On the Concurrent Composition of Quantum Zero-Knowledge

Prabhanjan Ananth[1], Kai-Min Chung[2], and Rolando L. La Placa[3]

[1] UCSB, Santa Barbara, USA
prabhanjan@cs.ucsb.edu
[2] Academia Sinica, Taiwan
kmchung@iis.sinica.edu.tw
[3] MIT, Cambridge, USA
rlaplaca@mit.edu

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

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

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

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

  Even in the standalone setting, 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.

1 Introduction

Zero-knowledge [GMR85] is one of the foundational concepts in cryptography. A zero-knowledge system for NP is an interactive protocol between a prover P, who

receives as input an instance x and a witness w, and a verifier V who receives as input an instance x. The (classical) zero-knowledge property roughly states that the view of the malicious probabilistic polynomial-time verifier V^* generated after interacting with the prover P can be simulated by a PPT simulator, who doesn't know the witness w.

Protocol Composition in the Quantum Setting. Typical zero-knowledge proof systems only focus on the case when the malicious verifier is classical. The potential threat of quantum computers forces us to revisit this definition. There are already many works [ARU14, BJSW16, BG19, BS20, ALP20, VZ20, ABG+20], starting with the work of Watrous [Wat09], that consider the definition of zero-knowledge against verifiers modeled as quantum polynomial-time (QPT) algorithms; 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 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 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 concurrency literature [Lin03, PR03, Pas04, PTW09]. Moreover, we note that the only other existing work that constructs quantum zero-knowledge against multiple verifiers albeit in the parallel composition setting, namely [ABG$^+$20]*, also works in the bounded setting. We prove the following.

Theorem 1 (Informal). *Assuming the existence of post-quantum one-way functions†, 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‡. We note that achieving public-coin *unbounded* concurrrent ZK is impossible [PTW09] even in the classical setting.

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 $|\Psi\rangle$[§]. Then there exists an efficient extractor, with

*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.

†That is, one-way functions secure against (non-uniform) quantum polynomial-time algorithms.

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

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

black-box access to the prover, that can output a witness w for x with probability δ. Additionally, it also outputs a quantum state $|\Phi\rangle$. Ideally, we require the following two conditions to hold: (i) $|\varepsilon - \delta|$ is negligible and, (ii) the states $|\Psi\rangle$ and $|\Phi\rangle$ are close in trace distance; this property is also referred to as simulatability property. Unruh [Unr12] presented a construction of quantum proofs of knowledge; their construction satisfies (i) but not (ii). Indeed, the prover's state, after it interacts with the extractor, could be completely destroyed. Condition (ii) is especially important if we were to use quantum proofs of knowledge protocols as a sub-routine inside larger protocols, for instance in secure multiparty computation protocols.

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

We show the following.

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

Unlike all of the previous quantum proof of knowledge protocols which make use of Unruh's rewinding technique, we 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 the existing 3-message Σ protocols.

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

We show the following.

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

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

Our construction considers a simplified version of the framewok of [BJSW16]* and instantiates the underlying primitives in their protocol with bounded concurrent secure constructions.

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

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

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 Guide to the Reader

We present the overview of our results in the technical sections, just before presenting a formal description of the results.

- In Section 2, we present the definitions of concurrent QZK proof systems for NP and QMA. In the same section, we present definitions of quantum proof of knowledge.
- **Bounded Concurrent QZK**: In Section 3, we present the construction of bounded concurrent QZK for NP. We first begin with an overview of the construction and then present the formal construction in the same section. The proofs are presented in the Appendix (see the relevant references at the end of Section 3).
- **QZK Proof of Knowledge**: In Section 4.1, we present the construction of bounded concurrent QZK proof of knowledge for NP. We first begin with an overview of the construction and then present the formal construction in the same section. This construction involves the tool of oblivious transfer; we present the definition and the construction of oblivious transfer in the Appendix.
- **Bounded Concurrent QZK for QMA**: Finally, we present a construction of bounded concurrent QZK for QMA in Section 5.

