A Black-Box Approach to Post-Quantum Zero-Knowledge in Constant Rounds

Nai-Hui Chia[1,2](✉), Kai-Min Chung[3], and Takashi Yamakawa[4]

[1] QuICS, University of Maryland, College Park, USA
naichia@iu.edu
[2] Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, USA
[3] Institute of Information Science, Academia Sinica, Taiwan, China
kmchung@iis.sinica.edu.tw
[4] NTT Secure Platform Laboratories, Tokyo, Japan
takashi.yamakawa.ga@hco.ntt.co.jp

Abstract. In a recent seminal work, Bitansky and Shmueli (STOC '20) gave the first construction of a constant round zero-knowledge argument for **NP** secure against quantum attacks. However, their construction has several drawbacks compared to the classical counterparts. Specifically, their construction only achieves computational soundness, requires strong assumptions of quantum hardness of learning with errors (QLWE assumption) and the existence of quantum fully homomorphic encryption (QFHE), and relies on non-black-box simulation.

In this paper, we resolve these issues at the cost of weakening the notion of zero-knowledge to what is called ϵ-zero-knowledge. Concretely, we construct the following protocols:

– We construct a constant round interactive proof for **NP** that satisfies *statistical* soundness and *black-box* ϵ-zero-knowledge against quantum attacks assuming the existence of *collapsing hash functions*, which is a quantum counterpart of collision-resistant hash functions. Interestingly, this construction is just an adapted version of the classical protocol by Goldreich and Kahan (JoC '96) though the proof of ϵ-zero-knowledge property against quantum adversaries requires novel ideas.

– We construct a constant round interactive argument for **NP** that satisfies computational soundness and *black-box* ϵ-zero-knowledge against quantum attacks only assuming the existence of post-quantum one-way functions.

At the heart of our results is a new quantum rewinding technique that enables a simulator to extract a committed message of a malicious verifier while simulating verifier's internal state in an appropriate sense.

1 Introduction

Zero-Knowledge Proof. Zero-knowledge (ZK) proof [GMR89] is a fundamental cryptographic primitive, which enables a prover to convince a verifier of a

T. Malkin and C. Peikert (Eds.): CRYPTO 2021, LNCS 12825, pp. 315–345, 2021.
https://doi.org/10.1007/978-3-030-84242-0_12

statement without giving any additional “knowledge” beyond that the statement is true. In the classical setting, there have been many feasibility results on ZK proofs for specific languages including quadratic residuosity [GMR89], graph isomorphism [GMW91], statistical difference problem [SV03] etc., and for all **NP** languages assuming the existence of one-way functions (OWFs) [GMW91,Blu86]. On the other hand, van de Graaf [Gra97] pointed out that there is a technical difficulty to prove security of these protocols against quantum attacks. Roughly, the difficulty comes from the fact that security proofs of these results are based on a technique called *rewinding*, which cannot be done when an adversary is quantum due to the no-cloning theorem. Watrous [Wat09] considered *post-quantum ZK proof*, which means a classical interactive proof that satisfies (computational) zero-knowledge property against quantum malicious verifiers, and showed that some of the classical constructions above are also post-quantum ZK. Especially, he introduced a new *quantum rewinding technique* which is also applicable to quantum adversaries and proved that 3-coloring protocol of Goldreich, Micali, and Wigderson [GMW91] is secure against quantum attacks assuming that the underlying OWF is post-quantum secure, i.e., uninvertible in quantum polynomial-time (QPT).[1] Since the 3-coloring problem is **NP**-complete, this means that there exists a post-quantum ZK proof for all **NP** languages assuming the existence of post-quantum OWFs.

Round Complexity. An important complexity measure of ZK proofs is *round complexity*, which is the number of interactions between a prover and verifier. In this aspect, the 3-coloring protocol [GMW91] (and its quantumly secure version [Wat09]) is not satisfactory since that requires super-constant number of rounds.[2] Goldreich and Kahan [GK96] gave the first construction of a constant round ZK proof for **NP** assuming the existence of collision-resistant hash function in the classical setting. However, Watrous' rewinding technique does not seem to work for this construction (as explained in Sect. 1.2), and it has been unknown if their protocol is secure against quantum attacks.

Recently, Bitansky and Shmueli [BS20] gave the first construction of post-quantum ZK *argument* [BC90] for **NP**, which is a weakened version of post-quantum ZK proof where soundness holds only against computationally bounded adversaries. In addition to weakening soundness to computational one, there are several drawbacks compared to classical counterparts. First, they assume strong assumptions of quantum hardness of learning with erros (QLWE assumption) [Reg09] and the existence of quantum fully homomorphic encryption (QFHE)

[1] Strictly speaking, Watrous' assumption is a statistically binding and post-quantum computationally hiding commitment scheme, and he did not claim that this can be constructed under the existence of post-quantum OWFs. However, we can see that such a commitment scheme can be obtained by instantiating the construction of [Nao91,HILL99] with a post-quantum OWF.

[2] 3-round suffices for achieving a constant soundness error, but super-constant times sequential repetitions are needed for achieving negligible soundness error (i.e., a cheating prover can let a verifier accept on a false statement only with a negligible probability). Negligible soundness error is a default requirement in this paper.

[Mah18a,Bra18]. Though the QLWE assumption is considered fairly standard due to reductions to worst-case lattice problems [Reg09,Pei09,BLP13], a construction of QFHE requires circular security of an QLWE-based encryption scheme, which has no theoretical evidence. In contrast, a constant round *classical* ZK argument for **NP** is known to exist under the minimal assumption of the existence of OWFs [FS90,PW09]. Second, their security proof of quantum ZK property relies on a novel *non-black-box* simulation technique, which makes use of the actual description of malicious verifier instead of using it as a black-box. In contrast, classical counterparts can be obtained by black-box simulation [FS90,GK96,PW09]. Therefore, it is of theoretical interest to ask if we can achieve constant round quantum ZK by black-box simulation. Third, somewhat related to the second issue, their construction also uses building blocks in a non-black-box manner, which makes the actual efficiency of the protocol far from practical. Again, classical counterparts are known based on black-box constructions [GK96,PW09].

