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 statistical receiver-private post-quantum oblivious transfer, there exists a bounded concurrent zero-knowledge proof system for NP satisfying quantum proof of knowledge property. We then instantiate the oblivious transfer protocol based on a new assumption, called learning with errors with cloning security.

  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 quantum verifiers; 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 would 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 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 resolve this problem conditionally. The main ingredient we use is an oblivious transfer protocol with statistical privacy for receivers and post-quantum computational privacy for senders.

[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 statistical receiver-private post-quantum oblivious transfer, there exists a 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.

Furthermore, we construct the oblivious transfer protocol based on a new assumption, called quantum hardness of learning with errors with cloning security. Roughly speaking, we require that learning with errors (LWE) holds even against adversaries that have access to cloned copies of their intermediate states. Ideally, we would like to not place any additional restriction on the adversary: for instance, the adversary can get access to cloned copies of their intermediate states at any point during its execution. However, it turns out that such an adversary is too powerful and can in fact break QLWE (such an attack was implicit in the works of [RZ20, KNY20]). However, we place certain restrictions on the behavior of the adversary that will allow us to make the security proof go through. We call this type of adversary, a reduction-friendly adversary; this is discussed in detail in Section 5.1. We conjecture that learning with errors holds against reduction-friendly adversaries. We leave open the problem of understanding the relationship between this assumption and QLWE or breaking our assumption.

Theorem 3 (Informal)**.** *Assuming QLWE with cloning security (Figure 4), there exists a statistical receiver-private post-quantum oblivious transfer protocol.*

Motivation behind the new assumption: Typically, to prove security of constructions based on LWE, we design a reduction that uses the existence of an attacker $\mathcal{A}$ to break LWE. But some reductions require many calls to $\mathcal{A}$ in order to break LWE. This presents a challenge if we were to port these proofs to the quantum setting. Due to the no-cloning theorem, if $\mathcal{A}$ is a QPT attacker, it might not be possible for the reduction to use $\mathcal{A}$ more than once. This is because $\mathcal{A}$ might be using an auxiliary quantum state ρ to break the primitive. In order to execute this attacker multiple times, we need to have multiple copies of ρ. To resolve this issue, we introduced the notion of cloning security.

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 4 (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 follows the framewok of [BJSW16] 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 also develop a bounded concurrent post-quantum coin-flipping protocol.

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 start with the overview of the construction for 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 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[5].

- **No-recording**: Before rewinding any round, the simulator needs to 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.

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

Once such a classical simulator is identified, we can then simulate quantum verifiers as follows: run the classical simulator and the quantum verifier[6] 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 – till 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 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 of slots, say τ, such that the value committed by the prover in this slot

[6] 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.

equals the bit sent by the verifier in the same slot. 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 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. 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 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 $120Q^7\lambda - 3Q^4\lambda$.

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.

Invoking WI. In the proof of bounded concurrent quantum zero-knowledge, we need to invoke quantum witness-indistinguishability property of the underlying system. Coming up with a reduction that uses the bounded concurrent adversary to break quantum witness-indistinguishability turns out to be tricky. To see why, lets say i be the session is such that we are invoking the witness-indistinguishability property of the i^{th} session. Now, the reduction, while running the bounded concurrent verifier in superposition, needs to determine which of the messages to forward to the external WI verifier. But since the verifier is being run in superposition, the reduction does not know which of the messages correspond to the i^{th} session messages. To circumvent this issue, we consider an alternate definition of quantum WI that allows the reduction to forward the messages even without knowing which of them correspond to the WI messages.

Quantum Proofs of Knowledge. Next, we focus on demonstrating a quantum zero-knowledge system satisfying quantum proof of knowledge property. This 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.

To satisfy this property, it suffices to design an extraction mechanism that allows us to extract a secret from the prover while ensuring that the verifier will be unable to learn this secret. We focus on designing such an extraction mechanism. Lets start with the simple case when the secret is a single bit.

Main Tool: Statistical Receiver-Private Oblivious Transfer. Our starting point is an oblivious transfer (OT) protocol [Rab05]. This protocol is defined between two entities: a sender and a receiver. The sender has two bits (m_0, m_1) and the receiver has a single bit b. At the end of the protocol, the receiver receives the bit m_b. The security against malicious senders (receiver privacy) states that the sender should not be able to distinguish (with non-negligible probability) whether the receiver's bit is 0 or 1. The security against malicious receivers (also called sender privacy) states that there is a bit b' such that the receiver cannot distinguish (with non-negligible probability) the case when the sender's input is (m_0, m_1) versus the setting when the sender's input is $(m_{b'}, m_{b'})$. We require receiver privacy to hold against unbounded senders while we require sender privacy needs to hold against quantum polynomial-time receivers. The reason we require receiver privacy against unbounded senders is because our goal is design extraction mechanism against computationally unbounded provers.

Construction with Post-Quantum Security. We show how to instantiate this statistical OT protocol by simplifying an existing construction of statistical OT by [GJJM20] and proving the security of the construction in the post-quantum setting.

We present an oversimplified version that conveys the essential ideas behind the construction. The starting point is a two-message oblivious transfer protocol with computational security against malicious receivers and unbounded security against malicious senders. Call this protocol $\Pi_{OT} = (OT_1, OT_2)$. Using this, we construct the desired statistical OT protocol as follows.

- The sender samples a random bit r. It takes the role of the receiver in the underlying two-message OT protocol. It then sends the first message of Π_{OT} with the receiver's message set

to be r.

- The receiver, on input choice bit β, samples another random bit r'. It takes the role of the sender in the underlying protocol Π_{OT}. It then sends the sender's message in Π_{OT}, where the sender's input in Π_{OT} is set to be $(r', r' \oplus \beta)$.
- Finally, 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 define formally the security notion. The malicious receiver R^*, on input state $|\Psi\rangle$, interacts with the sender and produces an auxiliary state $|\widetilde{\Psi}\rangle$ along with the second round message. Finally the third round message is computed using (m_0, m_1). We say that a QPT adversary $\mathcal{A}$, which gets as input $|\widetilde{\Psi}\rangle$, succeeds if it can guess each sender's message with probability significantly greater than $\frac{1}{2}$ (we only consider $m_0, m_1 \in \{0,1\}$). Let us first consider classical PPT adversaries. Suppose there exists a classical PPT receiver R^* and an adversary $\mathcal{A}$ who break the security notion then we use this receiver to violate the receiver privacy of Π_{OT}. We first turn $\mathcal{A}$ into a predictor that predicts m_0 with probability $\frac{1}{2} + \nu(\lambda)$ and predicts m_1 with probability $\frac{1}{2} + \nu(\lambda)$, where $\nu(\lambda)$ is a non-negligible function. We in turn use this to argue that there exists an efficient algorithm that can predict $\widetilde{r} \oplus r$ with probability $\frac{1}{2} + \nu(\lambda)$ and predicts $\widetilde{r}$ with probability $\frac{1}{2} + \nu(\lambda)$. The key point to note here is that the adversary does not necessarily simultaneously determine $\widetilde{r} \oplus r$ and $\widetilde{r}$ with probability $> \frac{1}{2} + \nu(\lambda)$. This means that we need to execute this adversary multiple times to recover the bits $\widetilde{r} \oplus r$ and $\widetilde{r}$. Once we recover both of these bits, we can then recover r. This then would violate the receiver privacy of Π_{OT}. In our actual protocol, we set the length of $r, r', \widetilde{r}$ to be λ and we use Goldreich-Levin decoding to recover $\widetilde{r} \oplus r$ and $\widetilde{r}$. From this, we can recover r.

An astute reader would already notice the issue in porting this approach to the case when R^* and $\mathcal{A}$ are QPT adversaries. In this case, we cannot hope to run $\mathcal{A}$ multiple times since it only gets a single copy of $|\widetilde{\Psi}\rangle$.

Fixing Issue Via Cloning Security. We fix this issue via our new assumption of LWE with cloning security. We first observe that assuming LWE with cloning security, there exists statistical sender-private two-message OT such that the receiver privacy holds against computational senders who are cloning adversaries. This follows from the work of [BD18]. Once we have such an OT scheme, then we can run $\mathcal{A}$ multiple times, and in each time, the input to $\mathcal{A}$ is $|\widetilde{\Psi}\rangle$.

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 reciever's role. The prover picks bits b and r uniformly at random and then sets the input to the sender to be (s, r) if $b = 0$, otherwise if $b = 1$, it sets the input to the receiver to be (r, s). The verifier sets the reciever'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 r.

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 the 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). In this case, the probability that the extractor succeeds is 0. Or the prover could set the OT sender's input to be (s, s) and in this case, success probability is 1. This means that the success probability depends on the state of the prover and thus we cannot apply Watrous rewinding.

We resolve this issue by additionally requiring the prover to prove to the verifier that there exists $b \in \{0, 1\}$ such that in the OT sender's message (m_0, m_1), $m_b = \perp$ and $m_{1-b} \neq \perp$. This is realized by using a quantum zero-knowledge protocol. We use a bounded concurrent QZK protocol we designed earlier; however, if one were to be only interested in the standalone setting, they can very well just use a standalone QZK protocol.

Thus, we have achieve an extraction mechanism that succeeds with probability negligibly close to 1. We are yet to calculate the probability with which a malicious QPT verifier can recover the secret bit.

Challenges in Proving QZK: We need to show that the verifier can recover the secret with small probability. A natural (but wrong) idea would be to design an inefficient extractor which can extract the verifier's bit, say $\widehat{b}$, and then set the OT sender's messages to be (s, r) if $\widehat{b} = 1$ and (r, s) if $\widehat{b} = 0$. However, such an extractor cannot exist. The reason being that the OT satisfies statistical binding property: this means that the receiver's transcript is consistent with the receiver's input bit being 0 and it is also consistent with the bit being 1. If such an extractor doesn't exist then it is unclear how we will be able to show that the receiver can get s with only small probability.

To overcome this issue, we give an alternate definition of statistical OT where the receiver's bit $\widetilde{b}$ is given as non-uniform advice[7] to the prover. Once this is fixed in the beginning, the prover can then set the OT sender's inputs appropriately to ensure that the receiver does not get the secret with high probability.

Error amplification. 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 bits $b_1, \ldots, b_\lambda$ uniformly at random. In the i^{th} execution, it sets the OT sender's input to be $(sh_i, \perp)$ if $b_i = 0$, otherwise it sets the OT sender's input to be $(\perp, sh_i)$. We can then argue that, with probability negligibly close to 1, the verifier cannot recover all the shares. If the verifier cannot recover all the shares, it cannot learn the bit s. This is argued again by non-uniformly fixing the verifier's bits at the beginning of the protocol. In order to fix the non-uniform advice, it is important to first sample the randomness of the prover along with

[7] It is non-trivial to define such a $\widetilde{b}$. We refer the reader to the technical sections to see how $\widetilde{b}$ is defined.

fixing the state of the verifier. Once this is done, the verifier's bits is defined to be an (inefficient) function of the prover's randomness and the verifier's initial state.

The extraction process for the new protocol proceeds by extracting all the shares, one at a time, and then reconstructing the secret bit s.

Extracting Multiple Bits. So far we only discussed how to achieve extraction of a single bit. To extract multiple bits, we sequentially repeat the above protocol, where each repetition is designed to extract a single bit.

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 5 (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.
- $\mathsf{View}_V\left(\langle P(y_1), V(y_2)\rangle\right)$ 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 $\mathsf{View}_P\left(\langle P(y_1), V(y_2)\rangle\right)$ 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 6 (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 7 (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 8. *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 9 (Watrous Rewinding Lemma)**.** *Suppose Q be a quantum circuit acting on $n+k$ qubits such that for every n-qubit state $|\psi\rangle$, the following holds:*

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

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

- $|p(\psi) - p_1| \leq \varepsilon$

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

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

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

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

2.4 Goldreich-Levin Theorem

We present Goldreich-Levin theorem [GL89] below. We state the version presented in [DGH+20].

Theorem 10 (Goldreich-Levin Theorem)**.** *There exists a PPT algorithm* GLDec *such that for any n, ν, any $y \in \{0,1\}^n$, and any function $\mathcal{P} : \{0,1\}^n \to \{0,1\}$ satisfying the following:*

$$\mathsf{Pr}_{u\xleftarrow{\$}\{0,1\}^n}[\langle u, y\rangle \leftarrow \mathcal{P}(u)] \geq \frac{1}{2} + \frac{1}{\nu},$$

we have:

$$\mathsf{Pr}\left[\mathsf{GLDec}\left(1^n, 1^\nu\right) = y\right] \geq \frac{1}{\mathrm{poly}(n,\nu)}$$

3 Concurrent Quantum ZK Proof Systems: Definitions

In Section 3.1, we define the notion of bounded concurrent QZK for NP.

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 11 (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 12. *In Section 5, 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 13 (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[8], 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 $\left((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q)\right)$. 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 14 (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.

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

Definition 15 (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:*

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

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 16 (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.*

- **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 17 (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 $\mathsf{msg}'_i = (\bot, \bot)$. 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 18. *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 19 (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 20 (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 fucntion negl.