2 Concurrent Quantum ZK Proof Systems: Definitions

We denote the security parameter by λ.

We denote the (classical) computational indistiguishability of the two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ by $\mathcal{D}_0 \approx_{c,\varepsilon} \mathcal{D}_1$, where ε is the distinguishing advantage. In the case when ε is negligible, we drop ε from this notation.

We define two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ to be quantum computationally indistinguishable if they cannot be distinguished by QPT distinguishers; we define this formally in the full version. We denote this by $\mathcal{D}_0 \approx_{\mathcal{Q},\varepsilon} \mathcal{D}_1$, where ε is the distinguishing advantage. 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 (classical) 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.

In Section 2.1, we define the notion of bounded concurrent QZK for NP. In Section 2.2, we define the notion of bounded concurrent ZK for QMA. We present the definition of quantum proof of knowledge in Section 2.3.

2.1 Bounded Concurrent QZK for NP

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

Definition 1 (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 1. We will later 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 2 (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 $\left((1,\mathsf{msg}_1),\ldots,\left(Q,\mathsf{msg}_Q\right)\right)$, where msg_i is defined as:

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

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, 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 3 (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 bipartite advice state, ρ_{AB}, on registers A and B, the following holds:*

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

  where V^* and Sim only have access to register A. In other words, only the identity is performed on register B.

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

Definition 4 (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:*

*That is, it has sent $(1,z_1)$ first, then $(2,z_2)$ and so on.

– 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 bipartite advice state,* ρ_{AB}*, on registers* A *and* B*, the following holds:*

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

where V^* *and* Sim *only have access to register* A*. In other words, only the identity is performed on register* B*.*

2.2 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 5 (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 define the notion of Q-sesssion adversaries.

Definition 6 (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 $\left((1, \mathsf{msg}_1), \ldots, (Q, \mathsf{msg}_Q)\right)$, where msg_i is defined as:

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

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 msg_i 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 $\big((1, \mathsf{msg}'_1), \ldots, (Q, \mathsf{msg}'_Q)\big)$. 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 2. To invoke Q different sessions, we assume that the prover has Q copies of the witness state.

Remark 3. 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 7 (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 $\mathcal{A} = \mathcal{A}_{yes} \cup \mathcal{A}_{no}$ 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 $x \in \mathcal{A}_{yes}$ and any witness $|\psi\rangle$, $\mathrm{poly}(\lambda)$-qubit bipartite advice state, ρ_{AB}, on registers A and B, the following holds:*

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

  where V^* and Sim only have access to register A. In other words, only the identity is performed on register B.

2.3 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 8 (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, and quantum state ρ it holds 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:*

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

Moreover, we require $T(\widetilde{\rho}, \widetilde{\rho}') = \mathsf{negl}(|x|)$*, where* $T(\cdot, \cdot)$ *denotes the trace distance and* negl *is a negligible function.*

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

Remark 4 (Comparison with Unruh's Proof of Knowledge [Unr12]). Our definition is a special case of Unruh's quantum proof of knowledge definition. Any proof system satisfying our definition is a quantum proof of knowledge system (according to Unruh's definition) with knowledge error κ, for any κ. Moreover, in Unruh's definition, the extraction probability is allowed to be polynomially related to the acceptance probability whereas in our case, the extraction probability needs to be negligibly close to the acceptance probability.

Definition 9 (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.*

2.4 Intermediate Tool: Quantum Witness-Indistinguishable Proofs for NP

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

Definition 10 (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.

3 Bounded Concurrent QZK for NP

We first give an overview of bounded concurrent QZK for NP.