Given the state of affairs, it is natural to ask the following questions:

1. Are there constant round post-quantum ZK proofs for **NP** instead of arguments?
2. Are there constant round post-quantum ZK proofs/arguments for **NP** from weaker assumptions than those in [BS20]?
3. Are there constant round post-quantum ZK proofs/arguments for **NP** based on black-box simulation and/or black-box construction?
4. Are known constructions of constant round classical ZK proofs/arguments for **NP** (e.g., [FS90,GK96,PW09]) secure against quantum attacks if we instantiate them with post-quantum building blocks?

1.1 Our Results

In this work, we partially answer the above questions affirmatively at the cost of weakening the quantum ZK property to *quantum ϵ-ZK*, which is the quantum version of ϵ-ZK introduced in [DNS04].[3]

Quantum ϵ-Zero-Knowledge. The standard quantum ZK property roughly requires that for any QPT V^*, there exists a QPT simulator $\mathcal{S}$ that simulates the interaction between V^* and an honest prover so that the simulation is indistinguishable from the real execution against any QPT distinguishers. On the other hand, in quantum ϵ-ZK, a simulator is allowed to depend on a "accuracy parameter" ϵ. That is, it requires that for any QPT malicious verifier V^* and a noticeable accuracy parameter ϵ, there exists a QPT simulator $\mathcal{S}$ *whose running time polynomially depends on* ϵ^{-1} that simulates the interaction between V^* and an honest prover so that no QPT distinguisher can distinguish it from real execution with advantage larger than ϵ. Though this is a significant relaxation of quantum ZK, this still captures meaningful security. For example, we can see

[3] ϵ-ZK was originally called ϵ-knowledge, but some later works [BKP18,FGJ18] call it ϵ-ZK. We use ϵ-ZK to clarify that this is a variant of ZK.

that quantum ϵ-ZK implies both quantum versions of witness indistinguishability and witness hiding similarly to the analogous claims in the classical setting [BKP19].[4] Moreover, by extending the observation in [DNS04] to the quantum setting, we can see the following: Suppose that a QPT malicious verifier solves some puzzle whose solution is efficiently checkable (e.g., finding a witness of an **NP** statement) after an interaction between an honest prover. Then, quantum ϵ-ZK implies that if the verifier succeeds in solving the puzzle with noticeable probability p after the interaction, then there is a QPT algorithm (whose running time polynomially depends on p^{-1}) that solves the same puzzle with noticeable probability (say, $p/2$) *without interacting with the honest prover.* This captures the naive intuition of the ZK property that "anything that can be done after the execution can be done without execution" in some sense, and this would be sufficient in many cryptographic applications. Thus we believe that quantum ϵ-ZK is conceptually a similar notion to the standard quantum ZK. More discussion on (quantum) ϵ-ZK and other related notions of ZK can be found in Sect. 1.3.

Our Constructions. We give two constructions of constant round quantum ϵ-ZK protocols.

- We construct a constant round quantum ϵ-ZK *proof* for **NP** assuming the existence of *collapsing hash functions* [Unr16b, Unr16a], which is considered as a counterpart of collision-resistant hash functions in the quantum setting. Especially, we can instantiate the construction based on the QLWE assumption. Our construction is fully black-box in the sense that both simulation and construction rely on black-box usage of building blocks and a malicious verifier. Interestingly, this construction is just an adapted version of the classical protocol of [GK96] though the proof of quantum ϵ-zero-knowledge property requires novel ideas.
- We construct a constant round quantum ϵ-ZK argument for **NP** assuming the minimal assumption of the existence of *post-quantum OWFs.* This construction relies on black-box simulation, but the construction itself is non-black-box.

At the heart of our results is a new quantum rewinding technique that enables a simulator to extract a committed message of a malicious verifier while simulating verifier's internal state in some sense. We formalize this technique as an *extraction lemma*, which we believe is of independent interest.

1.2 Technical Overview

Though we prove a general lemma which we call extraction lemma (Lemma 3.1) and then prove quantum ϵ-ZK of our constructions based on that in the main body, we directly explain the proof of quantum ϵ-ZK without going through such an abstraction in this overview.

[4] Actually, [BKP19] shows that even weaker notion called *weak ZK* suffices for witness indistinguishability and witness hiding. See also Sect. 1.3.

Known Classical Technique and Difficulty in Quantum Setting. First, we review a classical constant round ZK proof by Goldreich and Kahan [GK96] (referred to as GK protocol in the following), and explain why it is difficult to prove quantum ZK for this protocol by known techniques. GK protocol is based on a special type of 3-round proof system called Σ-protocol.[5] In a Σ-protocol, a prover sends the first message a, a verifier sends the second message e referred to as a *challenge*, which is just a public randomness, and the prover sends the third message z. A Σ-protocol satisfies a special type of honest-verifier ZK, which ensures that if a challenge e is fixed, then one can simulate the transcript (a, e, z) without using a witness. Though this may sound like almost the standard ZK property, a difficulty when proving ZK is that a malicious verifier may *adaptively* choose e depending on a, and thus we cannot fix e at the beginning. To resolve this issue, the idea of GK protocol is to let the verifier commit to a challenge e at the beginning of the protocol. That is, GK protocol roughly proceeds as follows:[6]

1. A verifier sends a commitment com to a challenge e of a Σ-protocol.
2. The prover sends the first message a of the Σ-protocol.
3. The verifier opens com to open a challenge e and its opening information r (i.e., the randomness used for the commitment).
4. The prover aborts if the verifier's opening is invalid. Otherwise it sends the third message z of the Σ-protocol.

When proving the ZK property of GK protocol, they rely on a *rewinding* argument. That is, a simulator first runs the protocol with a malicious verifier until Step 3 to extract a committed message e inside com, and then rewind the verifier's state back to just after Step 1, and then simulates the transcript by using the extracted knowledge of e.

On the other hand, this strategy does not work if we consider a quantum malicious verifier since a quantum malicious verifier may perform measurements in Step 3, which is in general not reversible. In other words, since we cannot copy the verifier's internal state after Step 1 due to the no-cloning theorem, we cannot recover that state after running the protocol until Step 3.