Remark 21 (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 polynomially related to the acceptance probability whereas in our case, the extraction probability needs to be negligibly close to the acceptance probability.*

Definition 22 (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 23 (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.

Alternate Definition. We present another (albeit messy) definition of quantum witness-indistinguishability below. This definition will be useful for basing the security of concurrent ZK based on quantum witness-indistinguishable proof systems. We remark later why this definition is equivalent to the previous definition.

Definition 24 (Quantum Witness-Indistinguishability; Alternate Definition)**.** *A k-message 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, it holds that the output of $\mathsf{Expt}(1^\lambda, x, w_0, w_1, 0)$ is computationally indistinguishable from the output of $\mathsf{Expt}(1^\lambda, x, w_0, w_1, 1)$, where $\mathsf{Expt}(1^\lambda, x, w_0, w_1, b)$ is defined as follows:*

  $\underline{\mathsf{Expt}(1^\lambda, x, w_0, w_1, b)}$*:*

  - *V^* chooses $Q = \mathrm{poly}(\lambda, k) \geq k$.*
  - *It interacts with challenger in Q number of messages. We denote by $\mathbf{X}_i$ to be the i^{th} register containing the i^{th} message exchanged between the challenger and the verifier.*
  - *At the end of the i^{th} message, for $i \in [Q]$, let the state of the system be $|\Psi_i\rangle$. A subset of these registers will be only accessed by the challenger, a different subset of registers will be accessed only by the verifier and the remaining registers can be accessed by both the challenger and the verifier.*

    From this point onwards, the contents in the register $\mathbf{X}_i$ are never modified. The challenger performs $\mathsf{CNOT}^{\otimes \mathrm{poly}}$ and copies the message in $\mathbf{X}_i$ into its local register. It then

performs a unitary operation that implements the next message function of the prover (with (x, w_b) hardwired) on its local registers. It sets the output message in the register $\mathbf{X}_{i+1}$. Call the resulting state to be $|\Psi_{i+1}\rangle$.

- *At the end of the protocol, measure the registers $\mathbf{X}_1, \ldots, \mathbf{X}_Q$. Trace out the registers associated with the challenger.*
- *Output the measurement outcomes along the residual state.*

Remark 25. *Looking ahead, the reason why we need the above (messy) definition is the following: in the intermediate hybrids, we need to invoke the WI property for the i^{th} session, for some i, to prove that the protocol satisfies concurrent QZK. However, at this stage, the malicious concurrent verifier will be executed in superposition. This means that the reduction which interacts with the concurrent verifier needs to use the verifier's messages, being run in superposition, to break the WI property. However, the reduction cannot know whether the i^{th} session verifier has sent the WI messages or not if the messages are being executed in superposition. So in this case, the reduction simply forwards the i^{th} verifier's superposition of messages to the external prover (who uses one of two witnesses) who then computes on this state and returns the state back to the reduction.*

Let us see why the above definition is equivalent to Definition 23. To show this, we consider a variant of Definition 24 where the prover or the verifier, after computing the i^{th} message, measure the register $\mathbf{X}_i$. Note that Definition 24 and its variant are equivalent because the measurements and the unitaries applied by the challenger commute. The implication from the variant of Definition 24 to Definition 23 is easy. We focus on the other direction; showing that any protocol that satisfies Definition 23 also satisfies the above described variant.

Suppose a protocol satisfies Definition 23. We claim that the same protocol satisfies the above described variant of Definition 24. This follows from the fact that using any verifier V_1 that breaks this variant, we can construct another verifier V_2 which can break Definition 23. This outer verifier V_1 receives messages from the inner verifier V_2 and only forwards non-zero messages to the challenger. Note that if the inner verifier can break the variant then the outer verifier can break Definition 23.

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 23). 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.

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 $(w, \bot, \ldots, \bot)$.

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

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 26. *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 27 (Matched Slot)**.** *We say a slot is matched if the bit committed by P equals V's response.*

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.
- L denotes 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 two 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 WI proof.
- $\mathbf{W}$: 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, prepare the state[9]:

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

     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 $\mathbf{Ver}_t$.

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

- If the t^{th} message is a Stage 2 message from the prover to the t^{th} session of the verifier V^*, 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 i^{th} session. If so, compute the WI of Stage 2 using these matched slots. Otherwise, abort and output $\perp$ on register $\mathbf{Sim}_t$[10].

(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 registers $\{(\mathbf{R}_t, \mathbf{Sim}_t, \mathbf{Ver}_t)\}_{i\in B_j}$), and let $\mu(T)$ denote the set of all slots that are inside B_j in the transcript T. In superposition, perform the unitary U' defined below. Let I be a register containing a subset of qubits in $\mathbf{X}$. We omit the subscripts of the registers associated with the transcript T.

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

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

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

Let $W_j = \mathsf{Amplifier}\left(U_j^{V^*}\right)$; where $\mathsf{Amplifier}$ is the circuit guaranteed by Lemma 9. 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.

Remark 28. *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*

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

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 29. *For any* $(x, w) \in \mathcal{R}(\mathcal{L})$*, and for any auxiliary* $\mathrm{poly}(\lambda)$*-qubits state*[11] $|\Psi\rangle$*, 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 $U_L^{V^*} \dots U_1^{V^*} \left(I \otimes X^{\otimes_{j\in[Q]}\mathbf{B}_j}\right) |\Psi_{0,w}\rangle$, then, it measures the registers corresponding to the transcript (that is, $\{(\mathbf{Sim}_t, \mathbf{Ver}_t)\}_{t\in[\ell_{\mathsf{prot}}]]}$) and outputs the measurement outcome along with the resulting verifier's private state.

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

$\underline{\mathsf{Hyb}_{2.i}, \text{ for } i = 1 \text{ to } L}$: 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 $B_1, \dots, B_i$. 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:

$$U_L^{V^*} \dots U_{i+1}^{V^*} W_i \dots W_2 W_1 \left(I \otimes X^{\otimes_{j\in[Q]}\mathbf{B}_j}\right) |\Psi_{0,w}\rangle$$

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

We defer the proof of the following claim to Section 4.3.

Claim 30. *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* $\mathsf{Hyb}_{2.i+1}$.

$\underline{\mathsf{Hyb}_{3.i} \text{ for } i \in [Q]}$: 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:

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

This change means that in Stage 2 of the protocol to a session V_k^* for $k \leq i$, 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 4.4.

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

Claim 31. *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.*

$\underline{\mathsf{Hyb}_4}$: The output of this hybrid is the output of the simulator.

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

□

4.3 Proof of Claim 30

Let $|\Psi_{0,w}^{i-1}\rangle = W_{i-1} \dots W_1 |\Psi_{0,w}\rangle$. Without loss of generality, we can write $U_i^{V^*}|\Psi_{0,w}^{i-1}\rangle$ the following way:

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

where:

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

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

Claim 32. *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. We prove the following hybrid argument.

$\underline{\mathsf{Hyb}_1^*}$: This hybrid is defined to be identical to $\mathsf{Hyb}_{2.i}.\mathsf{Sim}^{V^*}$ except that it terminates the execution after the i^{th} block. After termination of execution in the i^{th} block, measure all the registers in $(\mathbf{Sim}_1, \mathbf{Ver}_1, \dots, \mathbf{Sim}_t, \mathbf{Ver}_t)$, where t is the last message in the block B_i. Output the resulting state of the system.

$\underline{\mathsf{Hyb}_2^*}$: In this hybrid, we define an algorithm that interacts with V^* until the i^{th} block. This algorithm behaves exactly like $\mathsf{Hyb}_{2.i}.\mathsf{Sim}^{V^*}$ except that it does not run V^* in superposition. Before sending the message in the t^{th} round, it computes the state in the register $\mathbf{Sim}_t$ as before. It then measures this register to obtain the t^{th} message which is then sent to V^*. Output the resulting state of the system.

The only difference between Hyb_1^* and Hyb_2^* is the fact that the in Hyb_1^*, the message registers are measured only at the very end whereas in Hyb_2^* are measured during the execution of the i^{th} block. Since the distribution of measurement outcomes will be the same in both the cases, we have

that the output distributions of hybrids Hyb_1^* and Hyb_2^* are identical.

$\underline{\mathsf{Hyb}_3^*}$: In the previous hybrid, note that the value in $\mathbf{Dec}$ contains $1 \oplus \mathsf{Match}(T, \mathbf{c}, b')$, where $\mathbf{c}$ is some commitment in the block B_i. Instead, perform uniform superposition over bits in an ancilliary register and store the value $1 \oplus v_{b'}$, where $v_{b'} = 1$ if $b' = b$, where b is the bit in the ancilliary register, otherwise set $v_{b'} = 0$.

In Hyb_3^*, note that the final state will be of the following form:

$$\sqrt{q}|\Phi_{i,\mathsf{noslot}}\rangle|+\rangle_{\mathbf{Dec}} + \sqrt{(1-q)}\left(\sqrt{\frac{1}{2}}|\Phi_{\mathsf{yes}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{\frac{1}{2}}|\Phi_{\mathsf{no}}\rangle|1\rangle_{\mathbf{Dec}}\right)$$

And the final state in Hyb_2^* is of the following form:

$$\sqrt{q}|\Phi_{i,\mathsf{noslot}}\rangle|+\rangle_{\mathbf{Dec}} + \sqrt{(1-q)}\left(\sqrt{p(\Phi_{i,\mathsf{slot}})}|\Phi_{\mathsf{yes}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p(\Phi_{i,\mathsf{slot}})}|\Phi_{\mathsf{no}}\rangle|1\rangle_{\mathbf{Dec}}\right)$$

We can show that $|p(\Phi_{i,\mathsf{slot}}) - \frac{1}{2}|$ is negligible. Suppose not, the quantum concealing property of Comm is violated; this follows from the fact that given a commitment to a bit b (picked uniformly at random), a QPT algorithm cannot come up with a bit b' such that $b' = b$ with probability $\frac{1}{2} \pm \delta(\lambda)$, for some non-negligible function $\delta(\cdot)$. This completes the proof of Claim 32.

□

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

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

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 9), $\mathsf{Amplifier}\left(U_i^{V^*}\right)$ 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 $|\Psi_{i,\mathsf{Good}}\rangle$.

Now, the following two distributions are computationally indistinguishable from the quantum-concealing property of commitment schemes:

- Measure the $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t\leq i}$ registers in the state $|\Psi_{i,\mathsf{Good}}\rangle$ and output the measurement outcome along with the state in the register $\mathbf{Aux}$.
- Measure the $\{\mathbf{Sim}_t, \mathbf{Ver}_t\}_{t\leq i}$ registers in the state $\sqrt{p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Good}}\rangle|0\rangle_{\mathbf{Dec}} + \sqrt{1-p(\Phi_{i,\mathsf{slot}})}|\Psi_{i,\mathsf{Bad}}\rangle|1\rangle_{\mathbf{Dec}}$ and output the measurement outcome along with the state in the register $\mathbf{Aux}$.

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

4.4 Proof of Claim 31

Before we prove Claim 31, we first give an auxiliary definition and some claims.

4.4.1 Auxiliary Definitions and Claims

Definition 33 (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 $6Q^7\lambda + Q^4\lambda$ matched commitments, with probability neglibly close to 1, for every verifier before starting Stage 2.

Claim 34. *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:

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

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 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 35. *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 34, 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 upper bounds the probability that the simulator does not have enough matched slots for some session to use in the WI.

Claim 36. *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 35. 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 $6Q^7\lambda + Q^4\lambda$ matched commitments is negligibly small in λ.

4.4.2 Finishing Proof of Claim 31

We use the auxiliary claims from the previous section to complete the proof. We prove this via the following hybrid argument.

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

$\underline{\mathsf{Hyb}^{(2)}_{3.i}}$: 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 36, we have that the probability that this hybrid aborts is negligible in λ. Conditioned on this hybrid not aborting, the output distributions of $\mathsf{Hyb}^{(1)}_{3.i}$ and $\mathsf{Hyb}^{(2)}_{3.i}$ are identical.

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

The hybrids $\mathsf{Hyb}^{(2)}_{3.i}$ and $\mathsf{Hyb}^{(3)}_{3.i}$ are computationally indistinguishable from the quantum witness indistinguishability property of Π_{WI}. Specifically, we rely upon Definition 24. Suppose there exists a concurrent verifier V^* along with a distinguisher that can distinguish hybrids $\mathsf{Hyb}^{(2)}_{3.i}$ and $\mathsf{Hyb}^{(3)}_{3.i}$ then we construct a reduction that breaks the quantum WI property in Definition 24. The reduction upon receiving a state from V^* first checks (in superposition) if the i^{th} session verifier's message is a WI message. If so, it copies it to an ancillary register which is initially set to have zeroes. If not, it keeps the value in this register intact. It forwards this register to the challenger in the quantum WI security experiment. The challenger returns back the register along with another register that contains it response. The reduction uses the response of the external challenger to respond to V^*. If hybrids $\mathsf{Hyb}^{(2)}_{3.i}$ and $\mathsf{Hyb}^{(3)}_{3.i}$ can be distinguished with probability ε then even the quantum WI property can be broken with the same probability.