3.1 Bounded Concurrent QZK for NP

Black Box QZK via Watrous Rewinding. The traditional rewinding technique that has been used to prove powerful results on classical zero-knowledge cannot be easily ported to the quantum setting. The fundamental reason behind this difficulty is the fact that to carry out rewinding, it is necessary to clone the state of the verifier. While cloning comes for free in the classical setting, the no-cloning theorem of quantum mechanics prevents us from being able to clone arbitrary states. Nonetheless, the seminal work of Watrous [Wat09] demonstrates that there are rewinding techniques that are amenable to the quantum setting. Watrous used this technique 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*.
- **No-recording**: Before rewinding any round, the simulator could record (or remember) the transcript generated so far. This recorded transcript along with the rewound transcript will be used for simulation. For instance, in Goldreich and Kahan [GK96], the simulator first commits to garbage values and then waits for the verifier to decommit its challenges. The simulator then records the decommitments before rewinding and then changing its own commitments based on the decommitted values.
  The no-recording condition requires the following to hold: in order for the simulator to rewind from point i to point j ($i > j$), the simulator needs to forget the transcript generated from j^{th} round to the i^{th} round. Note that the simulator of [GK96] does not satisfy the no-recording condition.

Once such a classical simulator is identified, we can then simulate quantum verifiers as follows: run the classical simulator and the quantum verifier† 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

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

†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.

with the decision bits are independent of the verifier's state is used to argue that the state, after measuring the decision register, is essentially not disturbed. Using this fact, we can then reverse the computation and go back to an earlier round. Once the computation is reversed (or rewound to an earlier round), the simulator forgets all the messages exchanged from the point – to which its being rewound to – until the current round.

Incompatibility of Existing Concurrent ZK Techniques. To realize our goal of building bounded concurrent QZK, a natural direction to pursue is to look for classical concurrent ZK protocols with the guarantee that the classical simulator satisfies both the oblivious rewinding and no-recording conditions. However, most known classical concurrent ZK techniques are such that they satisfy one of these two conditions but not both. For example, the seminal work of [PRS02] proposes a concurrent ZK protocol and the simulator they describe satisfies the oblivious rewinding condition but not the no-recording condition. More relevant to our work is the work of Pass et 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 which session verifier sends a message in that round. This means that the probability of whether the simulator rewinds any given round depends on the scheduling of the messages of the verifiers. Unfortunately, the scheduling itself could be a function of the state of the verifier. The malicious verifier could look at the first bit of its auxiliary state. If it is 0, it will ask the first session verifier to send a message and if it is 1, it will ask the second session verifier to send a message and so on. This means that a simulator's decision to rewind could depend on the state of the verifier.

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

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

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

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

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

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

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

The simulator partitions the $\ell_{\mathsf{prot}} \cdot Q$ messages into many blocks with each block being of a fixed size (we discuss the parameters later). The simulator then runs the verifier till the end of first block. At this point, it checks if this block contains a slot. Note that the verifier can stagger the messages of a particular session across the different blocks such that the first message of a slot is in one block but the second message of this slot could be in a different block. The simulator only considers those slots such that both the messages of these slots are contained inside the first block. Let the set of all the slots in the first block be denoted by $\mu(B_1)$, where B_1 denotes the first block. Now, the simulator picks a random slot from the set $\mu(B_1)$. It then checks if this slot is matched or not. That is, it checks if the bit committed in the slot equals the bit sent by the verifier. If indeed they are equal, it continues to the next block, else it rewinds to the beginning of the first block and then executes the first block again. Before rewinding, it forgets the transcript collected in the first block. It repeats this process until the slot it picked is matched. The simulator then moves on to the second block and repeats the entire process. When the simulator needs to compute a witness-indistinguishable proof for a session, it first checks if the fraction of matched slots for that particular session is at least τ. If so, it uses this information 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 a block. Let us now analyze why the oblivious rewinding condition is satisfied. Suppose we are guaranteed that in every block there is at least one slot. Then, we claim that the probability that the simulator rewinds is $\frac{1}{2} \pm \mathsf{negl}(\lambda)$, where negl is a negligible function and λ is the security parameter. This is because the simulator rewinds only if the slot is not matched and the probability that a slot is not matched is precisely $\frac{1}{2} \pm \mathsf{negl}(\lambda)$, from the

hiding property of the commitment scheme. If we can show that every block contains a slot, then the oblivious rewinding condition would also be satisfied.

