step, leaving us with a straightline simulation that only requires rewinding in one place (at the QZK for NP step). As we will show, it then suffices to instantiate the QZK for NP with the bounded concurrent QZK for NP protocol in Section 4.

7.1 Bounded Concurrent QZK for QMA

We first recall the QZK for QMA construction from [BJSW16]. Their protocol is specifically designed for the QMA promise problem called k-local Clifford Hamiltonian, which they showed to be QMA-complete for $k = 5$. We restate it here for completeness.

Definition 47 (k-local Clifford Hamiltonian Problem [BJSW16])**.** *For all $i \in [m]$, let $H_i = C_i|0^{\otimes k}\rangle\langle 0^{\otimes k}|C_i^\dagger$ be a Hamiltonian term on k-qubits where C_i is a Clifford circuit.*

- *Input: $H_1, H_2, \ldots, H_m$ and strings 1^p, 1^q where p and q are positive integers satisfying $2^p > q$.*
- *Yes instances (A_{yes}): There exists an n-qubit state such that $\mathsf{Tr}[\rho \sum_i H_i] \leq 2^{-p}$*
- *No instances (A_{no}): For every n-qubit state ρ, the following holds: $\mathsf{Tr}[\rho \sum_i H_i] \geq \frac{1}{q}$*

BJSW Encoding. A key idea behind the construction from [BJSW16] is for the prover to encode its witness, $|\psi\rangle$, using a secret-key quantum authentication code (that also serves as an encryption) that satisfies the following key properties needed in the protocol. For any state $|\psi\rangle$, denote the encoding of $|\psi\rangle$ under the secret-key s by $\mathsf{E}_s(|\psi\rangle)$.

1. *Homomorphic evaluation of Cliffords.* Given $\mathsf{E}_s(|\psi\rangle)$, and given any Clifford circuit C, it is possible to compute $\mathsf{E}_{s'}(C|\psi\rangle)$ efficiently. Moreover, s' can be determined efficiently by knowing C and s.

2. *Homomorphic measurements of arbitrary Clifford basis.* For any Clifford circuit C and any state $|\psi\rangle$, a computational basis measurement on $C|\psi\rangle$ can be recovered from a computational basis measurement on $\mathsf{E}_{s'}(C|\psi\rangle)$ along with C and s. Formally, there is a classically efficiently computable function g such that if y is sampled from the distribution induced by measuring the state $\mathsf{E}_{s'}(C|\psi\rangle)$ in the computational basis, then $g(s, C, y)$ is sampled from the distribution induced by measuring the state $C|\psi\rangle$ in the computational basis.

3. *Authentication of measurement outcomes.* For any s and any clifford C, there is a set $\mathcal{S}_{s,C}$ such that for any state $|\psi\rangle$, and any computational basis measurement outcome y performed on $\mathsf{E}_{s'}(C|\psi\rangle)$, it holds that $y \in \mathcal{S}_{s,C}$. Furthermore, for any y, given s and C, it can be efficiently checked whether $y \in \mathcal{S}_{s,C}$.

4. *Simulatability of authenticated states*: there exists an efficient QPT algorithm B such that for any adversary $\mathcal{A}$, every $x \in A_{\mathsf{yes}}$ along with witness $|\psi\rangle$, poly(λ)-qubit advice ρ, the following holds: the probability that $\mathcal{P}(s, C^{\dagger}_{r^*}, \mathcal{A}(\mathsf{E}_s(|\psi\rangle)))$ outputs 1 is negligibly close to the probability that $\mathcal{P}(s, C^{\dagger}_{r^*}, \mathcal{A}(B(x, s, r^*)))$ outputs 1, where $\mathcal{P}$ is defined below.

$$\mathcal{P}(s, C^{\dagger}, y) = \begin{cases} 1 \text{ if } g(s, C^{\dagger}, y) \neq 0 \\ 0 \text{ otherwise} \end{cases}$$

In both the events, s and r^* are chosen uniformly at random.