5 Quantum Proofs of Knowledge

In this section, we construct a bounded concurrent QZK satisfying quantum proof of knowledge property.

We base our QZKPoK on a variant of QLWE, called QLWE with reduction-friendly cloning security. In the next subsection we introduce this assumption.

5.1 Reduction-friendly Cloning (RFC) Security

We introduce the notion of reduction-friendly cloning algorithms.

Definition 37 ($(\mathcal{R}, \mathcal{A})$-Reduction-friendly Cloning (RFC))**.** *We say that an algorithm, defined with respect to a tuple of QPT algorithms $\mathcal{A} = (\mathcal{A}_{pre}, \mathcal{A}_{onl}, \mathcal{A}_{post})$ and a QPT algorithm $\mathcal{R}$, is a reduction-friendly cloning algorithm, if it takes as input (x, r), quantum advice and executes in the following three phases:*

- *Pre-processing phase: On input x and non-uniform advice (pure state), it executes $\mathcal{A}_{pre}$ to obtain a pure state $|\Psi\rangle$.*
- *Online Phase: It executes $\mathcal{R}^{\mathcal{A}_{onl}(|\Psi\rangle)}$ on input r. For every query made by $\mathcal{R}$ to the oracle, $\mathcal{A}_{onl}$ is executed on input this query and a fresh copy of the state $|\Psi\rangle$. Denote $|\Phi\rangle$ to be the output of this online phase.*

- *Post-processing phase: On input $|\Phi\rangle$, execute $\mathcal{A}_{post}$ to obtain a bit b. Output b.*

Remark 38. *We can allow either classical or superposition access to the oracle. In this work, we restrict the algorithm $\mathcal{R}$ to make only classical queries.*

The above definition is restrictive and tailored to our security reduction. We consider a more natural definition called general cloning, in Appendix A.

QLWE with $\mathcal{R}$-Reduction-friendly Security. We revisit the learning with errors problem and analyze its security against reduction-friendly algorithms. We define the assumption of QLWE with reduction-friendly security in Figure 2.

QLWE with $\mathcal{R}$-reduction-friendly cloning security

Formally, QLWE with cloning security, parameterized by a field $\mathbf{q}$, integers n, m, noise distribution χ (typically, this is defined to be a discrete Gaussian distribution), QPT algorithm $\mathcal{R}$, is defined as follows: for any QPT algorithm $\mathcal{A}$, the $(\mathcal{R}, \mathcal{A})$-reduction-friendly cloning distinguisher can distinguish the following two distributions only with probability negligible in n:

- $\{\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}^T\}$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_{\mathbf{q}}^{n\times m}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_{\mathbf{q}}^{n}, \mathbf{e} \leftarrow \chi^m$, and,
- $\{\mathbf{A}, \mathbf{u}^T\}$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_{\mathbf{q}}^{n\times m}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_{\mathbf{q}}^{m}$.

We set $n = \text{poly}(\lambda)$, $m = \text{poly}(n\log(\mathbf{q}))$, $\mathbf{q} < 2^{\text{poly}(n)}$ and χ is a discrete Gaussian distribution of parameter $\alpha\mathbf{q} \geq 2\sqrt{n}$ with $0 < \alpha < 1$.

Figure 2:

First, we observe that this assumption is false for arbitrary $\mathcal{R}$. This attack is implicit in the work of [RZ20, KNY20].

- *Pre-processing phase*: upon receiving $(\mathbf{A}, \mathbf{b}^T)$, first prepare a uniform superposition over all the "short" vectors. That is, prepare the state $\sum_{x\in\{0,1\}^m} \frac{1}{\sqrt{2^m}}|x\rangle|0\rangle$. Then compute the unitary U, defined as follows: $U|x\rangle|0\rangle = |x\rangle|\mathbf{A}x\rangle$. Measure the second register to obtain y. Now, the state collapses to $\sum_{x\in\{0,1\}^n:\mathbf{A}x=y} \alpha_x|x\rangle$. Output $|\Psi\rangle = \sum_{x\in\{0,1\}^n:\mathbf{A}x=y} \alpha_x|x\rangle$.
- *Online phase*: every time, $\mathcal{A}_{onl}(|\Psi\rangle)$ is invoked, it measures the state to obtain x and outputs x. $\mathcal{R}$ runs $\mathcal{A}_{onl}$ twice to obtain x and x'. It outputs $x - x'$.
- *Post-processing phase*: If $A(x - x') \neq 0$ or if $x = x'$, then abort. Otherwise, check if $||\langle \mathbf{b}, (x - x')\rangle||_\infty \leq \frac{\mathbf{q}}{4}$ and if so, output 0. Otherwise, output 1.

If we set m to be sufficiently large then we can argue that with high probability, we will find x and x' such that $A(x - x') = 0$ and $x \neq x'$. Conditioned on this event, we can distinguish an LWE sample versus uniform with non-negligible probability. If $\mathbf{b}^T$ was an LWE sample then $||\langle \mathbf{b}, (x - x')\rangle||_\infty \leq \frac{\mathbf{q}}{4}$, otherwise with non-negligible probability, $||\langle \mathbf{b}, (x - x')\rangle||_\infty > \frac{\mathbf{q}}{4}$.

Assumption. We consider a specific reduction, denoted by $\mathcal{R}_{\mathsf{GL}}$, and we consider the assumption of QLWE secure against $\mathcal{R}_{\mathsf{GL}}$-reduction-friendly adversaries. This is the assumption that we rely upon for our construction of quantum proof of knowledge system. We first describe $\mathcal{R}_{\mathsf{GL}}$ in Figure 3. This algorithm has access to $\mathcal{A}$ which gets as input $|\Psi\rangle$ in every invocation.

$\underline{\mathcal{R}^{\mathcal{A}}_{\mathsf{GL}}(r)}$:

- $v_1 \leftarrow \mathsf{GLDec}^{\mathcal{A}(|\Psi\rangle)}$, where GLDec is as defined in Theorem 10 and every query to the oracle is appended with 1.
- $v_2 \leftarrow \mathsf{GLDec}^{\mathcal{A}(|\Psi\rangle)}$, where every query to the oracle is appended with 2.
- If $v_1 \oplus v_2 = r$, output 1. Otherwise, output 0.

Figure 3: Reduction $\mathcal{R}_{\mathsf{GL}}$

We state the assumption in Figure 4.

Assumption: $(\mathbf{q}, n, m, \chi)$-QLWE with $\mathcal{R}_{\mathsf{GL}}$-security holds, where:

- $\mathcal{R}_{\mathsf{GL}}$ is described in Figure 3,
- $n = \mathrm{poly}(\lambda)$,
- $m = \mathrm{poly}(n \log(\mathbf{q}))$
- $\mathbf{q} < 2^{\mathrm{poly}(n)}$ and,
- χ is a discrete Gaussian distribution of parameter $\alpha\mathbf{q} \geq 2\sqrt{n}$ with $0 < \alpha < 1$.

Figure 4:

5.2 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. Later, we will work with a different, albeit equivalent, definition. We consider three-round protocols for simplicity.

Definition 39 (Post-Quantum Statistical Receiver-Private Oblivious Transfer)**.** *A three-round oblivious transfer is a tuple of algorithms* $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$ *which specifies the following protocol.*

Round 1. *The sender* S *computes* $(\mathsf{ot}_1, \mathsf{st}_S) \leftarrow \mathsf{OT}_1(1^\lambda)$ *and sends* OT_1 *to the receiver* R.

Round 2. *The receiver* R *with input* $\beta \in \{0,1\}$*, computes* $(\mathsf{ot}_2, \mathsf{st}_R) \leftarrow \mathsf{ot}_2(1^\lambda, \mathsf{ot}_1, \beta)$*. Send* ot_2 *to* S*.*

Round 3. S *with input* $(m_0, m_1) \in \{0,1\}^2$ *computes* $\mathsf{ot}_3 \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, \mathsf{st}_S, m_0, m_1)$*. Send* ot_3 *to* R*.*

Reconstruction. *The receiver computes* $m' \leftarrow \mathsf{OT}_4(1^\lambda, \mathsf{ot}_1, \mathsf{ot}_2, \mathsf{ot}_3, \mathsf{st}_R)$*. Output* m'*.*

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

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

Statistical Receiver-Privacy. *For any (potentially maliciously generated)* OT_1^**, denote* $\left(\mathsf{ot}_2^{(0)}, \mathsf{st}_R^{(0)}\right) \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{OT}_1^*, 0)$ *and* $(\mathsf{ot}_2^{(1)}, \mathsf{st}_R^{(1)}) \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{OT}_1^*, 1)$*. Then the statistical distance between the marginal distributons* $\{\mathsf{ot}_2^{(0)}\}$ *and* $\{\mathsf{ot}_2^{(1)}\}$ *is a negligible function in* λ*.*

Post-Quantum Sender-Privacy (Definition #1). *For any non-uniform QPT distinguisher* $\mathcal{A}$ *and any malicious receiver* R^* *modeled as a unitary, 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 for the first round and the second round with* R^**. Specifically, the challenger executes* $(\mathsf{ot}_1, \mathsf{st}_S) \leftarrow \mathsf{OT}_1(1^\lambda)$*. Then send* ot_1 *to* R^**. Then the receiver* R^* *outputs a state in two registers* $\mathbf{A}$ *and* $\mathbf{B}$*. The register* $\mathbf{A}$ *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* $\mathsf{OT}_3(1^\lambda, \cdot, \mathsf{st}_S, m_{b_0}, m_1)$ *on* $\mathbf{A}$*. The resulting state 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* $\mathsf{OT}_3(1^\lambda, \cdot, \mathsf{st}_S, m_0, m_{b_1})$ *on* $\mathbf{A}$*. The resulting state 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: suppose $\mathcal{H}_\mathsf{S}$ *be the space of randomness of the sender.*

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

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 40. *Note that the requirement that* R^* *is a unitary is without loss of generality. Firstly, we can think of the input state given to* R^* *as being purified. Secondly, any measurements performed by* R^* *can be deferred to the distinguisher* $\mathcal{A}$*. The reason why we consider* R^* *to be unitary is because if it did perform measurements then in the sender-privacy experiment, the advantage need to be defined over the measurement outcomes of the receiver and not just over the randomness of the sender.*

Remark 41. *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.*

Alternate Formulation of Sender-Privacy. It helps to look at a different, but equivalent, formulation of post-quantum sender-privacy notion. While this definition is less intuitive than the previous one, it will be easier to use this formulation for our application.

Definition 42 (Post-Quantum Sender Privacy; Definition #2)**.** *Consider a three-round oblivious transfer protocol* $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$*.*

Let $\mathcal{A}$ *be a non-uniform QPT distinguisher and* R^* *be a malicious non-uniform QPT receiver. Without loss of generality, we assume that* R^* *is a unitary.*

Consider the following experiment.

Game $\widehat{G}(r, |\Phi\rangle, m_0, m_1, b)$*:*

- *The challenger computes* $(\mathsf{ot}_1, \mathsf{st}_S) \leftarrow \mathsf{OT}_1(1^\lambda; r)$*. It sends* ot_1 *to* R^**.*
- *The receiver* R^**, on input* $|\Phi\rangle$*, outputs* $(\mathsf{ot}_2^*, |\Psi\rangle)$*. Send* ot_2^* *to the challenger and send the secret state* $|\Psi\rangle$ *to* $\mathcal{A}$*.*
- *The challenger picks a bit* $\widehat{b}$ *uniformly at random. If* $b = 0$*, the challenger sends* $\mathsf{ot}_3^* \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_{\widehat{b}}, m_1)$*. If* $b = 1$*, the challenger sends* $\mathsf{ot}_3^* \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_0, m_{\widehat{b}})$*.*
- *Finally,* $\mathcal{A}$ *outputs two bits* (b'_0, b'_1)*. If* $b = 0$ *and* $b'_0 = \widehat{b}$*, then we say that* $\mathcal{A}$ *wins. If* $b = 1$ *and* $b'_1 = \widehat{b}$*, then we say that* $\mathcal{A}$ *wins.*

We say that $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$ *satisfies post-quantum sender privacy property (Definition #2) if the following holds: for every state* $|\phi\rangle$ *and for every* $(m_0, m_1) \in \{0,1\}^2$*, there exists a negligible function* $\nu(\cdot)$ *such that,*

$$\mathsf{Pr}\left[\min_{\leq \nu(\lambda)}\left|\left\{\mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}(r,|\Phi\rangle,m_0,m_1,0)\right]-\tfrac{1}{2}\right|,\left|\mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}(r,|\Phi\rangle,m_0,m_1,1)\right]-\tfrac{1}{2}\right|\right\}\right| : r \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}\right] \geq 1-\nu(\lambda),$$

We claim that the alternate formulation stated above is equivalent to post-quantum computational sender-privacy security.