Absence of Slots and Aborting Issues: We glossed over a couple of issues in the above description. Firstly, the malicious verifier could abort all the sessions in some 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 these blocks and this violates the oblivious rewinding condition: the decision to rewind would be based on whether the verifier aborted or whether there were any slots within a block. In turn, these two conditions could depend on the state of the verifier.

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

Parameters and Analysis: We now discuss the parameters associated with the system. We set the number of slots in the system to be $120Q^7\lambda$. We set τ to be $\lfloor \frac{60Q^7\lambda + Q^4\lambda}{120Q^7\lambda} \rfloor$. We set the number of blocks to be $24Q^6\lambda$. Thus, the size of each block is $\lfloor \frac{120Q^7\lambda}{24Q^6\lambda} \rfloor$. Recall that the reason why we need to set these parameters carefully is to ensure that the malicious prover cannot match more than τ slots with better than negligible probability whereas the simulator can beat this threshold with overwhelming probability.

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

Instead, we argue the following:

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

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

We note that although the simulation strategy of Pass et al. [PTW09] is quite different, their analysis follows the same template as above.

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.

3.2 Construction

We present the construction of quantum zero-knowledge proof system for NP in the bounded concurrent setting in Figure 1. 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, denoted by $(\mathsf{Comm}, \mathsf{R})$.
- Four round quantum witness-indistinguishable proof system Π_{WI} (Definition 10). The relation associated with Π_{WI}, denoted by $\mathcal{R}_{\mathsf{WI}}$, is defined as follows:

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

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 one execution as a slot. So, P and V execute $120Q^7\lambda$ number of slots.

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

Fig. 1. Construction of classical bounded concurrent ZK for NP.

We present the proofs of completeness, soundness and quantum zero-knowledge in the full version.

4 Quantum Proofs of Knowledge

We first present a construction of standalone quantum proof of knowledge for NP. We extend this construction to the bounded concurrent setting in Section 3.1.

4.1 Standalone Quantum Proofs of Knowledge

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

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

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

The security against malicious senders (receiver privacy) states that the sender should not be able to distinguish (with non-negligible probability) whether the receiver's bit is 0 or 1. The security against malicious receivers (also called sender privacy) states that there is a bit b' such that the receiver cannot distinguish (with non-negligible probability) the case when the sender's input is (m_0, m_1) versus the setting when the sender's input is $(m_{b'}, m_{b'})$.

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

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

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

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

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

*For instance, s could be the firt bit of the witness.

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

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

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

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

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

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

*For now, assume that there exists a predicate that can check if s is a valid secret bit.

†We would like to point out that we are designing the standalone PoK protocol as a stepping stone towards the bounded concurrent PoK protocol. If one were to be interested in just the standalone setting, then it might be possible to avoid the subtelties described above by making use of a simulation-secure OT rather than an indistinguishable-secure OT. The reason why we use an indistinguishable-secure OT in the concurrent PoK setting instead of a simulation-secure OT is because we want to avoid using more than one simulator in the analysis; otherwise, we would have multiple simulators trying to rewind the verifier, making the analysis significantly complicated.

the indistinguishability of the consecutive hybrids, $\mathsf{Hyb}_{2.j}$ and $\mathsf{Hyb}_{2.j+1}$, in this sequence turns out to be challenging.

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

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

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

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

Extraction of Multiple Bits. To design a quantum proof of knowledge protocol, we need to be able to extract not just one bit, but multiple bits. To achieve this, we design the prover as follows: on input a witness w, it sequentially executes the above extraction template for each bit of the witness. That is, for every $i \in [\ell_w]$, where ℓ_w is the length of w, it additively secret shares w_i into the shares $(sh_{i,1}, \ldots, sh_{i,\lambda})$. It then invokes $\ell_w \cdot \lambda$ number of OT executions, where in the $(i,j)^{th}$ execution, it chooses the input $(sh_{i,j}, \alpha_{i,j})$ if $b_{i,j} = 0$, or the input $(\alpha_{i,j}, sh_{i,j})$ if $b_{i,j} = 1$, where $\alpha_{i,j}, b_{i,j}$ are sampled uniformly at random. Finally, it uses a QZK protocol to prove that it behaved honestly in the earlier OT executions.