The QMA verifier of the k-local Clifford Hamiltonian problem measures terms of the form $C|0^{\otimes k}\rangle\langle 0^{\otimes k}|C^{\dagger}$ where C is a Clifford circuit on a witness $|\psi\rangle$. Specifically, a verifier will first apply $C^{\dagger}$ and then measure in the computational basis. If the outcome of the measurement is the 0 string, it rejects. Otherwise, it accepts. In the zero-knowledge case, the witness will be encoded, $\mathsf{E}_s(|\psi\rangle)$, but the verifier can still compute $\mathsf{E}_s(C^{\dagger}|\psi\rangle)$ and measure to obtain some string y. Then, the prover can prove to the verifier (in NP) that y corresponds to a non-zero outcome on a measurement of $C^{\dagger}|\psi\rangle$ instead using the predicate $\mathcal{P}$.

We follow the approach of BJSW [BJSW16], except that we instantiate the coin-flipping protocol in a specific way in order to get concurrency when instantiating the underlying QZK for NP with our bounded concurrent construction.

Construction. We use the following ingredients in our construction:

- Statistical-binding and quantum-concealing commitment scheme, $(\mathsf{Comm}, \mathsf{R})$ (Section 2.2).
- Bounded concurrent QZK proof system, denoted by Π_{NP}, for the following language (Section 4).

$$\mathcal{L} = \left\{ \left((\mathbf{r}, \mathbf{c}, \mathbf{r}', \mathbf{c}', r^*, y, b)\ ;\ (s, \ell, a, \ell')\right)\ :\ \begin{array}{c} \mathcal{P}(s, C^{\dagger}_{r^*}, y)=1 \\ \bigwedge \\ \mathsf{Comm}(1^{\lambda}, \mathbf{r}, s; \ell)=\mathbf{c} \\ \bigwedge \\ \mathsf{Comm}(1^{\lambda}, \mathbf{r}', a; \ell')=\mathbf{c}' \\ \bigwedge \\ a \oplus b = r^* \end{array} \right\}$$

Let Q be the maximum number of sessions associated with the protocol.

We describe the construction of bounded concurrent QZK for QMA (with bound Q) in Figure 7.1. We prove the following.

Theorem 48. *Assuming that Π_{NP} satisfies the definition of bounded concurrent QZK for NP, the protocol given in Figure 7.1 is a bounded concurrent QZK protocol for* QMA *with soundness $\frac{1}{\text{poly}}$.*

Remark 49. *The soundness of the above protocol can be amplified by sequential repetition. In this case, the prover needs as many copies of the witness as the number of repetitions.*

Proof Sketch. Completeness follows from [BJSW16]. The only difference in our scheme and BJSW is that we combine the coin-flipping with the QZK for NP. This means that completeness follows from [BJSW16].

Instance: A k-local Clifford Hamiltonian, $H = \sum_{r=1}^{M} C_r|0^{\otimes k}\rangle\langle 0^{\otimes k}|C_r^\dagger$.

Witness: $|\psi\rangle$

- $P \leftrightarrow V$: Prover P samples a secret-key $s \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(k,M)}$, and commits to s using the commitment protocol $(\mathsf{Comm}, \mathsf{R})$. Let $\mathbf{r}$ be the first message of the receiver (sent by V) and $\mathbf{c}$ be the commitment.

  We call this the secret-key commitment.

- $P \to V$: P sends $\mathsf{E}_s(|\psi\rangle)$.

- $P \leftrightarrow V$: Prover samples a random string $a \xleftarrow{\$} \{0,1\}^{\log(M)}$, and commits to a using the commitment protocol $(\mathsf{Comm}, \mathsf{R})$. Let $\mathbf{r}'$ be the first message of the receiver and $\mathbf{c}'$ be the commitment.

  We call this the coin-flipping commitment.

- $V \to P$: Verifier samples a random string $b \xleftarrow{\$} \{0,1\}^{\log(M)}$. Verifier sends b to the prover.

- $P \to V$: Prover sends $r^* := a \oplus b$ to the verifier.

- Verifier computes $\mathsf{Eval}\left(C_{r^*}^\dagger, \mathsf{E}_s(|\Psi\rangle)\right) \to \mathsf{E}_s(C_{r^*}^\dagger|\psi\rangle)$ and measures in the computational basis. Let y denote the measurement outcome. Verifier sends y to the prover.

- Prover checks that $y \in \mathcal{S}_{s,C_{r^*}^\dagger}$ and that $\mathcal{P}(s, C_{r^*}^\dagger, y) = 1$. If not, it aborts.

- Prover and verifier engage in a QZK protocol for NP, Π_{NP}, for the statement $(\mathbf{r}, \mathbf{c}, \mathbf{r}', \mathbf{c}', r^*, y, b)$ and the witness (s, ℓ, a, ℓ').