Lemma 43. *Definitions #1 and #2 of post-quantum sender security are equivalent.*

Proof. Suppose an OT protocol $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$ satisfies Definition #1. Then, we have $\mathbb{E}_{\mathcal{H}_\mathsf{S}}[\min\{p_0, p_1\}]$ to be a negligible function $\nu(\lambda)$, where $p_0, p_1, \mathcal{H}_\mathsf{S}$ is as defined in Definition 39. Let $S \subseteq \mathcal{H}_\mathsf{S}$ be a set such that for every $r \in S$, the following holds:

$$\min\left\{\left|\mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}(r, |\phi\rangle, m_0, m_1, 0)\right] - \frac{1}{2}\right|, \left|\mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}(r, |\phi\rangle, m_0, m_1, 1)\right] - \frac{1}{2}\right|\right\} \leq \sqrt{\nu(\lambda)}$$

If we show that $\frac{|S|}{|\mathcal{H}_\mathsf{S}|} \geq 1 - \sqrt{\nu(\lambda)}$ then this would show that the OT protocol also satisfies Definition #2. Note that $\mathbb{E}_{\mathcal{H}_\mathsf{S}}[\min\{p_0, p_1\}] > \frac{|\mathcal{H}_\mathsf{S}| - |S|}{|\mathcal{H}_\mathsf{S}|} \cdot \sqrt{\nu(\lambda)}$. Suppose $\frac{|S|}{|\mathcal{H}_\mathsf{S}|} < 1 - \sqrt{\nu(\lambda)}$ then $\frac{|\mathcal{H}_\mathsf{S}| - |S|}{|\mathcal{H}_\mathsf{S}|} \geq \sqrt{\nu(\lambda)}$. But this would mean that $\mathbb{E}_{\mathcal{H}_\mathsf{S}}[\min\{p_0, p_1\}] > \nu(\lambda)$, contradicting the hypothesis. Thus, $\frac{|S|}{|\mathcal{H}_\mathsf{S}|} \geq 1 - \sqrt{\nu(\lambda)}$.

Suppose an OT protocol $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$ satisfies Definition #2. This means that there exists a set $S \subseteq \mathcal{H}_\mathsf{S}$ such that $\frac{|S|}{|\mathcal{H}_\mathsf{S}|} \geq 1 - \nu(\lambda)$ and the following holds for every $r \in S$:

$$\min\left\{\left|\mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}(r, |\phi\rangle, m_0, m_1, 0)\right] - \frac{1}{2}\right|, \left|\mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}(r, |\phi\rangle, m_0, m_1, 1)\right] - \frac{1}{2}\right|\right\} \leq \nu(\lambda),$$

for negligible $\nu(\lambda)$. Now, we have $\mathbb{E}_{\mathcal{H}_\mathsf{S}}[\min\{p_0, p_1\}] \leq \sum_{r \in S} \frac{1}{|\mathcal{H}_\mathsf{S}|} \cdot \nu(\lambda) + \frac{|\mathcal{H}_\mathsf{S}| - |S|}{|\mathcal{H}_\mathsf{S}|} \leq 2\nu(\lambda)$. This shows that OT satisfies Definition #1. □

5.2.1 Instantiation of Post-Quantum Statistical Oblivious Transfer

In this section, we show how to instantiate a post-quantum statistical receiver-private oblivious transfer protocol. We adapt the construction of Goyal et al. [GJJM20].

Main Ingredient. 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.

Definition 44 (Post-Quantum Statistical Sender-Private OT)**.** *A two-round oblivious transfer is a tuple of algorithms* $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ *which specifies the following protocol.*

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

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

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

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

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

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.*

Reduction-friendly Cloning Security. We define a variant of post-quantum statistical sender-private OT below.

Definition 45 (Reduction-friendly Cloning Security)**.** *We say that a post-quantum statistical sender-private oblivious transfer satisfies* $\mathcal{R}$*-reduction-friendly cloning security if the following holds: for any QPT adversary* $\mathcal{A}$*, a* $\mathcal{R}$*-reduction-friendly adversary can distinguish the distributions* $\{\mathsf{OT}_1(1^\lambda, 0)\}$ *and* $\{\mathsf{OT}_1(1^\lambda, 1)\}$ *with only negligible probability.*

We observe that the construction of [BD18] satisfies the above definition assuming QLWE with $\mathcal{R}_{\mathsf{GL}}$-RFC security.

Corollary 46 ([BD18])**.** *Assuming QLWE with* $\mathcal{R}_{\mathsf{GL}}$*-RFC security, there exists a two-round post-quantum statistical sender-private oblivious transfer protocol with* $\mathcal{R}_{\mathsf{GL}}$*-RFC security.*

In proof of computational receiver privacy in [BD18] (Lemma 5.2), the reduction that breaks LWE feeds in the input sample into the adversary and the output of the adversary is set to be the output of the reduction. Thus, if the adversary was a $\mathcal{R}_{\mathsf{GL}}$-reduction-friendly cloning algorithm then so is the reduction.

Construction of Statistical Receiver-Private OT. We present the construction $\Pi_{\mathsf{OT}} = (\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$ below.

Denote the sender by S and receiver to be R. Denote the two-round *sender-private* oblivious protocol to be (OT_1, OT_2, OT_3). In the construction below, the sender will take the role of the receiver of (OT_1, OT_2, OT_3), and the receiver the role of the sender.

- $\underline{\mathsf{OT}_1(1^\lambda)}$: sample bits $r_i \xleftarrow{\$} \{0,1\}$, for $i \in [\lambda]$. Compute $\left(ot_1^{(i)}, st_R^{(i)}\right) \leftarrow OT_1\left(1^\lambda, r_i\right)$, for $i \in [\lambda]$. Output $\mathsf{ot}_1 = \left(ot_1^{(1)}, \ldots, ot_1^{(\lambda)}\right)$. Set r to be $r_1 \cdots r_\lambda$. Let the private state be $\mathsf{st}_S = \left(st_R^{(1)}, \ldots, st_R^{(\lambda)}\right)$.

- $\underline{\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, \beta)}$: sample bits $r'_i \xleftarrow{\$} \{0,1\}$, for $i \in [\ell]$. Compute $ot_2^{(i)} \leftarrow OT_2(1^\lambda, ot_1^{(i)}, (r'_i, r'_i \oplus \beta))$. Output $\mathsf{ot}_2 = \left(ot_2^{(1)}, \ldots, ot_2^{(\lambda)}\right)$. Set r' to be $r'_1 \cdots r'_\lambda$. Set $\mathsf{st}_R = r'$.

- $\underline{\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, \mathsf{st}_S, (m_0, m_1))}$: parse $\mathsf{st}_S = \left(st_R^{(\lambda)}, \ldots, st_R^{(\lambda)}\right)$. Compute $\widetilde{r}_i \leftarrow OT_3(1^\lambda, \mathsf{ot}_1^{(i)}, \mathsf{ot}_2^{(i)}, st_R^{(i)})$, for $i \in [\lambda]$. Set $\widetilde{r}$ to be $\widetilde{r}_1 \cdots \widetilde{r}_\lambda$. Sample $u_0 \xleftarrow{\$} \{0,1\}^\lambda$ and $u_1 \xleftarrow{\$} \{0,1\}^\lambda$. Output the following: $\mathsf{ot}_3 = (\langle \widetilde{r}, u_0 \rangle \oplus m_0, \langle \widetilde{r} \oplus r, u_1 \rangle \oplus m_1, u_0, u_1)$.

- **Reconstruction**, $\underline{\mathsf{OT}_4(1^\lambda, \mathsf{ot}_1, \mathsf{ot}_2, \mathsf{ot}_3, \mathsf{st}_R)}$: parse ot_3 as $(\mathsf{msg}_0, \mathsf{msg}_1, u_0, u_1)$. Output $\langle r', u_\beta \rangle \oplus \widetilde{m_\beta}$.

Correctness. Let $\beta \in \{0,1\}, m_0, m_1 \in \{0,1\}$. Let $\mathsf{ot}_1, \mathsf{ot}_2, \mathsf{ot}_3$ be computed as specified in the protocol. By the correctness of (OT_1, OT_2, OT_3), we have the following: if $\beta = 0$, then $\mathsf{msg}_0 = \langle r', u_0\rangle \oplus m_0$. The reconstruction algorithm correctly outputs m_0. If $\beta = 1$, then $\mathsf{msg}_1 = \langle r', u_1\rangle \oplus m_1$. The reconstruction algorithm correctly outputs m_1.

Statistical Receiver Privacy. To argue this, we invoke the statistical sender privacy property of (OT_1, OT_2, OT_3). There exists an unbounded extractor that can extract r from *any* first round message. We consider two cases for every bit of r. For every $i \in [\lambda]$, if $r_i = 0$, then the sender privacy property of (OT_1, OT_2, OT_3) states that the statistical distance between the distributions $\{\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, (r'_i, r'_i \oplus \beta))\}$ and $\{\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, (r'_i, r'_i))\}$ is negligible. This means that β is information-theoretically hidden from the sender. If $r_i = 1$, then the sender privacy property of (OT_1, OT_2, OT_3) states that the statistical distance between the distributions $\{\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, (r'_i, r'_i \oplus \beta))\}$ and $\{\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, (r'_i \oplus \beta, r'_i \oplus \beta))\}$ is negligible. Thus, even in this case, β is information-theoretically hidden from the sender.

Post-Quantum Sender Privacy with $\mathcal{R}$-RFC security. Our proof template closely follows the proof technique of [DGH+20, GJJM20]. Suppose there exists a QPT malicious receiver R^*, auxiliary state $|\Psi\rangle$ and a distinguisher $\mathcal{A}$ that violates the post-quantum sender security definition (Definition 42) with probability $\nu(\lambda)$. That is, there exists m_0, m_1 and a an inverse polynomial function $\nu(\lambda)$ such that:

$$\mathsf{Pr}\left[\begin{array}{c}\left|\mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}(\mathbf{r}, |\Phi\rangle, m_0, m_1, 0)\right] - \frac{1}{2}\right| \geq \nu(\lambda) \\ \wedge \\ \left|\mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}(\mathbf{r}, |\Phi\rangle, m_0, m_1, 1)\right] - \frac{1}{2}\right| \geq \nu(\lambda)\end{array} : \mathbf{r} \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}\right] \geq \nu(\lambda),$$

We can rephrase the above inequality as follows.

$$\mathsf{Pr}\left[\begin{array}{c}\left|\mathsf{Pr}\left[(b_0,b_1) \leftarrow \mathcal{A}(|\Psi\rangle, m_0, m_1, \langle \widetilde{r}, u_0\rangle \oplus m_b, \langle \widetilde{r} \oplus r, u_1\rangle \oplus m_1, u_0, u_1) \ \wedge \ b_0 = b \colon b \xleftarrow{\$} \{0,1\}\right] - \frac{1}{2}\right| \geq \nu(\lambda) \\ \wedge \\ \left|\mathsf{Pr}\left[(b_0,b_1) \leftarrow \mathcal{A}(|\Psi\rangle, m_0, m_1, \langle \widetilde{r}, u_0\rangle \oplus m_0, \langle \widetilde{r} \oplus r, u_1\rangle \oplus m_b, u_0, u_1) \ \wedge \ b_1 = b \colon b \xleftarrow{\$} \{0,1\}\right] - \frac{1}{2}\right| \geq \nu(\lambda)\end{array}\right] \geq \nu(\lambda),$$

The outer probability is computed over the randomness of the sender. The inner probabilities are computed over u_0, u_1 and over the measurement outcomes of $\mathcal{A}$.

We now observe the following: (i) the distributions $\{\langle \widetilde{r}, u_0\rangle \oplus m_0\}$ and $\{\langle \widetilde{r}, u_0\rangle \oplus m_1\}$ can be distinguished with probability $\nu(\lambda)$ and, (ii) the distributions $\{\langle \widetilde{r} \oplus r, u_1\rangle \oplus m_0\}$ and $\{\langle \widetilde{r} \oplus r, u_1\rangle \oplus m_1\}$ can be distinguished with probability $\nu(\lambda)$. From this, it follows that (i) the distribution on $\{\langle \widetilde{r}, u_0\rangle\}$ and the uniform distribution on $\{0,1\}$ can be distinguished with probability $\frac{\nu(\lambda)}{2}$ and, (ii) the distribution $\{\langle \widetilde{r} \oplus r, u_1\rangle\}$ and the uniform distribution on $\{0,1\}$ can be distinguished with probability $\frac{\nu(\lambda)}{2}$.