Watrous [Wat09] proved that we can apply a rewinding argument for quantum verifiers under a certain condition. Roughly speaking, the condition is that there is a simulator that succeeds in simulation for quantum verifiers with a fixed (verifier-independent) and noticeable probability. For example, if the challenge space is polynomial size, then a simulator that simply guesses a challenge e suffices. However, for achieving negligible soundness error, the challenge space should be super-polynomial size, in which case it seems difficult to construct such a simulator. Also, relaxing quantum ZK to quantum ϵ-ZK does not seem to resolve the issue in any obvious way.

[5] In this paper, we use Σ-protocol to mean a parallel repetition version where soundness error is reduced to negligible.

[6] We note that this construction is based on an earlier work of [BCY91].

Quantum Analysis of GK Protocol. In spite of the above mentioned difficulty, we succeed in proving quantum ϵ-ZK for a slight variant of GK protocol. In the following, we explain the idea for our results.

Simplified Goal: Simulation of Non-Aborting Case. First, we apply a general trick introduced in [BS20], which simplifies the task of proving quantum ZK. In GK protocol, we say that a verifier aborts if it fails to provide a valid opening to com in Step 3. Then, for proving quantum ZK of the protocol, it suffices to construct two simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ that work only when the verifier aborts and does not abort and they do not change the probability that the verifier aborts too much, respectively. The reason is that if we randomly choose either of these two simulators and just run the chosen one, then the simulation succeeds with probability 1/2 since the guess of if the verifier aborts is correct with probability 1/2. Then, we can apply Watrous' rewinding technique to convert it to a full-fledged simulator. Essentially the same trick also works for quantum ϵ-ZK.

Moreover, it is easy to construct $\mathsf{Sim}_{\mathsf{a}}$ because the first message of a Σ-protocol can be simulated without witness, and one need not provide the third message to the verifier when it aborts. Therefore, the problem boils down to constructing a simulator $\mathsf{Sim}_{\mathsf{na}}$ that works only when the verifier does not abort.

Initial Observations. For explaining how to construct $\mathsf{Sim}_{\mathsf{na}}$, we start by considering the simplest case where a verifier never aborts. Moreover, suppose that the commitment scheme used for committing to a challenge e satisfies the strict-binding property [Unr12], i.e., for any commitment com, there is at most one valid message and randomness. Then, a rewinding strategy similar to the classical case works since, in this case, the verifier's message in Step 3 is information-theoretically determined, and such a deterministic computation does not collapse a quantum state in general.[7] However, for ensuring statistical soundness, we have to use a statistically hiding commitment, which cannot be strict-binding. Fortunately, this problem can be resolved by using *collapse-binding* commitments [Unr16b], which roughly behave similarly to strict-binding commitments for any *computationally bounded* adversaries.[8] Since this is rather a standard technique, in the rest of this overview, we treat the commitment as if it satisfies the strict-binding property.

Next, we consider another toy example where a verifier sometimes aborts. Suppose that a malicious verifier V^* is given an initial state $\frac{1}{\sqrt{2}}(|\psi_{\mathsf{a}}\rangle + |\psi_{\mathsf{na}}\rangle)$ in its internal register $\mathbf{V}$ where $|\psi_{\mathsf{a}}\rangle$ and $|\psi_{\mathsf{na}}\rangle$ are orthogonal, and runs as follows:

1. V^* randomly picks e, honestly generates a commitment com to e, and sends it to the prover (just ignoring the initial state).

[7] This is also observed in [BS20].

[8] Strictly speaking, we need to use a slightly stronger variant of collapse-binding commitments which we call *strong collapse-binding* commitments. Such commitments can be constructed under the QLWE assumption or the existence of collapsing hash functions in more general. See Sect. 2.2 for more details.

2. After receiving a, V^* performs a projective measurement $\{|\psi_{\mathsf{a}}\rangle\langle\psi_{\mathsf{a}}|, I - |\psi_{\mathsf{a}}\rangle\langle\psi_{\mathsf{a}}|\}$ on $\mathbf{V}$, and immediately aborts if $|\psi_{\mathsf{a}}\rangle\langle\psi_{\mathsf{a}}|$ is applied, and otherwise honestly opens (e, r).
3. After completing the protocol, V^* outputs its internal state in $\mathbf{V}$.

It is trivial to construct a simulator for this particular V^* since it just ignores prover's messages. But for explaining our main idea, we examine what happens if we apply the same rewinding strategy as the classical case to the above verifier. After getting a commitment com from V^*, a simulator sends a random a to V^* to extract e. Since we are interested in constructing a simulator that works in the non-aborting case, suppose that V^* does not abort, i.e., sends back a valid opening (e, r). At this point, V^*'s internal state collapses to $|\psi_{\mathsf{na}}\rangle$. Then the simulator cannot "rewind" this state to the original verifier's state $\frac{1}{\sqrt{2}}(|\psi_{\mathsf{a}}\rangle + |\psi_{\mathsf{na}}\rangle)$ in general, and thus the simulation seems to get stuck. However, our key observation is that, conditioned on that V^* does not abort, V^*'s state always collapses to $|\psi_{\mathsf{na}}\rangle$ even in the real execution. Since our goal is to construct $\mathsf{Sim}_{\mathsf{na}}$ that is only required to work for the non-aborting case, it does not matter if V^*'s state collapses to $|\psi_{\mathsf{na}}\rangle$ when the simulator runs extraction. More generally, extraction procedure may collapse verifier's internal state if a similar collapsing happens even in the real execution conditioned on that the verifier does not abort.

Our Idea: Decompose Verifier's Space. To generalize the above idea, we want to decompose verifier's internal state after Step 1 into *aborting part* and *non-aborting part*. However, the definition of such a decomposition is non-trivial since a verifier may determine if it aborts depending on the prover's message a in addition to its internal state. Therefore, instead of decomposing it into always-aborting part and always-non-aborting part as in the example of the previous paragraph, we set a noticeable threshold t and decompose it into "not-abort-with-probability $< t$ part" and "not-abort-with-probability $\geq t$ part" over the randomness of a.