Figure 5: Bounded-Concurrent QZK for QMA

Soundness. Once we argue that r^* produced in the protocol is uniformly distributed, even when the verifier is interacting with the malicious prover, we can then invoke the soundness of [BJSW16] to prove the soundness of our protocol.

Suppose the verifier accepts the Π_{NP} proof produced during the execution of the above protocol. From the soundness of Π_{NP}, we have that $r^* = a \oplus b$ where a is the string that the prover initially committed to in $\mathbf{c}'$. By the statistical binding security of the commitment, and the fact that b is chosen at random after a has been committed to, we have that r^* is sampled uniformly from $[M]$.

Bounded-Concurrent Quantum Zero-Knowledge. Suppose $x \in A_{\mathsf{yes}}$. Suppose V^* is a non-uniform malicious QPT Q-session verifier. Then we construct a QPT simulator Sim as follows.

Description of Sim: it starts with the registers $\mathbf{X}_{zk}, \mathbf{X}_{anc}, \mathbf{M}, \mathbf{Aux}$. The register $\mathbf{X}_{zk}$ is used by the simulator of the bounded concurrent QZK protocol, $\mathbf{X}_{anc}$ is an ancillary register, $\mathbf{M}$ is used to store the messages exchanged between the simulator and the verifier and finally, the register $\mathbf{Aux}$

is used for storing the private state of the verifier. Initialize the registers $\mathbf{X}_{zk}, \mathbf{M}$ with all zeroes. Initialize the register $\mathbf{X}_{anc}$ with $(\bigotimes_{j=1}^{Q} |s_j\rangle\langle s_j|) \otimes (\bigotimes_{j=1}^{Q} |r_j^*\rangle\langle r_j^*|) \otimes (\bigotimes_{j=1}^{Q} \rho_j) \otimes |0^{\otimes \mathrm{poly}}\rangle\langle 0^{\otimes \mathrm{poly}}|$, where s_i, r_i^* are generated uniformly at random and $\rho_j \leftarrow B(x, s_j, r_j^*)$ is defined in bullet 4 under BJSW encoding.

Sim applies the following unitary for Q times on the above registers. This unitary is defined as follows: it parses the message $((1, \mathsf{msg}_1), \ldots, (Q, \mathsf{msg}_Q))$ in the register $\mathbf{M}$. For every round of conversation, it does the following: if it is V^*'s turn to talk, it applies V^* on $\mathbf{Aux}$ and $\mathbf{M}$. Otherwise,

- Let S_1 be the set of indices such that for every $i \in S_1$, msg_i is a message in the protocol Π_{NP}. Finally, let $S_2 = [Q] \backslash S_1$.
- It copies $((1, \mathsf{msg}_1), \ldots, (Q, \mathsf{msg}_Q))$ into $\mathbf{X}_{zk}$ (using many CNOT operations) and for every $i \notin S_1$, replaces msg_i with N/A. We note that msg_i is a quantum state (for instance, it could be a superposition over different messages).
- For every $i \in S_2$, if msg_i is the first prover's message of the i^{th} session, then set msg'_i to be $|\mathbf{c}_i\rangle\langle\mathbf{c}_i| \otimes \rho_i$, where $\mathbf{c}_i$ is the secret-key commitment of 0. If msg_i corresponds to the coin-flipping commitment, then set msg'_i to be $|\mathbf{c}'_i\rangle\langle\mathbf{c}'_i|$ where $\mathbf{c}'_i$ is a commitment to 0.
- It applies the simulator of Π_{NP} on $\mathbf{X}_{zk}$ to obtain $((1, \mathsf{msg}'_{1,zk}), \ldots (Q, \mathsf{msg}'_{Q,zk}))$. The i^{th} session simulator of Π_{NP} takes as input $(\mathbf{r}_i, \mathbf{c}_i, \mathbf{r}'_i, \mathbf{c}'_i, r_i^*, y_i, b_i)$, where r_i^* was generated in the beginning and $\mathbf{r}_i, \mathbf{c}_i, \mathbf{r}'_i, \mathbf{c}'_i, y_i, b_i$ are generated as specified in the protocol.
- Determine $((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q))$ as follows. Set $\mathsf{msg}'_i = \mathsf{msg}_{i,\mathsf{zk}}$, if $i \in S_1$. Output of this round is $((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q))$.