From this, it follows that there exists two 1-bit QPT predictors $\mathcal{A}'_1$ and $\mathcal{A}'_2$ such that the following holds:

$$\mathsf{Pr}\left[\begin{array}{c}\mathsf{Pr}\left[(b_0,b_1) \leftarrow \mathcal{A}'_1(|\Psi\rangle, m_0, m_1, \langle \widetilde{r} \oplus r, u_1\rangle \oplus m_1, u_0, u_1) \ \wedge \ b_0 = \langle \widetilde{r}, u_0\rangle\right] \geq \frac{1}{2} + \frac{\nu(\lambda)}{2} \\ \wedge \\ \mathsf{Pr}\left[(b_0,b_1) \leftarrow \mathcal{A}'_2(|\Psi\rangle, m_0, m_1, \langle \widetilde{r}, u_0\rangle \oplus m_0, u_0, u_1) \ \wedge \ b_1 = \langle \widetilde{r} \oplus r, u_1\rangle\right] \geq \frac{1}{2} + \frac{\nu(\lambda)}{2}\end{array}\right] \geq \nu(\lambda)$$

The outer probability is computed over the randomness of the sender. Thus, there exists a set Good such that for every sender's randomness $\mathbf{r} \in \mathsf{Good}$, the following holds:

- $\mathsf{Pr}\left[(b_0,b_1) \leftarrow \mathcal{A}_1'\left(|\Psi\rangle, m_0, m_1, \langle \widetilde{r} \oplus r, u_1\rangle \oplus m_1, u_0, u_1\right) \ \wedge \ b_0 = \langle \widetilde{r}, u_0\rangle\right] \geq \frac{1}{2} + \frac{\nu(\lambda)}{2}$ and,
- $\mathsf{Pr}\left[(b_0,b_1) \leftarrow \mathcal{A}_2'\left(|\Psi\rangle, m_0, m_1, \langle \widetilde{r}, u_0\rangle \oplus m_0, u_0, u_1\right) \ \wedge \ b_1 = \langle \widetilde{r} \oplus r, u_1\rangle\right] \geq \frac{1}{2} + \frac{\nu(\lambda)}{2}$

By an averaging argument, there exist two sets Good_0' and Good_1' defined as follows. Good_0' consists of tuples of the form $v_0 = (\mathbf{r}, u_1)$ such that upon fixing v_0, over the randomness of $\mathcal{A}_1'$, the event in bullet 1 is at least $\frac{1}{2} + \frac{\nu(\lambda)}{2}$. Similarly, we define Good_1'.

By an averaging argument, the following holds: for every $\mathbf{r} \in \mathsf{Good}$,

- $\mathsf{Pr}\left[(\mathbf{r}, u_1) \in \mathsf{Good}_0'\right] \geq \frac{1}{2} + \frac{\nu(\lambda)}{4}$, where the randomness is computed over u_1.
- $\mathsf{Pr}\left[(\mathbf{r}, u_0) \in \mathsf{Good}_1'\right] \geq \frac{1}{2} + \frac{\nu(\lambda)}{4}$, where the randomness is computed over u_0.

We can now apply Goldreich-Levin Theorem (Theorem 10) to obtain the following.

- For every fixing of the values $(\mathbf{r}, u_1) \in \mathsf{Good}_1'$, $\mathsf{Pr}\left[\widetilde{r} \leftarrow \mathsf{GLDec}^{\mathcal{A}_1'(|\Psi\rangle, m_0, m_1, \langle \widetilde{r} \oplus r, u_1\rangle, \cdot, u_1)}\right] \geq \nu'(\lambda)$, for some polynomial $\nu'(\lambda)$. Every time, the oracle is invoked, $\mathcal{A}_1'$ is executed on a fresh copy of $|\Psi\rangle$.
- For every fixing of the values $(\mathbf{r}, u_0) \in \mathsf{Good}_1'$, $\mathsf{Pr}\left[\widetilde{r} \oplus r \leftarrow \mathsf{GLDec}^{\mathcal{A}_2'(|\Psi\rangle, m_0, m_1, \langle \widetilde{r}, u_0\rangle, u_0, \cdot)}\right] \geq \nu'(\lambda)$. As above, every time the oracle is invoked, $\mathcal{A}_2'$ is executed on a fresh copy of $|\Psi\rangle$.

Sample two bits c_0, c_1 uniformly at random. Now for every fixing of the values $\mathbf{r}, u_0, u_1$ such that $(\mathbf{r}, u_1) \in \mathsf{Good}_0', (\mathbf{r}, u_0) \in \mathsf{Good}_1'$, we have the following:

$$\mathsf{Pr}\left[\widetilde{r} \leftarrow \mathsf{GLDec}^{\mathcal{A}_1'(|\Psi\rangle, m_0, m_1, c_1, \cdot, u_1)}\right] \geq \frac{\nu'(\lambda)}{2} \text{ and } \mathsf{Pr}\left[\widetilde{r} \oplus r \leftarrow \mathsf{GLDec}^{\mathcal{A}_2'(|\Psi\rangle, m_0, m_1, c_0, u_0, \cdot)}\right] \geq \frac{\nu'(\lambda)}{2}$$

We now construct the following distinguisher.

$\underline{\mathcal{D}\left(\left(ot_1^{(1)}, \ldots, ot_1^{(\lambda)}\right), m_0, m_1, r, |\Phi\rangle\right)}$:

Preprocessing Phase:

- Set $\mathsf{ot}_1 = \left(ot_1^{(1)}, \ldots, ot_1^{(\lambda)}\right)$.
- Compute R^* on input $(\mathsf{ot}_1, |\Phi\rangle)$ to obtain ot_2^* and $|\Psi'\rangle$.
- Sample $c_0 \xleftarrow{\$} \{0,1\}, c_1 \xleftarrow{\$} \{0,1\}$ and $u_0 \xleftarrow{\$} \{0,1\}^\lambda, u_1 \xleftarrow{\$} \{0,1\}^\lambda$.
- Output the state $|\Psi\rangle = |\Psi'\rangle \otimes |c_0 c_1 u_0 u_1\rangle$.

Online Phase: We define the following adversary $\mathcal{A}_{onl}(|\Psi\rangle)$ which does the following: on input (b, u^*), for $b \in \{1,2\}$, $u^* \in \{0,1\}^\lambda$, it computes $\mathcal{A}_1'(|\Psi'\rangle, m_0, m_1, c_1, \cdot, u_1)$ if $b = 1$, otherwise it computes $\mathcal{A}_2'(|\Psi'\rangle, m_0, m_1, c_0, u_0, \cdot)$.

We compute $R^{\mathcal{A}}(r)$, where $R^{\mathcal{A}}(r)$ is defined as follows.

- Compute $\mathsf{GLDec}^{\mathcal{A}_{onl}(|\Psi\rangle)}$ to obtain v_1. Each oracle query by GLDec is appended with 1.
- Compute $\mathsf{GLDec}^{\mathcal{A}_{onl}(|\Psi\rangle)}$ to obtain v_2. Each oracle query by GLDec is appended with 2.

- If $v_1 \oplus v_2 = r$, output 1.

Post-processing phase: This is an identity function.

We note that the above distinguisher is a $\mathcal{R}_{\mathsf{GL}}$-RFC algorithm.

We claim that the probability that $\mathcal{D}$ outputs 1 is non-negligible. The probability that $\mathcal{D}$ outputs 1 is at least $\frac{(\nu'(\lambda)^2)\nu(\lambda)}{4}$. This follows from the fact that the probability that $v_1 = \widetilde{r} \oplus r$ and $v_2 = \widetilde{r}$ is at least $\frac{(\nu'(\lambda)^2)\nu(\lambda)}{4}$.

We claim the following.

Claim 47. *Assuming $\mathcal{R}_{\mathsf{GL}}$-RFC-security of QLWE, the following holds: for every $i \in [\lambda]$,*

$$\left|\mathsf{Pr}\left[1 \leftarrow \mathcal{D}\left(\left(ot_1^{(1)}, \ldots, ot_1^{(\lambda)}\right), m_0, m_1, r, |\Phi\rangle\right) : \begin{smallmatrix}\forall j \leq i, ot_1^j \leftarrow OT_1(1^\lambda, r_j) \\ \forall j > i, ot_1^j \leftarrow OT_1(1^\lambda, 0)\end{smallmatrix}\right] - \right.$$

$$\left.\mathsf{Pr}\left[1 \leftarrow \mathcal{D}\left(\left(ot_1^{(1)}, \ldots, ot_1^{(\lambda)}\right), m_0, m_1, r, |\Phi\rangle\right) : \begin{smallmatrix}\forall j < i, ot_1^j \leftarrow OT_1(1^\lambda, r_j) \\ \forall j \geq i, ot_1^j \leftarrow OT_1(1^\lambda, 0)\end{smallmatrix}\right]\right| \leq \mathsf{negl}(\lambda)$$

Proof. Suppose there exists $i^* \in [\lambda]$ such that the above statement is not true. Then, we construct a $\mathcal{R}_{\mathsf{GL}}$-RFC algorithm B that violates QLWE.

The algorithm B does the following: it first executes the pre-processing phase. In the pre-processing phase, it receives as input ot_1^* and advice $(m_0, m_1, r, |\Phi\rangle)$. It computes $ot_1^{(j)} \leftarrow OT_1(1^\lambda, r_j)$, for every $j < i$ and $ot_1^{(j)} \leftarrow OT_1(1^\lambda, 0)$ for every $j > i$. It then computes the preprocessing phase of $\mathcal{D}$ on input $((ot_1^{(1)}, \ldots, ot_1^{(i-1)}, ot_1^*, ot_1^{(i+1)}, \ldots, ot_1^{(\lambda)}))$. The output of the preprocessing phase is input to the online phase. The online phase and the post-processing phases of B is the same as that of $\mathcal{D}$. Note that B is also a $\mathcal{R}_{\mathsf{GL}}$-RFC algorithm.

The probability of B distinguishing is the same as the probability that $\mathcal{D}$ distinguishes. Thus, the above claim follows. □

This means that the probability that $\mathcal{D}$, on input $(ot_1^{(1)}, \ldots, ot_1^{(\lambda)}, m_0, m_1, r, |\Phi\rangle)$, where $ot_1^{(i)} \leftarrow OT_1(1^\lambda, 0)$ for all i, outputs 1 is non-negligible. This means that $v_1 \oplus v_2 = r$ with non-negligible probability, where v_1, v_2 is computed during the execution of the online phase.

But till the step v_1, v_2 is computed, r is never used either in the execution of $\mathcal{D}$. This means that r is information-theoretically hidden. Thus, the probability that $v_1 \oplus v_2 = r$ is negligible in $|r| = \lambda$. Thus, we have a contradiction.

5.3 Construction of Bounded Concurrent QZKPoK

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

- A post-quantum three-round statistical receiver-private oblivious transfer protocol, denoted by $\Pi_{\mathsf{OT}} = (\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$ (Section 5.2) with perfect correctness.

  We say that a transcript τ, consisting of messages $(\mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3)$, is valid with respect to sender's randomness r and its input bits (m_0, m_1) if the following holds: $(\mathsf{msg}_1, \mathsf{st}) \leftarrow \mathsf{OT}_1(1^\lambda; r)$ and $\mathsf{msg}_3 \leftarrow \mathsf{OT}_3(\mathsf{st}, \mathsf{msg}_2, m_0, m_1)$[12].

[12] We can assume without loss of generality that the third OT message does not require randomness.

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

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

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

We describe the construction in Figure 5.

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

- For every $i \in [\ell_w]$, P samples the shares $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random conditioned on $\oplus_{j=1}^{\lambda} sh_{i,j} = w_i$, where w_i is the i^{th} bit of w.
- For every $i \in [\ell_w]$, P samples the bits $\alpha_{i,1}, \ldots, \alpha_{i,\lambda}$ uniformly at random.
- For $i \in [\ell_w], j \in [\lambda]$, do the following:
  - $P \leftrightarrow V$: P and V execute Π_{OT} with V playing the role of the receiver in Π_{OT} and P playing the role of the sender in Π_{OT}. The input of the receiver in this protocol is 0, while the input of the sender is set to be $(sh_{i,j}, \alpha_{i,j})$ if $b_{i,j} = 0$, otherwise it is set to be $(\alpha_{i,j}, sh_{i,j})$ if $b_{i,j} = 1$, where the bit $b_{i,j}$ is sampled uniformly at random.
    Call the resulting transcript of the protocol to be $\tau_{\mathsf{OT}}^{(i,j)}$ and let $r_{\mathsf{OT}}^{(i,j)}$ be the randomness used by the sender in OT.
  - $P \rightarrow V$: P sends $b_{i,j}$ to V.
- $P \leftrightarrow V$: P and V execute Π_{zk} with P playing the role of the prover of Π_{zk} and V playing the role of the verifier of Π_{zk}. The instance is $\left(x, \left\{ \tau_{\mathsf{OT}}^{(i,j)}, b_{i,j} \right\}_{i\in[\ell_w],j\in[\lambda]} \right)$ and the witness is $\left(w, \left\{ r_{\mathsf{OT}}^{(i,j)}, sh_{i,j}, \alpha_{i,j} \right\}_{i\in[\ell_w],j\in[\lambda]} \right)$. If the verifier in Π_{zk} rejects, then V rejects.

Figure 5: Construction of Bounded Concurrent QZKPoK for NP.

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.

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[13]:

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

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

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

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

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

Let $W_{i,j} = \mathsf{Amplifier}\left(U^{(2)}_{i,j}U^{(1)}_{i,j}\right)$, where $\mathsf{Amplifier}$ is defined in Lemma 9. 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 $r^{i,j}_{\mathsf{OT}}$ 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.

$\underline{\mathsf{Hyb}_3}$: 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.

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

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

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

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

Hyb_1: In this hybrid, P and V^* interact with each other. The output of this hybrid is the output of V^*.