For implementing this idea, we rely on Jordan's lemma (e.g., see a lecture note by Regev [AR06]) in a similar way to the work by Nagaj, Wocjan, and Zhang [NWZ09] on the amplification theorem for **QMA**. Let Π be a projection that corresponds to "Step 2 + Step 3 + Check if the verifier does not abort" in GK protocol. A little bit more formally, let $\mathbf{V}$ be a register for verifier's internal state and $\mathbf{Aux}$ be an auxiliary register. Then Π is a projection over $\mathbf{V} \otimes \mathbf{Aux}$ that works as follows:

1. Apply a unitary U_{aux} over $\mathbf{Aux}$ that maps $|0\rangle_{\mathbf{Aux}}$ to $\frac{1}{\sqrt{|\mathcal{R}|}}\sum_{\mathsf{rand}\in\mathcal{R}} |\mathsf{rand}, a_{\mathsf{rand}}\rangle_{\mathbf{Aux}}$ where $\mathcal{R}$ is the randomness space to generate the first message of the Σ-protocol and a_{rand} is the first message derived from the randomness rand.[9]

[9] $\mathbf{Aux}$ stores multiple qubits, but we denote by $|0\rangle_{\mathbf{Aux}}$ to mean $|0^\ell\rangle_{\mathbf{Aux}}$ for the appropriate length ℓ for notational simplicity.

2. Apply a unitary U_V that corresponds to Step 3 for prover's message a_{rand} in **Aux** except for measurement,
3. Apply a projection to the subspace spanned by states that contain valid opening (e, r) for com in designated output registers,
4. Apply $(U_V U_{\mathsf{aux}})^\dagger$.

One can see that the probability that the verifier does not abort (i.e., sends a valid opening) is $\|\Pi \left|\psi\right\rangle_{\mathbf{V}} \left|0\right\rangle_{\mathbf{Aux}}\|^2$ where $\left|\psi\right\rangle_{\mathbf{V}}$ is verifier's internal state after Step 1. Then Jordan's lemma gives an orthogonal decomposition of the Hilbert space of $\mathbf{V} \otimes \mathbf{Aux}$ into many one- or two-dimensional subspaces $S_1, ..., S_N$ that are invariant under Π and $\left|0\right\rangle_{\mathbf{Aux}} \left\langle 0\right|_{\mathbf{Aux}}$ such that we have the following:

1. For any $j \in [N]$ and $\left|\psi_j\right\rangle_{\mathbf{V}} \left|0\right\rangle_{\mathbf{Aux}} \in S_j$, the projection Π succeeds with probability p_j, i.e., $\|\Pi \left|\psi_j\right\rangle_{\mathbf{V}} \left|0\right\rangle_{\mathbf{Aux}}\|^2 = p_j$.
2. A success probability of projection Π is "amplifiable" in each subspace. That is, there is an "amplification procedure" Amp that maps any $\left|\psi_j\right\rangle_{\mathbf{V}} \left|0\right\rangle_{\mathbf{Aux}} \in S_j$ to $\Pi \left|\psi_j\right\rangle_{\mathbf{V}} \left|0\right\rangle_{\mathbf{Aux}}$ with overwhelming probability within $\mathsf{poly}(\lambda, p_j^{-1})$ times iteration of the same procedure (that does not depend on j) for any $j \in [N]$. Moreover, this procedure does not cause any interference between different subspaces.

Then we define two subspaces

$$S_{<t} := \bigoplus_{j:p_j<t} S_j, \quad S_{\geq t} := \bigoplus_{j:p_j \geq t} S_j.$$

Then for any $\left|\psi\right\rangle_{\mathbf{V}}$, we can decompose it as

$$\left|\psi\right\rangle_{\mathbf{V}} = \left|\psi_{<t}\right\rangle_{\mathbf{V}} + \left|\psi_{\geq t}\right\rangle_{\mathbf{V}}$$

by using (sub-normalized) states $\left|\psi_{<t}\right\rangle_{\mathbf{V}}$ and $\left|\psi_{\geq t}\right\rangle_{\mathbf{V}}$ such that $\left|\psi_{<t}\right\rangle_{\mathbf{V}} \left|0\right\rangle_{\mathbf{Aux}} \in S_{<t}$ and $\left|\psi_{\geq t}\right\rangle_{\mathbf{V}} \left|0\right\rangle_{\mathbf{Aux}} \in S_{\geq t}$. In this way, we can formally define a decomposition of verifier's internal state into "not-abort-with-probability $<t$ part" and "not-abort-with-probability $\geq t$ part".

Extraction and Simulation. Then we explain how we can use the above decomposition to implement extraction of e for simulation of non-aborting case. First, we consider an easier case where the verifier's state after Step 1 only has $S_{\geq t}$ component $\left|\psi_{\geq t}\right\rangle_{\mathbf{V}}$. In this case, we can use Amp to map $\left|\psi_{\geq t}\right\rangle_{\mathbf{V}} \left|0\right\rangle_{\mathbf{Aux}}$ onto the span of Π within $\mathsf{poly}(\lambda, t^{-1})$ times iteration. After mapped to Π, we can extract (e, r) without collapsing the state by the definition of Π and our assumption that the commitment is strict-binding. This means that given $\left|\psi_{\geq t}\right\rangle_{\mathbf{V}}$, we can extract (e, r), which is information theoretically determined by com, with overwhelming probability. In general, such a deterministic computation can be implemented in a reversible manner, and thus we can extract (e, r) from $\left|\psi_{\geq t}\right\rangle_{\mathbf{V}}$ almost without damaging the state.

On the other hand, the same procedure does not work for $\left|\psi_{<t}\right\rangle_{\mathbf{V}}$ since $\mathsf{poly}(\lambda, t^{-1})$ times iteration is not sufficient for amplifying the success probability of Π to overwhelming in this subspace. Our idea is to let a simulator run

the above extraction procedure in superposition even though $S_{<t}$ component may be damaged.

Specifically, our extraction procedure Ext works as follows:

1. Given a verifier's internal state $|\psi\rangle_{\mathbf{V}}$ after Step 1, initialize $\mathbf{Aux}$ to $|0\rangle_{\mathbf{Aux}}$ and runs Amp for $\mathsf{poly}(\lambda, t^{-1})$ times iteration. Abort if a mapping onto Π does not succeed. Otherwise, proceed to the next step.
2. Apply $U_V U_{\mathsf{aux}}$, measure designated output registers to obtain $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$, and apply $(U_V U_{\mathsf{aux}})^{\dagger}$. We note that $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ is always a valid opening of com since Ext runs this step only if it succeeds in mapping the state onto Π in the previous step. We also note that this step does not collapse the state at all by the strict-binding property of the commitment.
3. Uncompute Step 1 and measure $\mathbf{Aux}$. Abort if the measurement outcome is not 0. Otherwise, proceed to the next step.
4. Output the extracted opening $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ along with a "post-extraction state" $|\psi'\rangle_{\mathbf{V}}$ in register $\mathbf{V}$. For convenience, we express $|\psi'\rangle_{\mathbf{V}}$ as a sub-normalized state whose norm is the probability that Ext does not abort and the post-extraction state conditioned on that the extraction succeeds is $\frac{|\psi'\rangle_{\mathbf{V}}}{\||\psi'\rangle_{\mathbf{V}}\|}$.