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

4.2 Construction of (Standalone) QZKPoK

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

– A post-quantum statistical receiver-private oblivious transfer protocol, $\Pi_{\mathsf{OT}} = (\mathsf{S}, \mathsf{R})$ satisfying perfect correctness property.
We say that a transcript τ is valid with respect to sender's randomness r and its input bits (m_0, m_1) if τ can be generated with a sender that uses r as randomness for the protocol and uses (m_0, m_1) as inputs.
– A (standalone) QZK proof system Π_{zk} for $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$. We describe the relation $\mathcal{R}(\mathcal{L}_{\mathsf{zk}})$, parameterized by security parameter λ, below.

$$\mathcal{R}\left(\mathcal{L}_{\mathsf{zk}}\right) = \left\{ \left(\left(x, \{\tau_{\mathsf{OT}}^{(i,j)}, b_{i,j}\}_{i\in[\ell_w], j\in[\lambda]} \right); \left(w, \left\{ r_{\mathsf{OT}}^{(i,j)}, sh_{i,j}, \alpha_{i,j} \right\}_{i\in[\ell_w], j\in[\lambda]} \right) \right) : \right.$$

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

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

We describe the construction in Figure 2.

We present the proofs of completeness, quantum proof of knowledge and quantum zero-knowledge in the full version.

4.2.1 Quantum PoK in the Bounded Concurrent Setting Our construction of bounded concurrent quantum proof of knowledge is the same as the one described in Section 4.1, except that we instantiate Π using the bounded concurrent QZK protocol that we constructed in Section 3*.

However, proving the bounded concurrent QZK protocol turns out to be even more challenging than the standalone setting. To grasp the underlying difficulties, let us revisit the proof of QZK in Section 4.1. To prove the indistinguishability of the real and the ideal world, we first simulated the protocol Π. Since we are in the bounded concurrent setting, the simulator of Π is now simultaneously simulating multiple sessions of the verifier. Then using a sequence of intermediate hybrids, we changed the inputs used in the OT executions of all the sessions one at a time. However, in the bounded concurrent setting, the OT messages can be interleaved with QZK messages. This means that the simulator of QZK could be rewinding the OT messages along with the QZK messages. This makes it difficult to invoke the security of OT.

To reduce the indistinguishability of hybrids to breaking OT, we will carefully design the security reduction such that it does not rewind the blocks (the definition of a block is the same as the one described in Section 3.1) containing the

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

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.

Fig. 2. Construction of (standalone) QZKPoK for NP.

messages of the OT protocol. This ensures that we can embed the messages exchanged with the external challenger (in the OT game) without the fear of being rewound. Of course, we need to be cautious: the decision to not rewind a specific block could leak information about the private state of the verifier and this could affect the zero-knowledge propety of the underlying QZK protocol. To overcome this issue, for a block containing the OT messages, we perform a dummy rewind where the transcript of conversation in this block does not change. Thus, we can still interact with the external challenger using the messages in this block. Another issue that arises is that we might end up not rewinding as many blocks as the round complexity of the underlying OT protocol, which is polynomially many rounds. We show that the simulator of the bounded concurrent QZK we constructed in Section 3.1 can be modified in such a way that it can successfully simulate all the sessions even if polynomially many blocks are ignored.

We present the formal details in Section 3.1.

5 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, a commitment scheme and a coin-flipping protocol. We first simplify the protocol of [BJSW16] as follows: their protocol requires security of the coin-flipping protocol to hold against malicious adversaries whereas we only require the security to hold against adversaries who don't deviate from the protocol specification. Once we simplify this step, the resulting protocol will satisfy the property that the QZK simulator only rewinds during the execution of the underlying simulator simulating the QZK protocol for NP. This modification makes it easier for us to extend this protocol to the bounded concurrent setting. We simply instantiate the underlying QZK for NP protocol with its bounded concurrent version.