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 zero-knowledge property of Π_{zk}.

Hyb_3: The output of this hybrid is the output of $\mathsf{Sim}(x, |\Psi\rangle)$.

Claim 48. *Assuming the post-quantum sender-privacy (Definition #2) of Π_{OT}, the output of Hyb_2 is computationally indistinguishable from the output of Hyb_3.*

Proof. Consider the intermediate hybrids.

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

$\underline{\mathsf{Hyb}_{2.2}}$: This is essentially the same as $\mathsf{Hyb}_{2.1}$ except that the whole protocol is executed in superposition and the measurements are deferred to the end.

Hybrids $\mathsf{Hyb}_{2.1}$ and $\mathsf{Hyb}_{2.2}$ are identical from the fact that the measurements and the unitaries applied by the verifier commute.

In the hybrids below, we non-uniformly hardwire advice in the hybrid prover. We describe the (randomized) advice generation function below.

$\underline{\mathsf{AdviceGen}_1\left(|\Psi\rangle, w\right)}$:

- For every $i \in [\ell_w]$, $j \in [\lambda]$, sample the strings $r_{i,j}$ uniformly at random.
- For every $i \in [\ell_w]$, generate the bits $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random such that $\oplus_{j=1}^{\lambda} sh_{i,j} = w_i$.
- For every $i \in [\ell_w]$, generate the bits $\alpha_{i,1}, \ldots, \alpha_{i,\lambda}$ uniformly at random.
- For every $i \in [\ell_w]$, generate the bits $b_{i,1}, \ldots, b_{i,\lambda}$ uniformly at random.
- Let $|\Psi_{0,0}\rangle = |\Psi\rangle$. Set $m_0^{(i,j)} = sh_{i,j}$ and $m_1^{(i,j)} = \alpha_{i,j}$, if $b_{i,j} = 0$. Set $m_0^{(i,j)} = \alpha_{i,j}$ and $m_1^{(i,j)} = sh_{i,j}$, if $b_{i,j} = 1$. For $i \in [\ell_w], j \in [\lambda]$, do the following:
  - If $j = \lambda$ then set $\widehat{b}_{i,j} = 0$, otherwise set $\widehat{b}_{i,j}$ to be the bit such that the following holds:

$$\left| \mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}\left(r_{i,j}, |\Psi_{i,j}\rangle, m_0^{(i,j)}, m_1^{(i,j)}, 1 - \widehat{b}_{i,j}\right)\right] - \frac{1}{2} \right| \leq \nu(\lambda),$$

    where $\nu(\lambda)$ is a negligible function.
  - Execute the $(i,j)^{th}$ execution of the OT protocol with the sender's input being $\left(m_0^{(i,j)}, m_1^{(i,j)}\right)$ and also execute the receiver on input $|\Psi_{i,j}\rangle$. Let the output state of the receiver be $|\Psi_{i',j'}\rangle$, where the element immediately after (i,j) in the total ordering is (i',j').

Output the advice $\left(\left\{r_{i,j}\right\}_{i\in[\ell_w],j\in[\lambda]}, |\Psi\rangle, \left\{\left(m_0^{(i,j)}, m_1^{(i,j)}\right), b_{i,j}, \widehat{b}_{i,j}\right\}_{i\in[\ell_w],j\in[\lambda]}\right)$.

$\underline{\mathsf{Hyb}_{2.3}}$: We define a hybrid prover $\mathsf{Hyb}_{2.3}.P$ as follows. The following state is hardwired inside the prover:

$$\left(\left\{r_{i,j}\right\}_{i\in[\ell_w],j\in[\lambda]}, |\Psi\rangle, \left\{\left(m_0^{(i,j)}, m_1^{(i,j)}\right), b_{i,j}, \widehat{b}_{i,j}\right\}_{i\in[\ell_w],j\in[\lambda]}\right)$$

generated using $\mathsf{AdviceGen}_1\left(|\Psi\rangle, w\right)$. As in the previous hybrid, the verifier V^* is run in superposition. The input $\mathsf{Hyb}_{2.3}.P$ uses for the $(i,j)^{th}$ execution of the OT protocol is $\left(m_0^{(i,j)}, m_1^{(i,j)}\right)$. Immediately after the $(i,j)^{th}$ execution of the OT protocol, the hybrid prover sends $b_{i,j}$ to the verifier.

The output of this hybrid is identical to $\mathsf{Hyb}_{2.2}$.

We define an alternate advice generation as follows. This function is parameterized by (i^*, j^*).

$\mathsf{AdviceGen}_{(i^*,j^*)}(|\Psi\rangle, w)$:

- For every $i \in [\ell_w]$, $j \in [\lambda]$, sample the strings $r_{i,j}$ uniformly at random.
- For every $i \in [\ell_w]$, generate the bits $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random such that $\oplus_{j=1}^{\lambda} sh_{i,j} = w_i$.
- For every $i \in [\ell_w]$, generate the bits $\alpha_{i,1}, \ldots, \alpha_{i,\lambda}$ uniformly at random.
- Let $|\Psi_{0,0}\rangle = |\Psi\rangle$. Set $m_0^{(i,j)} = sh_{i,j}$ and $m_1^{(i,j)} = \alpha_{i,j}$ if $b_{i,j} = 0$. Set $m_0^{(i,j)} = \alpha_{i,j}$ and $m_1^{(i,j)} = sh_{i,j}$ if $b_{i,j} = 1$. For $i \in [\ell_w], j \in [\lambda]$, do the following:
  - If $j = \lambda$ then set $\widehat{b}_{i,j} = 0$, otherwise set $\widehat{b}_{i,j}$ to be the bit such that the following holds:

$$\left| \mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}\left(r_{i,j}, |\Psi_{i,j}\rangle, m_0^{(i,j)}, m_1^{(i,j)}, 1 - \widehat{b}_{i,j}\right)\right] - \frac{1}{2} \right| \leq \nu(\lambda),$$

    where $\nu(\lambda)$ is a negligible function.
  - If $(i,j) < (i^*, j^*)$ and $j \neq \lambda$, execute the OT protocol with the sender's input being $\left(m_{\widehat{b}_{i,j}}^{(i,j)}, m_{\widehat{b}_{i,j}}^{(i,j)}\right)$. If $(i,j) \geq (i^*, j^*)$ or if $j = \lambda$, execute the OT protocol with the sender's input being $\left(m_0^{(i,j)}, m_1^{(i,j)}\right)$. The receiver is executed on input the state $|\Psi_{i,j}\rangle$. Let the output state of the receiver be $|\Psi_{i',j'}\rangle$, where the element after (i,j) is (i',j').

Output the advice $\left(\{r_{i,j}\}_{i\in[\ell_w],j\in[\lambda]}, |\Psi\rangle, \left\{\left(m_0^{(i,j)}, m_1^{(i,j)}\right), b_{i,j}, \widehat{b}_{i,j}\right\}_{i\in[\ell_w],j\in[\lambda]}\right)$.

$\mathsf{Hyb}_{2.4.(i^*,j^*)}$ for $i \in [\ell_w], j \in [\lambda]$: We define a hybrid prover $\mathsf{Hyb}_{2.4.(i^*,j^*)}.P$ as follows. The following state is hardwired inside the prover:

$$\left(\{r_{i,j}\}_{i\in[\ell_w],j\in[\lambda]}, |\Psi\rangle, \left\{\left(m_0^{(i,j)}, m_1^{(i,j)}\right), b_{i,j}, \widehat{b}_{i,j}\right\}_{i\in[\ell_w],j\in[\lambda]}\right)$$

generated using $\mathsf{AdviceGen}_{(i^*,j^*)}(|\Psi\rangle, w)$. As in the previous hybrid, the verifier V^* is run in superposition. The input $\mathsf{Hyb}_{2.4.(i^*,j^*)}.P$ uses for the $(i,j)^{th}$ execution of the OT protocol, if $(i,j) < (i^*, j^*)$ and if $j \neq \lambda$, is $\left(m_{\widehat{b}_{i,j}}^{(i,j)}, m_{\widehat{b}_{i,j}}^{(i,j)}\right)$. Otherwise if $(i^*, j^*) \leq (i,j)$ or if $j = \lambda$, the input $\mathsf{Hyb}_{2.4.(i^*,j^*)}.P$ uses for the $(i,j)^{th}$ execution of the OT protocol is $\left(m_0^{(i,j)}, m_1^{(i,j)}\right)$. Immediately after the $(i,j)^{th}$ execution of the OT protocol, the hybrid prover sends $b_{i,j}$ to the verifier.

The output distributions of the previous hybrid and $\mathsf{Hyb}_{2.4.(i^*,j^*)}$ is computationally indistinguishable from the post-quantum sender security of OT.

$\underline{\mathsf{Hyb}_{2.5}}$: In this hybrid, we define a hybrid prover $\mathsf{Hyb}_{2.5}.P$ that aborts if there exists $i \in [\ell_w]$ s.t. $b_{i,j} = \widehat{b}_{i,j}$ for all $j \in [\lambda - 1]$. Otherwise, it executes the same procedure as the prover in the previous hybrid.

We claim that the hybrid prover, $\mathsf{Hyb}_{2.5}.P$ aborts with probability negligible in the security parameter. This follows from that fact that for any $i \in [\ell_w]$, the probability that for all $j \in [\lambda - 1]$, $b_{i,j} = \widehat{b}_{i,j}$ holds is $\frac{1}{2^{\lambda-1}}$. This means that the probability of aborting is $\frac{\ell_w}{2^{\lambda-1}}$. Conditioning on not aborting, the output in $\mathsf{Hyb}_{2.4.(l_w,\lambda)}$ and $\mathsf{Hyb}_{2.5}$ are the same. Thus, the hybrids $\mathsf{Hyb}_{2.4.(l_w,\lambda)}$ and $\mathsf{Hyb}_{2.5}$ are statistically close.

Remark 49. *To understand the reason why we only upper bound the probability of $b_{i,j} = \widehat{b}_{i,j}$ for all $j \leq \lambda - 1$, consider the following. Suppose the intial state of the verifier contains the witness w. Fix i. In the definition of OT, we allow the bit $\widehat{b}_{i,j}$ to depend on the messages $(m_0^{(i,j)}, m_1^{(i,j)})$ and the state of the receiver just before the $(i,j)^{th}$ execution of the OT protocol. Moreover, the state of the receiver just before the $(i,j)^{th}$ execution contains the bits $b_{i,j'}$ for all $j' < j$ in addition to the messages $\{(m_0^{i,j'}, m_1^{(i,j')})\}$ and the witness w. Now, consider the case when $j = \lambda$. Then from $\{(m_0^{i,j'}, m_1^{(i,j')})\}_{j' \leq j}, \{b_{i,j'}\}_{j'<j}$ and w, we can determine $b_{i,j}$. Thus, $\widehat{b}_{i,j}$ could be the same as $b_{i,j}$.*

We define a final hybrid advice generation algorithm.

$\underline{\mathsf{AdviceGen}_2^{(i^*)}(|\Psi\rangle, w)}$:

- For every $i \in [\ell_w]$, $j \in [\lambda]$, sample the strings $r_{i,j}$ uniformly at random.
- For every $i \in [\ell_w]$, generate the bits $sh_{i,1}, \ldots, sh_{i,\lambda}$ uniformly at random such that $\oplus_{j=1}^{\lambda} sh_{i,j} = w_i$.
- For every $i \in [\ell_w]$, generate the bits $\alpha_{i,1}, \ldots, \alpha_{i,\lambda}$ uniformly at random.
- Let $|\Psi_{0,0}\rangle = |\Psi\rangle$. Set $m_0^{(i,j)} = sh_{i,j}$ and $m_1^{(i,j)} = \alpha_{i,j}$ if $b_{i,j} = 0$. Set $m_0^{(i,j)} = \alpha_{i,j}$ and $m_1^{(i,j)} = sh_{i,j}$ if $b_{i,j} = 1$. For $i \in [\ell_w], j \in [\lambda]$, do the following:
  - If $j = \lambda$ then set $\widehat{b}_{i,j} = 0$, otherwise set $\widehat{b}_{i,j}$ to be the bit such that the following holds:
$$\left| \mathsf{Pr}\left[\mathcal{A} \text{ wins } \widehat{G}\left(r_{i,j}, |\Psi_{i,j}\rangle, m_0^{(i,j)}, m_1^{(i,j)}, 1 - \widehat{b}_{i,j}\right)\right] - \frac{1}{2} \right| \leq \nu(\lambda)$$
  - If $i \geq i^*$ and $j = \lambda$, execute the OT protocol with the sender's input $(m_0^{(i,j)}, m_1^{(i,j)})$. Otherwise if $(i < i^*) \wedge (j = \lambda)$ or $j \neq \lambda$, execute the OT protocol with the sender's input $\left(m_{\widehat{b}_{i,j}}^{(i,j)}, m_{\widehat{b}_{i,j}}^{(i,j)}\right)$.