In the following, we analyze Ext. We consider the decomposition of $|\psi\rangle_{\mathbf{V}}$ as defined in the previous paragraph:

$$|\psi\rangle_{\mathbf{V}} = |\psi_{<t}\rangle_{\mathbf{V}} + |\psi_{\geq t}\rangle_{\mathbf{V}}\,.$$

Suppose that Ext does not abort, i.e., it outputs a valid opening $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ along with a post-extraction state $|\psi'\rangle_{\mathbf{V}}$. Then, $|\psi'\rangle_{\mathbf{V}}$ can be expressed as

$$|\psi'\rangle_{\mathbf{V}} = |\psi'_{<t}\rangle_{\mathbf{V}} + |\psi'_{\geq t}\rangle_{\mathbf{V}}$$

for some $|\psi'_{<t}\rangle_{\mathbf{V}}$ and $|\psi'_{\geq t}\rangle_{\mathbf{V}}$ such that $|\psi'_{<t}\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}} \in S_{<t}$, $|\psi'_{\geq t}\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}} \in S_{\geq t}$, and $|\psi_{\geq t}\rangle_{\mathbf{V}} \approx |\psi'_{\geq t}\rangle_{\mathbf{V}}$ since there is no interference between $S_{<t}$ and $S_{\geq t}$ when running Amp and $S_{\geq t}$ component hardly changes as observed above. This is not even a close state to the original state $|\psi\rangle_{\mathbf{V}}$ in general since the $S_{<t}$ component may be completely different. However, our key observation is that, conditioned on that the verifier does not abort, at most "t-fraction" of $S_{<t}$ component survives even in the real execution by the definition of the subspace $S_{<t}$. That is, in the verifier's final output state conditioned on that it does not abort, the average squared norm of a portion that comes from $S_{<t}$ component is at most t. Thus, even if a simulator fails to simulate this portion, this only impacts the accuracy of the simulation by a certain function of t, which is shown to be $O(t^{1/3})$ in the main body.

With this observation in mind, the non-aborting case simulator $\mathsf{Sim}_{\mathsf{na}}$ works as follows.

1. Run Step 1 of the verifier to obtain com and let $|\psi\rangle_{\mathbf{V}}$ be verifier's internal state at this point.

2. Run Ext on input $|\psi\rangle_{\mathbf{V}}$. Abort if Ext aborts. Otherwise, obtain an extracted opening $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ and a post-extraction state $|\psi'\rangle_{\mathbf{V}}$, and proceed to the next step.
3. Simulate a transcript (a, e_{Ext}, z) by the honest-verifier ZK property of the Σ-protocol.
4. Send a to the verifier whose internal state is replaced with $|\psi'\rangle_{\mathbf{V}}$. Let (e, r) be the verifier's response. Abort if (e, r) is not a valid opening to com. Otherwise send z to the verifier.
5. Output the verifier's final output.

By the above analysis, we can see that $\mathsf{Sim}_{\mathsf{na}}$'s output distribution is close to the real verifier's output distribution with an approximation error $O(t^{1/3})$ conditioned on that the verifier does not abort. Furthermore, the probability that the verifier does not abort can only be changed by at most $O(t^{1/3})$. If we could set t to be a negligible function, then we would be able to achieve quantum ZK rather than quantum ϵ-ZK. However, since we have to ensure that Amp's running time $\mathsf{poly}(\lambda, t^{-1})$ is polynomial in λ, we can only set t to be noticeable. Since we can set t to be an arbitrarily small noticeable function, we can make the approximation error $O(t^{1/3})$ be an arbitrarily small noticeable function. This means that the protocol satisfies quantum ϵ-ZK.

Black-Box Simulation. So far, we did not pay attention to the black-box property of simulation. We briefly explain the definition of black-box quantum ZK and that our simulator satisfies it. First, we define black-box quantum ZK by borrowing the definition of quantum oracle machine by Unruh [Unr12]. Roughly, we say that a simulator is black-box if it only accesses unitary part of a verifier and its inverse in a black-box manner, and does not directly act on the verifier's internal registers. With this definition, one part where it is unclear if our simulator is black-box is the amplification procedure Amp. However, by a close inspection, we can see that Amp actually just performs sequential measurements $\{\Pi, I_{\mathbf{V},\mathbf{Aux}} - \Pi\}$ and $\{|0\rangle_{\mathbf{Aux}} \langle 0|_{\mathbf{Aux}}, I_{\mathbf{V},\mathbf{Aux}} - |0\rangle_{\mathbf{Aux}} \langle 0|_{\mathbf{Aux}}\}$, which can be done by black-box access to the verifier as seen from the definition of Π. Therefore, we can see that our simulator is black-box.

A Remark on Underlying Σ-Protocol. In the original GK protocol, any Σ-Protocol can be used as a building block. However, in our technique, we need to use *delayed-witness* Σ-protocol where the first message a can be generated without knowledge of a witness due to a technical reason. An example of delayed-witness Σ-protocol is Blum's Graph Hamiltonicity protocol [Blu86]. Roughly, the reason to require this additional property is for ensuring that a simulator can perfectly simulate the first message a of the Σ-protocol when running the extraction procedure. In the classical setting, a computationally indistinguishable simulation of a works, but we could not prove an analogous claim in our setting.