We claim that the output distribution of Sim (ideal world) is computationally indistinguishable from the output distribution of V^* when interacting with the prover (real world).

$\underline{\mathsf{Hyb}_1}$: This corresponds to the real world.

$\underline{\mathsf{Hyb}_2}$: This is the same as Hyb_1 except that the verifier V^* is run in superposition and the transcript is measured at the end.

The output distributions of Hyb_1 and Hyb_2 are identical.

$\underline{\mathsf{Hyb}_3}$: Simulate the zero-knowledge protocol Π_{NP} simultaneously for all the sessions. Other than this, the rest of the hybrid is the same as before.

The output distributions of Hyb_2 and Hyb_3 are computationally indistinguishable from the bounded concurrent QZK property of Π_{NP}.

$\underline{\mathsf{Hyb}_{4.i} \text{ for } i \in [Q]}$: For every $j \leq i$, the coin-flipping commitment in the j^{th} session is a commitment to 0 instead of a_i. For all $j > i$, the commitment is computed as in the previous hybrid.

The output distributions of $\mathsf{Hyb}_{4.i-1}$ (or Hyb_3 if $i = 1$) and $\mathsf{Hyb}_{4.i}$ are computationally indistinguishable from the quantum concealing property of $(\mathsf{Comm}, \mathsf{R})$.

$\underline{\mathsf{Hyb}_{5.i} \text{ for } i \in [Q]}$: For every $j \leq i$, the secret-key commitment in the j^{th} session is a commitment to 0. For all $j > i$, the commitment is computed as in the previous hybrid.

The output distributions of $\mathsf{Hyb}_{5.i-1}$ (or $\mathsf{Hyb}_{4.Q}$ if $i = 1$) and $\mathsf{Hyb}_{5.i}$ are computationally indistinguishable from the quantum concealing property of $(\mathsf{Comm}, \mathsf{R})$.

$\underline{\mathsf{Hyb}_{6.i}}$ for $i \in [Q]$: For every $j \leq i$, the encoding of the state is computed instead using $B(x, s_i, r_i^*)$, where s_i, r_i^* is generated uniformly at random.

The output distributions of $\mathsf{Hyb}_{6.i-1}$ and $\mathsf{Hyb}_{6.i}$ are statistically indistinguishable from simulatability of authenticated states property of BJSW encoding (bullet 4). This follows from the following fact: conditioned on the prover not aborting, the output distributions of the two worlds are identical. Moreover, the property of simulatability of authenticated states shows that the probability of the prover aborting in the previous hybrid is negligibly close to the probability of the prover aborting in this hybrid.

$\underline{\mathsf{Hyb}_7}$: This corresponds to the ideal world.

The output distributions of $\mathsf{Hyb}_{6.Q}$ and Hyb_7 are identical.

□

Proof of Quantum Knowledge with better witness quality. We can define an anologous notion of proof of knowledge in the context of interactive protocols for QMA. This notion is called proof of *quantum* knowledge. See [CVZ20] for a definition of this notion. Coladangelo, Vidick and Zhang [CVZ20] show how to achieve quantum proof of quantum knowledge generically using quantum proof of classical knowledge. Their protocol builds upon [BJSW16] to achieve their goal. We can adopt their idea to achieve proof of quantum knowledge property for a bounded concurrent QZK for QMA system. In Figure 7.1, include a quantum proof of classical knowledge system for NP (for instance, the one we constructed in Section 6.1) just after the prover sends encoding of the witness state $|\Psi\rangle$, encoded using the key s. Using the quantum proof of classical knowledge system, the prover convinces the verifier of its knowledge of the s. The rest of the protocol is the same as Figure 7.1.

To see why this satisfies proof of quantum knowledge, note that an extractor can extract s with probability negligibly close to the acceptance probability and using s, can recover the witness $|\Psi\rangle$. For the first time, we get proof of quantum knowledge (even in the standalone setting) with $1 - \mathsf{negl}$ quality if the acceptance probability is negligibly close to 1, where the quality denotes the closeness to the witness state. Previous proof of quantum knowledge [BG19, CVZ20] achieved only $1 - \frac{1}{\mathrm{poly}}$ qualtiy; this is because these works use Unruh's quantum proof of classical knowledge technique [Unr12] and the extraction probability in Unruh is not negligibly close to the acceptance probability.