Output the advice $\left(\{r_{i,j}\}_{i\in[\ell_w],j\in[\lambda]}, |\Psi\rangle, \left\{\left(m_0^{(i,j)}, m_1^{(i,j)}\right), b_{i,j}, \widehat{b}_{i,j}\right\}_{i\in[\ell_w],j\in[\lambda]}\right)$.

$\underline{\mathsf{Hyb}_{2.6.i^* \text{ for } i^* \in [\ell_w]}}$: We define a hybrid prover $\mathsf{Hyb}_{2.6.i^*}.P$ as follows. The following state is hardwired inside the prover:

$$\left(\{r_{i,j}\}_{i\in[\ell_w],j\in[\lambda]}, |\Psi\rangle, \left\{\left(m_0^{(i,j)}, m_1^{(i,j)}\right), b_{i,j}, \widehat{b}_{i,j}\right\}_{i\in[\ell_w],j\in[\lambda]}\right)$$

generated using $\mathsf{AdviceGen}_2^{(i^*)}(|\Psi\rangle, w)$. It aborts if there exists $i \in [\ell_w]$ such that $b_{i,j} = \widehat{b}_{i,j}$ for all $j \in [\lambda - 1]$. Otherwise, it does the following: it computes the $(i,j)^{th}$ execution of the OT protocol, for all $i \in [\ell_w], j \in [\lambda]$, where the input of the OT sender in the $(i,j)^{th}$ execution is $\left(m_{\widehat{b}_{i,j}}^{(i,j)}, m_{\widehat{b}_{i,j}}^{(i,j)}\right)$ if $(i < i^*) \wedge (j > \lambda)$ or $j \neq \lambda$, otherwise if $i \geq i^* \wedge j = \lambda$, the input to the sender of the $(i,j)^{th}$ execution of the OT protocol is $\left(m_0^{(i,j)}, m_1^{(i,j)}\right)$. Immediately after the $(i,j)^{th}$ execution of the OT protocol, the hybrid prover sends $b_{i,j}$ to the verifier.

$\underline{\mathsf{Hyb}_{2.7}}$: This is identical to Hyb_3.

The output distribution of this hybrid is identical to the output distribution of $\mathsf{Hyb}_{2.6.\ell_w}$. Conditioned on not aborting, the distribution of the inputs the prover uses in all the OT protocols is the same in both the hybrids.

□

6 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 – if we instantiate the coin-flipping protocol and the QZK for NP with ones satisfying (resp., bounded) concurrent security, the resulting QZK for QMA protocol would also satisfy (resp., bounded) concurrent security. We already constructed a bounded concurrent QZK for NP protocol in Section 4. First we construct a bounded concurrent coin-flipping protocol. Then we will show how to use this to construct a bounded concurrent QZK for QMA.

6.1 Post-Quantum Concurrent Coin-Flipping

We begin by defining concurrent coin-flipping in the post-quantum setting. Informally, a coin-flipping protocol allows an unbounded party, P_1, to interact with a QPT party, P_2, in order for both of them to agree and output a random string.

To define concurrent coin-flipping formally, we define $\mathcal{F}$ to be a an input-less two-party functionality that, upon being invoked, outputs a random string to both the parties. We define the security using the simulation paradigm [Lin17].

We also define the notion of Q-session quantum adversaries below. We define this along the lines of Definition 13.

Definition 50 (Q-session Quantum Adversary)**.** *Let $Q \in \mathbb{N}$. Let Π be an interactive protocol between a classical PPT algorithm P_1 and a classical PPT algorithm P_2 for the relation $\mathcal{R}(\mathcal{L})$. Let $(x,w) \in \mathcal{R}(\mathcal{L})$. We say that an adversarial non-uniform QPT verifier P_2^* is a* **Q-session adversary** *if it invokes Q sessions with P_1.*

Moreover, we assume that the interaction P_1 and P_2^* is defined as follows: denote by $P_{2,i}^*$ to be the algorithm used by P_2^* in the i^{th} session and denote by $P_{1,i}$ to be the i^{th} invocation of P_1 interacting with $P_{2,i}^*$. Every message sent by P_2^* is of the form $\left((1, \mathsf{msg}_1), \ldots, (Q, \mathsf{msg}_Q)\right)$, where

msg_i *is defined as:*

$$\mathsf{msg}_i = \begin{cases} \mathsf{N/A}, & \text{if } P^*_{2,i} \text{ doesn't send a message,} \\ (t, z), & \text{if } P^*_{2,i} \text{ sends } z \text{ in the round } t \end{cases}$$

P_i *responds to* msg_i*. If* $\mathsf{msg}_i = \mathsf{N/A}$ *then it sets* $\mathsf{msg}'_i = \mathsf{N/A}$*. If* $P^*_{2,i}$ *has sent the messages in the correct order, then* $P_{1,i}$ *applies the next message function on its own private state and* msg_i *to obtain the quantum state* z' *and sets* $\mathsf{msg}'_i = (t+1, z')$*. Otherwise, it sets* $\mathsf{msg}'_i = (\bot, \bot)$*. Finally,* P^*_2 *receives* $\Big((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q)\Big)$.

Using the notion of Q-session quantum adversaries, we define (bounded) concurrent post-quantum coin-flipping protocols below.

Definition 51 ((Bounded) Concurrent Post-Quantum Coin-Flipping)**.** *Let* $Q \in \mathbb{N}$*. A protocol* Π*, between a classical probabilistic polynomial-time (in* Q*) algorithm* P_1 *and a classical probabilistic polynomial-time (in* Q*) algorithm* P_2*, if the following two conditions are satisfied.*

- **Completeness***: If both* P_1 *and* P_2 *behave according to the protocol, then the string* $r \in \{0,1\}^\lambda$ *output by both the parties is distributed according to the uniform distribution.*
- **Unbounded Simulation Security Against** P_1*: For every malicious (computationally unbounded) party* P^*_1*, there exists an unbounded simulator* Sim *such that such that the output distribution of the honest party along with* P^*_1*'s output state in the real world is statistically indistinguishable from the output distribution of the honest party along with* P^*_1*'s output state in the ideal world.*
- **Post-Quantum Simulation Security Against Bounded Concurrent** P_2*: For every malicious non-uniform* Q*-session adversary* P^*_2*, exists a QPT simulator* Sim *such that the output distribution of the honest party along with* P^*_2*'s output state in the real world is computationally indistinguishable (even against QPT distinguishers) from the output distribution of the honest party along with* P^*_2*'s output state in the ideal world.*

6.1.1 Construction

We start with the template of the quantum-secure instantiation, namely [DL09], of Blum's coin-flipping protocol [Blu83]. However,[DL09] uses Watrous rewinding to prove its security. Instead we use a generic zero-knowledge protocol to avoid using Watrous rewinding (at least explicitly). This will also help us achieve concurrent security.

We construct a one-bit coin-flipping protocol. This can be extended to a coin-flipping protocol for long strings by sequential repetition.

To construct a bounded concurrent coin-flipping protocol Π, we use the following ingredients:

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

$$\mathcal{L} = \Big\{((\mathbf{r}, \mathbf{c}, a)\ ;\ (r))\ :\ \mathsf{Comm}(1^\lambda, \mathbf{r}, a; r) = \mathbf{c}\Big\}$$

We note that the same construction would yield fully concurrent coin-flipping protocol if one were to start with fully concurrent QZK.

We describe the construction Π in Figure 6.

- $P_1 \leftrightarrow P_2$: P_1 chooses a bit $a \xleftarrow{\$} \{0,1\}$. It then commits to a using the commitment protocol $(\mathsf{Comm}, \mathsf{R})$. Let $\mathbf{r}$ be the message sent by the receiver and let $\mathbf{c}$ be the message sent by the prover. Let r be the randomness used by P_1 such that $\mathsf{Comm}(1^\lambda, \mathbf{r}, a; r) = \mathbf{c}$.
- $P_2 \rightarrow P_1$: P_2 sends a bit $b \xleftarrow{\$} \{0,1\}$ to P_1.
- $P_1 \rightarrow P_2$: P_1 sends a.
- $P_1 \leftarrow P_2$: P_1 and P_2 execute the zero-knowledge protocol, Π_{NP}, with P_1 playing the role of the prover and P_2 playing the role of the verifier. The instance is $(\mathbf{r}, \mathbf{c}, a)$ and the witness is r.
- Finally, both parties output $s = a \oplus b$.

Figure 6: Construction of Bounded Concurrent Coin-Flipping Protocol.

We prove the security of Π below.

Lemma 52. Π *is a secure post-quantum coin-flipping bounded-concurrent protocol (Definition 51).*

Proof. Completeness follows from the completeness of Π_{NP}.

Unbounded Simulation Security Against P_1. Suppose P_1^* is a malicious unbounded adversary.

We define a simulator Sim as follows. It receives as input a commitment from P_1^*. It then runs in exponential time and finds the bit a committed to by P_1. It receives the bit s from the ideal world functionality. It sets $b = s \oplus a$. It sends b to P_1^*. It receives a' from P_1. If $a' \neq a$ then abort. It then runs the verifier in the ZK protocol Π_{NP}.

The statistical indistinguishability of the output distributions of the ideal and the real world follow from the statistical binding property of $(\mathsf{Comm}, \mathsf{R})$.

Post-Quantum Simulation Security Against Bounded Concurrent P_2. Suppose P_2^* is a malicious QPT adversary.

We define a simulator Sim as follows. It commits to 0 in the protocol $(\mathsf{Comm}, \mathsf{R})$. It receives b from P_2^*. It also receives the string s from the ideal world functionality. It computes $a = s \oplus b$. It sends a to P_2^*. It then runs the simulator of Π_{zk}.

We prove the computational indistinguishability of the output distributions of the ideal and the real world via a hybrid argument.

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

$\underline{\mathsf{Hyb}_2}$: This is the same as the previous hybrid except that the ZK protocol is simulated.

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

$\underline{\mathsf{Hyb}_3}$: This is the same as the previous hybrid except that the bit committed to in $(\mathsf{Comm}, \mathsf{R})$ is 0.

The computational indistinguishability of Hyb_2 and Hyb_3 follows from the computational hiding property of $(\mathsf{Comm}, \mathsf{R})$.

$\underline{\mathsf{Hyb}_4}$: This corresponds to the ideal world.

The output distributions of Hyb_3 and Hyb_4 are identical.

□

6.2 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 53 (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 $|\psi\rangle$, $\mathrm{poly}(\lambda)$-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], but we instantiate the underlying primitives with ones satisfying bounded concurrent security.

Construction. We use the following ingredients in our construction:

- Statistical-binding and quantum-concealing commitment scheme, $(\mathsf{Comm}, \mathsf{R})$ (Section 2.2).
- Post-quantum bounded-concurrent coin flipping protocol (Section 6.1). Let Q be the maximum number of sessions associated with this protocol.
- Bounded concurrent QZK proof system, denoted by Π_{NP}, for the following language (Section 4).

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

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 6.2. We prove the following.

Theorem 54. *Assuming that Π_{coin} satisfies the definition of post-quantum bounded concurrent coin flipping and Π_{NP} satisfies the definition of bounded concurrent QZK for NP, the protocol given in Figure 6.2 is a bounded concurrent QZK protocol for QMA with soundness $\frac{1}{\mathrm{poly}}$.*

Remark 55. *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. The completeness and soundness follow from [BJSW16]. We sketch the proof of bounded concurrent zero-knowledge below.

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 and $\mathbf{c}$ be the commitment.
- $P \to V$: P sends $\mathsf{E}_s(|\psi\rangle)$.
- Prover and verifier perform the coin-flipping protocol Π_{coin} to choose a random element of $\{1, 2, ..., M\}$. Let r^* be the output.
- 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 $(\mathbf{r}, \mathbf{c}, r^*, y)$ and the witness (s, ℓ).

Figure 7: Bounded-Concurrent QZK for QMA

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 $\mathbf{X}_{zk}, \mathbf{X}_{coin}, \mathbf{X}_{anc}, \mathbf{M}, \mathbf{Y}_{ver}$. The register $\mathbf{X}_{zk}$ is used by the simulator of the bounded concurrent QZK protocol, $\mathbf{X}_{coin}$ is used by the simulator of the bounded concurrent coin flipping 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 $\mathbf{Y}_{ver}$ is used for storing the private state of the verifier. Initialize the registers $\mathbf{X}_{zk}, \mathbf{X}_{coin}, \mathbf{M}$ 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 $\mathbf{Y}_{ver}$ 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}. Let S_2 be the set of indices such that for every $i \in S_2$, msg_i is a message in the protocol Π_{coin}. Finally, let $S_3 = [Q] \backslash (S_1 \cup S_2)$.
- 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. It copies $((1, \mathsf{msg}_1), \ldots, (Q, \mathsf{msg}_Q))$ into $\mathbf{X}_{\mathrm{coin}}$ and for every

$i \notin S_2$, 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 $i \in S_3$, 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 commitment of s_i.
- 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 $(\mathbf{r}_i, \mathbf{c}_i, r_i^*, y_i)$, where r_i^* was generated in the beginning and $\mathbf{r}_i, \mathbf{c}_i, y_i$ is generated as specified in the protocol. It applies the simulator of Π_{coin} on $\mathbf{X}_{coin}$ to obtain $((1, \mathsf{msg}'_{1,coin}), \ldots (Q, \mathsf{msg}'_{Q,coin}))$. The input of the i^{th} session simulator of Π_{coin} is r_i^*.
- Determine $((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q))$ as follows. Set $\mathsf{msg}'_i = \mathsf{msg}_{i,\mathsf{zk}}$, if $i \in S_1$. Set $\mathsf{msg}'_i = \mathsf{msg}_{i,coin}$ if $i \in S_2$. Set $\mathsf{msg}'_i = \mathsf{msg}_{i,rest}$ if $i \in S_3$. 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 ZK property of Π_{NP}.