OWF-Based Construction. Next, we briefly explain our OWF-based quantum ϵ-ZK argument. The reason why we need a stronger assumption in our first construction is that we need to implement the commitment for the challenge by a constant round statistically hiding commitment, which is not known to exist from OWF. Then, a natural idea is to relax it to computationally hiding one if we only need computational soundness. We can show that the extraction technique as explained above also works for statistically binding commitments with a small tweak. However, we cannot prove soundness of the protocol without any modification due to a malleability issue. For explaining this, we recall that the first message a of a Σ-protocol itself is also implemented as a commitment. Then, the computational hiding of commitment does not prevent a computationally bounded prover, which is given a commitment com to e, from generating a "commitment" a whose committed message depends on e. Such a dependence leads to an attack against soundness. To prevent this, an extractable commitment scheme is used to generate a in the classical setting [PW09]. However, since it is unclear if the extractable commitment scheme used in [PW09] is secure against quantum adversaries, we take an alternative approach that we let a prover prove that it knows a committed message inside a by using a proof of knowledge before a verifier opens a challenge as is done in [Gol01, Sec.4.9], [Gol04, App.C.3]. A naive approach to implement this idea would be to use ZK proof of knowledge, but this does not work since a constant round ZK argument is what we are trying to construct. Fortunately, we can instead use witness indistinguishable proof of knowledge (WIPoK) with a simple OR proof trick. Specifically, we let a prover prove that "I know committed message in a" OR "I know witness w for x" where x is the statement being proven in the protocol. In the proof of soundness, since we assume x is a false statement, a witness for the latter statement does not exist. Then we can extract a committed message inside a to break the hiding property of the commitment scheme used by the verifier if the committed message depends on e. On the other hand, in the proof of ϵ-ZK property, we can use the real witness w in an intermediate hybrid to simulate WIPoK without using knowledge of a committed message. In such a hybrid, we can rely on honest-verifier ZK of the Σ-protocol to change a to a simulated one for an extracted challenge e.

Finally, we remark that though we are not aware of any work that explicitly claims the existence of a constant round WIPoK that works for quantum provers from OWFs, we observe that a combination of known works easily yields such a construction. (See the full version for more details.) As a result, we obtain constant round quantum ϵ-ZK argument from OWFs.

1.3 Related Work

ϵ-Zero-Knowledge and Related Notions. Though we are the first to consider ϵ-ZK in the quantum setting, there are several works that consider ϵ-ZK in the classical setting. We briefly review them. We note that all of these results are in the classical setting, and it is unknown if similar results hold in the quantum setting. The notion of ϵ-ZK (originally called ϵ-knowledge) was introduced by Dwork,

Naor, and Sahai [DNS04] in the context of concurrent ZK proofs. Bitansky, Kalai, and Paneth [BKP18] gave a construction of 4-round ϵ-ZK proof for **NP** assuming the existence of key-less multi-collision resistant hash function.[10] Barak and Lindell [BL02] showed the impossibility of constant round black-box ZK proof with strict-polynomial time simulation, and observed that strict-polynomial time simulation is possible if we relax ZK to ϵ-ZK. This can be understood as a theoretical separation between ZK and ϵ-ZK. On the other hand, Fleischhacker, Goyal, and Jain [FGJ18] showed that there does not exist 3-round ϵ-ZK proof for **NP** even with non-black-box simulation under some computational assumptions, which is the same lower bound as that for ZK proofs if we allow non-black-box simulation.

Another relaxation of ZK is *super-polynomial simulation (SPS)-ZK* [Pas03], where a simulator is allowed to run in super-polynomial time. One may find a similarity between ϵ-ZK and SPS-ZK in the sense that the latter can be seen as a variant of ϵ-ZK where we set the accuracy parameter ϵ to be negligible. On the other hand, it has been considered that ϵ-ZK is much more difficult to achieve than SPS-ZK. For example, the work of Bitansky, Khurana, and Paneth [BKP19] gave a construction of a 2-round argument for **NP** that achieves a weaker notion of ZK than ϵ-ZK, and the result is considered a significant breakthrough in the area even though there is a simple construction of 2-round SPS-ZK argument for **NP** [Pas03].

Several works considered other weakened notions of ZK [DNRS03,BP12, CLP15,JKKR17,BKP19]. Some of them are weaker than ϵ-ZK, and others are incomparable. For example, "weak ZK" in [BP12,CLP15] is incomparable to ϵ-ZK whereas "weak ZK" in [BKP19] is weaker than ϵ-ZK.

Post-Quantum Zero-Knowledge with Classical Computational Soundness. Ananth and La Placa [AL20] gave a construction of post-quantum ZK argument for **NP** with *classical* computational soundness assuming the QLWE assumption. Though such a protocol would be easy to obtain if we assume average-case classical hardness of certain problems in **BQP** (e.g., factoring) in addition to the QLWE assumption, what is interesting in [AL20] is that they only assume the QLWE assumption.

Post-Quantum Zero-Knowledge with Trusted Setup. Several works studied (non-interactive) post-quantum ZK proofs for **NP** in the common random/reference string model [Kob03,DFS04,PS19]. Among them, Peikert and Shiehian [PS19] proved that there exists non-interactive post-quantum ZK proof for **NP** in the common reference string model assuming the QLWE assumption.[11]

[10] The protocol achieves full-fledged ZK if we allow the simulator to take non-uniform advice or assume a super-polynomial assumption.

[11] In [PS19], they do not explicitly claim ZK against quantum adversaries. However, since their security proof does not rely on rewinding, it immediately extends to post-quantum security if we assume the underlying assumption against quantum adversaries.

Zero-Knowledge for **QMA**. The complexity class **QMA** is a quantum analogue of **NP**. Broadbent, Ji, Song, and Watrous [BJSW20] gave a construction of a ZK proof for **QMA**. Recently, Broadbent and Grilo [BG20] gave an alternative simpler construction of a ZK proof for **QMA**. Bitansky and Shmueli [BS20] gave a constant round ZK argument for **QMA** by combining the construction of [BG20] and their post-quantum ZK argument for **NP**. We believe that our technique can be used to construct a constant round ϵ-ZK proof for **QMA** by replacing the delayed-witness Σ-protocol for **NP** with the delayed-witness quantum Σ-protocol for **QMA** recently proposed by Brakerski and Yuen [BY20].[12] This is beyond the scope of this paper, and we leave a formal proof as a future work.

Several works studied non-interactive ZK proofs/arguments for **QMA** in preprocessing models [CVZ20, BG20, Shm20, ACGH20].