Acknowledgements

We thank Abhishek Jain for many enlightening discussions, Zhengzhong Jin for patiently answering questions regarding [GJJM20], Dakshita Khurana for suggestions on constructing oblivious transfer, Ran Canetti for giving an overview of existing classical concurrent ZK techniques, Aram Harrow and Takashi Yamakawa for discussions on the assumption of cloning security (included in a previous version of this paper) and Andrea Coladangelo for clarifications regarding [CVZ20].

References

[ABG+20] Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-quantum multi-party computation in constant rounds. *arXiv preprint arXiv:2005.12904*, 2020.

[ALP20] Prabhanjan Ananth and Rolando L La Placa. Secure quantum extraction protocols. In *TCC*, 2020.

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

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

[BG19] Anne Broadbent and Alex B Grilo. Zero-knowledge for qma from locally simulatable proofs. *arXiv preprint arXiv:1911.07782*, 2019.

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

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

[BS05] Boaz Barak and Amit Sahai. How to play almost any mental game over the net-concurrent composition via super-polynomial simulation. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 543–552. IEEE, 2005.

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In *STOC*, 2020.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In *Annual International Cryptology Conference*, pages 19–40. Springer, 2001.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 494–503, 2002.

[CLP15] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In *Annual Cryptology Conference*, pages 287–307. Springer, 2015.

[CVZ20] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive zero-knowledge arguments for qma, with preprocessing. In *Annual International Cryptology Conference*, pages 799–828. Springer, 2020.

[DCO99] Giovanni Di Crescenzo and Rafail Ostrovsky. On concurrent zero-knowledge with preprocessing. In *Annual International Cryptology Conference*, pages 485–502. Springer, 1999.

[DGH+20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from cdh or lpn. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 768–797. Springer, 2020.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *Journal of the ACM (JACM)*, 51(6):851–898, 2004.

[DS98] Cynthia Dwork and Amit Sahai. Concurrent zero-knowledge: Reducing the need for timing constraints. In *Annual International Cryptology Conference*, pages 442–457. Springer, 1998.

[FKP19] Cody Freitag, Ilan Komargodski, and Rafael Pass. Non-uniformly sound certificates with applications to concurrent zero-knowledge. In *Annual International Cryptology Conference*, pages 98–127. Springer, 2019.

[GJJM20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 668–699. Springer, 2020.

[GJO+13] Vipul Goyal, Abhishek Jain, Rafail Ostrovsky, Silas Richelson, and Ivan Visconti. Concurrent zero knowledge in the bounded player model. In *Theory of Cryptography Conference*, pages 60–79. Springer, 2013.

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

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC*, pages 291–304, 1985.

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

[JKMR06] Rahul Jain, Alexandra Kolla, Gatis Midrijanis, and Ben W Reichardt. On parallel composition of zero-knowledge proofs with black-box quantum simulators. *arXiv preprint quant-ph/0607211*, 2006.

[KKS18] Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 34–65. Springer, 2018.

[KSVV02] Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

[Lin03] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 683–692, 2003.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of cryptology*, 4(2):151–158, 1991.

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

[Pas04] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *STOC*, pages 232–241, 2004.

[PR03] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 404–413. IEEE, 2003.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *FOCS*, pages 366–375. IEEE, 2002.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for np from (plain) learning with errors. In *Annual International Cryptology Conference*, pages 89–114. Springer, 2019.

[PTV14] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent zero knowledge, revisited. *Journal of cryptology*, 27(1):45–66, 2014.

[PTW09] Rafael Pass, Wei-Lung Dustin Tseng, and Douglas Wikström. On the composition of public-coin zero-knowledge protocols. In *Annual International Cryptology Conference*, pages 160–176. Springer, 2009.

[PV08] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. On constant-round concurrent zero-knowledge. In *Theory of Cryptography Conference*, pages 553–570. Springer, 2008.

[Rab05] Michael O Rabin. How to exchange secrets with oblivious transfer. *IACR Cryptol. ePrint Arch.*, 2005(187), 2005.

[RK99] Ransom Richardson and Joe Kilian. On the concurrent composition of zero-knowledge proofs. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 415–431. Springer, 1999.

[Unr10] Dominique Unruh. Universally composable quantum multi-party computation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–505. Springer, 2010.

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

[VZ20] Thomas Vidick and Tina Zhang. Classical zero-knowledge arguments for quantum computations. *Quantum*, 4:266, 2020.

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