$\underline{\mathsf{Hyb}_4}$: Simulate the coin-flipping protocol Π_{coin} simultaneously for all the sessions. Other than this, the rest of the hybrid is the same as before.

The output distributions of Hyb_3 and Hyb_4 are computationally indistinguishable from the bounded concurrent security of the coin-flipping protocol Π_{coin}.

$\underline{\mathsf{Hyb}_{5.i} \text{ for } i \in [Q]}$: For every $j \leq i$, the 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 $\mathsf{Hyb}_{5.i-1}$ and $\mathsf{Hyb}_{5.i}$ are computationally indistinguishable from the quantum concealing property of $(\mathsf{Comm}, \mathsf{R})$.

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

The output distributions of $\mathsf{Hyb}_{6.i-1}$ 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 6.2, include a quantum proof of classical knowledge system for NP (for instance, the one we constructed in Section 5.3) 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 6.2.

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 and Zhengzhong Jin for patiently answering questions regarding [GJJM20], 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 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.

[Blu83] Manuel Blum. Coin flipping by telephone a protocol for solving impossible problems. *ACM SIGACT News*, 15(1):23–27, 1983.

[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 pre-processing. In *Annual International Cryptology Conference*, pages 485–502. Springer, 1999.

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

[DL09] Ivan Damgård and Carolin Lunemann. Quantum-secure coin-flipping and applications. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 52–69. Springer, 2009.

[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.

[GL89] Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, 1989.

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

[Har] Aram Harrow. Personal communication.

[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.

[KNY20] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. *arXiv preprint arXiv:2010.11186*, 2020.

[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.

[Lin17] Yehuda Lindell. How to simulate it–a tutorial on the simulation proof technique. In *Tutorials on the Foundations of Cryptography*, pages 277–346. Springer, 2017.

[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 *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, 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 *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 366–375. IEEE, 2002.

[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.

[RZ20] Bhaskar Roberts and Mark Zhandry. Franchised quantum money. *http://people.eecs.berkeley.edu/ bhaskarr/documents/FQM_RZ20.pdf*, 2020.

[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.

A General Cloning Security

The definition of reduction-friendly cloning security formulated earlier was closely tied to our security reduction. We could consider a more general and natural definition of cloning security, where we allow the QPT adversary to have access to a cloning oracle; that is, an oracle that on input a state $|\Psi\rangle$ and an integer k, outputs $|\Psi\rangle^{\otimes k}$. We can furthermore parameterize this assumption with q, where q is the number of queries made by the oracle. We note that QLWE is insecure against general cloning adversaries even if they make just one query; the attack is the same as the one presented in Section 5.1.

However, it is still meaningful to investigate the security of other cryptographic concepts with respect to the general cloning security definition. For instance, does there exist a one-way function secure against general cloning adversaries? Answering this question in the affirmative might open up the possibility of proving post-quantum security of existing interactive protocols. As an example, this might be useful in proving post-quantum security of interactive protocols that use hardcore bits [GL89]. In the security reduction, we execute the adversary multiple times to break the one-way function and thus, in this scenario, cloning security might be useful. On the other hand, if the answer to the above question is no, then this might be useful in constructing primitives with uncloneability property; that is, primitives where cloning would violate the security definition (for example, quantum money). As an example, [RZ20] showed that SIS is insecure against general cloning adversaries and then turned this fact into a construction of franchised quantum money.

On the negative side, we show later, that any NP language can be solved by a general cloning algorithm making large number of queries. This rules out the existence of any cryptographic assumption secure against general cloning adverasries, where no upper bound is placed on the number of cloning queries made by the QPT adversary.

We now formally introduce the notion of general cloning.

Definition 56 (q-General Cloning)**.** *We say that a (non-uniform) QPT algorithm is a general cloning algorithm if it takes as input x and makes q queries to an oracle $\mathcal{O}$, where $\mathcal{O}$ is an oracle that takes as input $|\Psi\rangle$, integer k and outputs $|\Psi\rangle^{\otimes k}$.*

We investigate the power of general cloning algorithms. We show, in Section A.1, that any NP language can be solved by a q-general cloning algorithm, for a large q. Moreover, even for $q = 1$, it was shown in [RZ20] that SIS (and hence, LWE for the parameters we consider) is false even against 1-general cloning algorithms.

A.1 Adaptive Attack

The following attack is based on a suggestion by Harrow [Har]: for any language L in NP, there exists a quantum polynomial-time algorithm, making q (adaptive) queries to the cloning oracle, that solves L.

Theorem 57. *Consider an NP relation $\mathcal{R} = \{(x, w)\}$. Let the size of a witness associated with instance x be upper bounded by $p(|x|)$. Then there exists a quantum polynomial-time algorithm, taking as input x, making $p(|x|)$ queries to oracle $\mathcal{O}$ (defined below), it outputs a witness w for x with probability $\geq \frac{1}{3}$.*

Oracle $\mathcal{O}$ takes as input a pure state and a positive integer, i.e. $(|\Psi\rangle, k)$, and outputs $|\Psi\rangle^{\otimes k}$.

Input: instance $x \in L$. Let n be the size of the witness.

Prepare the following state:

$$\begin{aligned} |\Psi_0\rangle &= \sum_y \frac{1}{\sqrt{2^n}}|y\rangle|C(x,y)\rangle \\ &= \frac{1}{\sqrt{2^n}}\left(\sum_{y:C(x,y)=1}|y\rangle|1\rangle + \sum_{y:C(x,y)=0}|y\rangle|0\rangle\right) \\ &= \sqrt{\varepsilon}|\phi_1\rangle|1\rangle + \sqrt{1-\varepsilon}|\phi_0\rangle|0\rangle \end{aligned}$$

For $i \in [n]$, do the following:

- Let $|\Psi_{i-1}\rangle = \sqrt{\varepsilon_{i-1}}|\phi_1\rangle|1\rangle + \sqrt{\delta_{i-1}}|\phi_0\rangle|0\rangle$
- Query the oracle with $(|\Psi_{i-1}\rangle, |x|)$ to obtain $|\Psi_{i-1}\rangle^{\otimes|x|}$.
- For $j \in [|x|]$, do the following:
  - Measure $|\Psi_{i-1}\rangle$ with respect to $\{I \otimes M_0, I \otimes M_1\}$, where:

$$M_0 = \frac{1}{\sqrt{2}}|0\rangle\langle 0| + |1\rangle\langle 1|$$

$$M_1 = \frac{1}{\sqrt{2}}|0\rangle\langle 0|$$

  - If the measurement outcome is 0, abort this loop and go to the $(i+1)^{th}$ execution. Let $|\Psi_i\rangle$ be the post-measurement state. Otherwise if the measurement outcome is 1, continue.
- If in all the $|x|$ iterations, the measurement outcome is 1, output $\perp$.

Measure the first register with respect to computational basis.
Output the resulting measurement outcome y.

Figure 8: Adaptive Attack

Proof. Let C be the verification circuit associated with R. We describe the algorithm in Figure 8.

We now argue that with probability $\geq \frac{1}{3}$, the algorithm in Figure 8 outputs a witness for x.

We first argue how δ_i evolves during the computation by performing induction on i. The base case corresponds to δ_1. Note that $\delta_1 = \frac{\delta_0}{2\alpha_1}$, where $\delta_0 = 1 - \varepsilon$. Moreover, $\alpha_1 = \langle\Psi_0|M_0^\dagger M_0|\Psi_0\rangle =$

$\frac{(1+\varepsilon)}{2}$. We now prove the induction hypothesis for $i \geq 2$.

Lemma 58. *Suppose* $\delta_{i-1} = \frac{\delta_0}{(\prod_{j=1}^{i-1}\alpha_j)\cdot 2^{i-1}}$, *where* $|\Psi_{i-1}\rangle = \sqrt{\varepsilon_{i-1}}|\phi_1\rangle|1\rangle + \sqrt{\delta_{i-1}}|\phi_0\rangle|0\rangle$, *is the state computed at the beginning of the* i^{th} *iteration, where* $\alpha_j = \langle\Psi_{j-1}|M_0^\dagger M_0|\Psi_{j-1}\rangle = \frac{(1+(2^j-1)\varepsilon)}{2^j}$.

The probability that the i^{th} *iteration aborts is* $\frac{1}{2^{|x|}}$. *Conditioned on the* i^{th} *iteration not aborting, the state* $|\Psi_i\rangle$ *at the beginning of the* $(i+1)^{th}$ *iteration is* $\sqrt{\varepsilon_i}|\phi_1\rangle|1\rangle + \sqrt{\delta_i}|\phi_0\rangle|0\rangle$, *where:*

- $\alpha_i = \frac{(1+(2^i-1)\varepsilon)}{2^i}$ *and,*
- $\delta_i = \frac{\delta_0}{(\prod_{j=1}^{i}\alpha_j)\cdot 2^i}$.

Proof. The probability that the i^{th} iteration aborts is $(\beta_i)^{|x|}$ since β_i is the probability that the measurement outcome is 1. We calculate β_i as follows:

$$\begin{aligned}\beta_i &= \langle\Psi_{i-1}|M_1^\dagger M_1|\Psi_{i-1}\rangle \\ &= \frac{\delta_{i-1}}{2} \leq \frac{1}{2}\end{aligned}$$

Thus, we have the probability that the i^{th} iteration aborts is $\frac{1}{2^{|x|}}$.

Suppose the i^{th} iteration does not abort then the state $|\Psi_i\rangle$ is computed as follows:

$$|\Psi_i\rangle = \frac{(I\otimes M_0)|\Psi_{i-1}\rangle}{\sqrt{\alpha_i}} = \frac{1}{\sqrt{\alpha_i}}\left(\sqrt{\varepsilon_{i-1}}|\phi_1\rangle|1\rangle + \sqrt{\frac{\delta_{i-1}}{2}}|\Psi_{i-1}\rangle|0\rangle\right),$$

where $\alpha_i = \langle\Psi_{i-1}|M_0^\dagger M_0|\Psi_{i-1}\rangle$.

We have $\delta_i = \frac{\delta_{i-1}}{2\alpha_i}$. By substituting in δ_{i-1}, we have that: $\delta_i = \frac{\delta_0}{(\prod_{j=1}^{i}\alpha_i)\cdot 2^i}$.

Finally, we prove that $\prod_{j=1}^{i}\alpha_j = \frac{(1+(2^n-1)\varepsilon)}{2^n}$.

$$\begin{aligned}\alpha_i = \langle\Psi_i|M_0^\dagger M_0|\Psi_i\rangle &= \frac{1}{\prod_{j=1}^{i-1}\alpha_j}\left(\frac{1-\varepsilon}{2^i}+\varepsilon\right) &(1)\\ &= \frac{1}{\prod_{j=1}^{i-1}\alpha_j}\cdot\left(\frac{1-\varepsilon}{2^i}+\varepsilon\right) &(2)\\ &= \frac{1}{\prod_{j=1}^{i-1}\alpha_i}\cdot\left(\frac{1+(2^i-1)\varepsilon}{2^i}\right) &(3)\end{aligned}$$

Now, we have $\prod_{j=1}^{i}\alpha_j = \alpha_i\cdot\prod_{j=1}^{i-1}\alpha_j = \frac{1}{\prod_{j=1}^{i-1}\alpha_i}\cdot\left(\frac{1+(2^i-1)\varepsilon}{2^i}\right)\cdot\prod_{j=1}^{i-1}\alpha_i = \left(\frac{1+(2^i-1)\varepsilon}{2^i}\right)$. □

The following lemma will finish the proof of the theorem.

Lemma 59. *The probability that the algorithm aborts in any of the* n *iterations is at most* $\frac{n}{2^{|x|}}$. *Conditioned on the algorithm not aborting, the final state of the system is the following:*

$$|\Psi_n\rangle = \sqrt{\varepsilon_n}|\phi_1\rangle|1\rangle + \sqrt{\delta_n}|\phi_0\rangle|0\rangle,$$

where $\delta_n \leq \frac{2}{3}$.

Proof. Thus, we have the following:

$$\begin{aligned}
\delta_n &= \frac{\delta_0}{(\prod_{i=1}^{n} \alpha_i) \cdot 2^n} \\
&\leq \frac{\delta_0}{\left(\frac{2-\varepsilon}{2^n}\right) \cdot 2^n} \\
&= \frac{\delta_0}{2-\varepsilon} \\
&\leq \frac{\delta_0}{1.5} \\
&\leq \frac{2}{3}
\end{aligned}$$

□

□