Collapsing Hash Functions. The notion of collapsing hash functions was introduced by Unruh [Unr16b] for a replacement of collision-resistant hash functions in post-quantum setting. Unruh [Unr16a] gave a construction of a collapsing hash function under the QLWE assumption. Actually, the construction is generic based on any lossy function with sufficiently large "lossy rate".[13] Currently, we are not aware of any other construction of collapsing hash function based on standard assumptions, but any new construction of collapsing hash function yields a new instantiation of our first construction.

Zhandry [Zha19] proved that any collision-resistant hash function that is not collapsing yields a stronger variant of public-key quantum money (with infinitely often security). Given the difficulty of constructing public key quantum money, he suggested that most natural post-quantum collision-resistant hash functions are likely already collapsing.

Relation to [CCY20]. Our idea of decomposing a verifier's internal space into "aborting space" and "non-aborting space" is inspired by a recent work of Chia, Chung, and Yamakawa [CCY20]. In [CCY20], the authors consider a decomposition of a prover's internal space into "know-answer space" and "not-know-answer space" to prove soundness of parallel repetition version of Mahadev's classical verification of quantum computation protocol [Mah18b]. Though the conceptual idea and some technical tools are similar, the ways of applying them to actual problems are quite different. For example, in our case, we need a careful analysis to make sure that a post-extraction state is close to the original one in some sense while such an argument does not appear in their work since their goal is proving soundness rather than ZK. On the other hand, their technical core is a approximated projection to each subspace, which is not needed in this paper.

Subsequent work. Subsequently to this work, Chia, Chung, Liu, and Yamakawa [CCLY21] proved that there does not exist a constant round post-quantum ZK

[12] Actually, their protocol is delayed-input, i.e., the first message generation does not use the statement either.

[13] A lossy function is defined similarly to a lossy trapdoor function [PW08] except that we do not require the existence of trapdoor.

argument for **NP** unless **NP** $\in$ **BQP**, which is highly unlikely. This justifies the relaxation to ϵ-ZK in our constructions.

2 Preliminaries

Basic Notations. We use λ to denote the security parameter throughout the paper. For a positive integer $n \in \mathbb{N}$, $[n]$ denotes a set $\{1, 2, ..., n\}$. For a finite set $\mathcal{X}$, $x \overset{\$}{\leftarrow} \mathcal{X}$ means that x is uniformly chosen from $\mathcal{X}$. A function $f : \mathbb{N} \rightarrow [0, 1]$ is said to be negligible if for all polynomial p and sufficiently large $\lambda \in \mathbb{N}$, we have $f(\lambda) < 1/p(\lambda)$, said to be overwhelming if $1 - f$ is negligible, and said to be noticeable if there is a polynomial p such that we have $f(\lambda) \geq 1/p(\lambda)$ for sufficiently large $\lambda \in \mathbb{N}$. We denote by poly an unspecified polynomial and by negl an unspecified negligible function. We use PPT and QPT to mean (classical) probabilistic polynomial time and quantum polynomial time, respectively. For a classical probabilistic or quantum algorithm $\mathcal{A}$, $y \overset{\$}{\leftarrow} \mathcal{A}(x)$ means that $\mathcal{A}$ is run on input x and outputs y. When $\mathcal{A}$ is classical probabilistic algorithm, we denote by $\mathcal{A}(x; r)$ to mean the execution of $\mathcal{A}$ on input x and a randomness r. When $\mathcal{A}$ is a quantum algorithm that takes a quantum advice, we denote by $\mathcal{A}(x; \rho)$ to mean the execution of $\mathcal{A}$ on input x and an advice ρ. For a quantum algorithm $\mathcal{A}$, a unitary part of $\mathcal{A}$ means the unitary obtained by deferring all measurements by $\mathcal{A}$ and omitting these measurements. We use the bold font (like $\mathbf{X}$) to denote quantum registers, and $\mathcal{H}_{\mathbf{X}}$ to mean the Hilbert space corresponding to the register $\mathbf{X}$. For a quantum state ρ, $M_{\mathbf{X}} \circ \rho$ means a measurement in the computational basis on the register $\mathbf{X}$ of ρ. For quantum states ρ and ρ', $\mathsf{TD}(\rho, \rho')$ denotes trace distance between them. When we consider a sequence $\{X_\lambda\}_{\lambda \in \mathbb{N}}$ of some objects (e.g., bit strings, quantum states, sets, Hilbert spaces etc.) indexed by the security parameter λ, we often simply write X to mean X_λ or $\{X_\lambda\}_{\lambda \in \mathbb{N}}$, which will be clear from the context. Similarly, for a function f in the security parameter λ, we often simply write f to mean $f(\lambda)$.

Standard Computational Models

- A PPT algorithm is a probabilistic polynomial time (classical) Turing machine. A PPT algorithm is also often seen as a sequence of uniform polynomial-size circuits.
- A QPT algorithm is a polynomial time quantum Turing machine. A QPT algorithm is also often seen as a sequence of uniform polynomial-size quantum circuits.
- An adversary (or malicious party) is modeled as a non-uniform QPT algorithm $\mathcal{A}$ (with quantum advice) that is specified by sequences of polynomial-size quantum circuits $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ and polynomial-size quantum advice $\{\rho_\lambda\}_{\lambda \in \mathbb{N}}$. When $\mathcal{A}$ takes an input of λ-bit, $\mathcal{A}$ runs $\mathcal{A}_\lambda$ taking ρ_λ as an advice.

Interactive Quantum Machine and Oracle-Aided Quantum Machine. We rely on the definition of an interactive quantum machine and oracle-aided quantum machine that is given oracle access to an interactive quantum machine following [Unr12]. Roughly, an interactive quantum machine $\mathcal{A}$ is formalized by a unitary over registers $\mathbf{M}$ for receiving and sending messages and $\mathbf{A}$ for maintaining $\mathcal{A}$'s internal state. For two interactive quantum machines $\mathcal{A}$ and $\mathcal{B}$ that share the same message register $\mathbf{M}$, an interaction between $\mathcal{A}$ and $\mathcal{B}$ proceeds by alternating invocations of $\mathcal{A}$ and $\mathcal{B}$ while exchanging messages over $\mathbf{M}$.