5.1 Bounded Concurrent QZK for QMA

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

Definition 11 (k-local Clifford Hamiltonian Problem [BJSW16]). *For all $i \in [m]$, let $H_i = C_i|0^{\otimes k}\rangle\langle 0^{\otimes k}|C_i^\dagger$ be a Hamiltonian term on k-qubits where C_i is a Clifford circuit.*

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

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

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

2. *Homomorphic measurements of arbitrary Clifford basis.* For any Clifford circuit C and any state $|\psi\rangle$, a computational basis measurement on $C|\psi\rangle$ can be recovered from a computational basis measurement on $\mathsf{E}_{s'}(C|\psi\rangle)$ along with C and s. Formally, there is a classically efficiently computable function g such that if y is sampled from the distribution induced by measuring the state $\mathsf{E}_{s'}(C|\psi\rangle)$ in the computational basis, then $g(s, C, y)$ is sampled from the distribution induced by measuring the state $C|\psi\rangle$ in the computational basis.
3. *Authentication of measurement outcomes.* For any s and any clifford C, there is a set $\mathcal{S}_{s,C}$ such that for any state $|\psi\rangle$, and any computational basis measurement outcome y performed on $\mathsf{E}_{s'}(C|\psi\rangle)$, it holds that $y \in \mathcal{S}_{s,C}$. Furthermore, for any y, given s and C, it can be efficiently checked whether $y \in \mathcal{S}_{s,C}$.
4. *Simulatability of authenticated states*: there exists an efficient QPT algorithm B such that for any adversary $\mathcal{A}$, every $x \in A_{\mathsf{yes}}$ along with witness $|\psi\rangle$, poly(λ)-qubit advice ρ, the following holds: the probability that $\mathcal{P}(s, C^\dagger_{r^*}, \mathcal{A}(\mathsf{E}_s(|\psi\rangle)))$ outputs 1 is negligibly close to the probability that $\mathcal{P}(s, C^\dagger_{r^*}, \mathcal{A}(B(x, s, r^*)))$ outputs 1, where $\mathcal{P}$ is defined below.

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

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

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

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

Construction. We use the following ingredients in our construction:

- Statistical-binding and quantum-concealing commitment scheme, $(\mathsf{Comm}, \mathsf{R})$.
- Bounded concurrent QZK proof system, denoted by Π_{NP}, for the following language (Section 3.2).

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

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

We describe the construction of bounded concurrent QZK for QMA (with bound Q) in Figure 5.1. We prove the following.

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

Witness: $|\psi\rangle$

- $P \leftrightarrow V$: Prover P samples a secret-key $s \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(k,M)}$, and commits to s using the commitment protocol $(\mathsf{Comm}, \mathsf{R})$. Let $\mathbf{r}$ be the first message of the receiver (sent by V) and $\mathbf{c}$ be the commitment.
  // We call this commitment, the secret-key commitment.
- $P \to V$: P sends $\mathsf{E}_s(|\psi\rangle)$.
- $P \leftrightarrow V$: Prover samples a random string $a \xleftarrow{\$} \{0,1\}^{\log(M)}$, and commits to a using the commitment protocol $(\mathsf{Comm}, \mathsf{R})$. Let $\mathbf{r}'$ be the first message of the receiver and $\mathbf{c}'$ be the commitment.
  // We call this commitment, the coin-flipping commitment.
- $V \to P$: Verifier samples a random string $b \xleftarrow{\$} \{0,1\}^{\log(M)}$. Verifier sends b to the prover.
- $P \to V$: Prover sends $r^* := a \oplus b$ to the verifier.
- Verifier computes $\mathsf{Eval}\left(C_{r^*}^\dagger, \mathsf{E}_s(|\Psi\rangle)\right) \to \mathsf{E}_s(C_{r^*}^\dagger|\psi\rangle)$ and measures in the computational basis. Let y denote the measurement outcome. Verifier sends y to the prover.
- Prover checks that $y \in \mathcal{S}_{s,C_{r^*}^\dagger}$ and that $\mathcal{P}(s, C_{r^*}^\dagger, y) = 1$. If not, it aborts.
- Prover and verifier engage in a QZK protocol for NP, Π_{NP}, for the statement $(\mathbf{r}, \mathbf{c}, \mathbf{r}', \mathbf{c}', r^*, y, b)$ and the witness (s, ℓ, a, ℓ').

Fig. 3. Bounded-Concurrent QZK for QMA

Theorem 4. *Assuming that Π_{NP} satisfies the definition of bounded concurrent QZK for NP, the protocol given in Figure 5.1 is a bounded concurrent QZK protocol for* QMA *with soundness* $\frac{1}{\mathrm{poly}}$.

Remark 5. 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 (Proof Sketch).
Completeness follows from [BJSW16].

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

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

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

Description of Sim: it starts with the registers $\mathbf{X}_{zk}, \mathbf{X}_{anc}, \mathbf{M}, \mathbf{Aux}$. The register $\mathbf{X}_{zk}$ is used by the simulator of the bounded concurrent QZK protocol, $\mathbf{X}_{anc}$ is an ancillary register, $\mathbf{M}$ is used to store the messages exchanged between the simulator and the verifier and finally, the register $\mathbf{Aux}$ is used for storing the private state of the verifier. Initialize the registers $\mathbf{X}_{zk}, \mathbf{M}$ with all zeroes. Initialize the register $\mathbf{X}_{anc}$ with $(\bigotimes_{j=1}^{Q} |s_j\rangle\langle s_j|) \otimes (\bigotimes_{j=1}^{Q} |r_j^*\rangle\langle r_j^*|) \otimes (\bigotimes_{j=1}^{Q} \rho_j) \otimes |0^{\otimes \mathrm{poly}}\rangle\langle 0^{\otimes \mathrm{poly}}|$, where s_i, r_i^* are generated uniformly at random and $\rho_j \leftarrow B(x, s_j, r_j^*)$ is defined in bullet 4 under BJSW encoding.

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

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

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

prover (real world).

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

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

The output distributions of Hyb_1 and Hyb_2 are identical.

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

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

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

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

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

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

$\underline{\mathsf{Hyb}_{6.i} \text{ for } i \in [Q]}$: 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 5.1, include a quantum

proof of classical knowledge system for NP (for instance, the one we constructed in Section 4.2) 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 5.1. To see why this satisfies proof of quantum knowledge, note that an extractor can extract s with probability negligibly close to the acceptance probability and using s, can recover the witness $|\Psi\rangle$.

For the first time, we get proof of quantum knowledge (even in the standalone setting) with $(1-\mathsf{negl})$-quality if the acceptance probability is negligibly close to 1, where the quality denotes the closeness to the witness state. Previous proof of quantum knowledge [BG19, CVZ20] achieved only $1-\frac{1}{\text{poly}}$ qualtiy; this is because these works use Unruh's quantum proof of classical knowledge technique [Unr12] and the extraction probability in Unruh is not negligibly close to the acceptance probability.

Acknowledgements

We thank Abhishek Jain for many enlightening discussions, Zhengzhong Jin for patiently answering questions regarding [GJJM20], Dakshita Khurana for suggestions on constructing oblivious transfer, Ran Canetti for giving an overview of existing classical concurrent ZK techniques, Aram Harrow and Takashi Yamakawa for discussions on the assumption of cloning security (included in a previous version of this paper) and Andrea Coladangelo for clarifications regarding [CVZ20]. RL was funded by NSF grant CCF-1729369. MIT-CTP/5289.

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.

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

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

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

[BS05] Boaz Barak and Amit Sahai. How to play almost any mental game over the net-concurrent composition via super-polynomial simulation. In *46th*

Annual IEEE Symposium on Foundations of Computer Science (FOCS'05), pages 543–552. IEEE, 2005.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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