An oracle-aided quantum machine $\mathcal{S}$ given oracle access to an interactive quantum machine $\mathcal{A}$ with an initial internal state ρ (denoted by $\mathcal{S}^{\mathcal{A}(\rho)}$) is allowed to apply unitary part of $\mathcal{A}$ and its inverse in a black-box manner where $\mathcal{S}$ can act on $\mathcal{A}$'s internal register $\mathbf{A}$ only through oracle access. We refer to [Unr12] for more formal definitions of interactive quantum machines and black-box access to them.

Indistinguishability of Quantum States. We define computational and statistical indistinguishability of quantum states similarly to [BS20].

We may consider random variables over bit strings or over quantum states. This will be clear from the context. For ensembles of random variables $\mathcal{X} = \{X_i\}_{\lambda\in\mathbb{N},i\in I_\lambda}$ and $\mathcal{Y} = \{Y_i\}_{\lambda\in\mathbb{N},i\in I_\lambda}$ over the same set of indices $I = \bigcup_{\lambda\in\mathbb{N}} I_\lambda$ and a function δ, we write $\mathcal{X} \overset{comp}{\approx}_\delta \mathcal{Y}$ to mean that for any non-uniform QPT algorithm $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}$, there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$, $i \in I_\lambda$, we have

$$|\Pr[\mathcal{A}_\lambda(X_i;\rho_\lambda)] - \Pr[\mathcal{A}_\lambda(Y_i;\rho_\lambda)]| \leq \delta(\lambda) + \mathsf{negl}(\lambda).$$

Especially, when we have the above for $\delta = 0$, we say that $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable, and simply write $\mathcal{X} \overset{comp}{\approx} \mathcal{Y}$.

Similarly, we write $\mathcal{X} \overset{stat}{\approx}_\delta \mathcal{Y}$ to mean that for any unbounded time algorithm $\mathcal{A}$, there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$, $i \in I_\lambda$, we have

$$|\Pr[\mathcal{A}(X_i)] - \Pr[\mathcal{A}(Y_i)]| \leq \delta(\lambda) + \mathsf{negl}(\lambda).$$

Especially, when we have the above for $\delta = 0$, we say that $\mathcal{X}$ and $\mathcal{Y}$ are statistically indistinguishable, and simply write $\mathcal{X} \overset{stat}{\approx} \mathcal{Y}$. Moreover, we write $\mathcal{X} \equiv \mathcal{Y}$ to mean that X_i and Y_i are distributed identically for all $i \in I$[14].

2.1 Post-Quantum One-Way Functions and Collapsing Hash Functions

A post-quantum one-way function (OWF) is a classically computable function that is hard to invert in QPT. A collapsing hash function is a quantum counterpart of collision-resistant hash function introduced by Unruh [Unr16b]. Unruh [Unr16a] gave a construction of collapsing hash functions based on the QLWE

[14] In other words, $\mathcal{X} \overset{stat}{\approx}_\delta \mathcal{Y}$ means that there exists a negligible function negl such that the trace distance between ρ_{X_i} and ρ_{Y_i} is at most $\delta(\lambda) + \mathsf{negl}(\lambda)$ for all $\lambda \in \mathbb{N}$ and $i \in I_\lambda$ where ρ_{X_i} and ρ_{Y_i} denote density matrices corresponding to X_i and Y_i.

assumption. We give formal definitions in the full version since they are only used for constructing other cryptographic primitives and not directly used in our constructions.

2.2 Commitment

We use commitments in our constructions. Though they are mostly standard, we need one new security notion which we call *strong collapse-binding*, which is a stronger variant of collapse-biding introduced by Unruh [Unr16b]. Roughly speaking, this security requires that for any superposition of messages and randomness corresponding the same commitment generated by an adversary, the adversary cannot distinguish if the message and randomness registers are measured or not. The difference from the original collapse-binding property is that both message and randomness registers are measured rather than only the message register. We observe that the collapse-binding commitment based on collapsing hash functions in [Unr16b] also satisfies the strong collapse-binding property. Especially, there exists a strong collapse-binding commitment under the QLWE assumption. See the full version for details of the definition and construction of strong collapse-binding commitments.

2.3 Interactive Proof and Argument

We define interactive proofs and arguments similarly to [BS20].

Notations. For an **NP** language L and $x \in L$, $R_L(x)$ is the set that consists of all (classical) witnesses w such that the verification machine for L accepts (x, w).

A (classical) interactive protocol is modeled as an interaction between interactive quantum machines P referred to as a prover and V referred to as a verifier that can be implemented by PPT algorithms. We denote by $\langle P(x_P), V(x_V)\rangle(x)$ an execution of the protocol where x is a common input, x_P is P's private input, and x_V is V's private input. We denote by $\mathsf{OUT}_V\langle P(x_P), V(x_V)\rangle(x)$ the final output of V in the execution. An honest verifier's output is $\top$ indicating acceptance or $\bot$ indicating rejection, and a quantum malicious verifier's output may be an arbitrary quantum state.

Definition 2.1 (Interactive Proof and Argument for NP). *An interactive proof or argument for an* **NP** *language L is an interactive protocol between a PPT prover P and a PPT verifier V that satisfies the following:*

Perfect Completeness. For any $x \in L$, and $w \in R_L(x)$, we have

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

Statistical/Computational Soundness. We say that an interactive protocol is statistically (resp. computationally) sound if for any unbounded-time (resp. non-uniform QPT) cheating prover P^*, there exists a negligible function negl *such that for any $\lambda \in \mathbb{N}$ and any $x \in \{0,1\}^\lambda \setminus L$, we have*

$$\Pr[\mathsf{OUT}_V\langle P^*, V\rangle(x) = \top] \leq \mathsf{negl}(\lambda).$$

We call an interactive protocol with statistical (resp. computational) soundness an interactive proof (resp. argument).

Delayed-Witness Σ-Protocol. We introduce a special type of Σ-protocol which we call *delayed-witness Σ-protocol* where the first message can be generated without witness.

Definition 2.2 (Delayed-Witness Σ-Protocol). *A (post-quantum) delayed-witness Σ-protocol for an* **NP** *language L is a 3-round interactive proof for* **NP** *with the following syntax.*

Common Input: An instance $x \in L \cap \{0,1\}^\lambda$ for security parameter $\lambda \in \mathbb{N}$.
P's **Private Input:** A classical witness $w \in R_L(x)$ for x.

1. P generates a "commitment" a and a state st. For this part, P only uses the statement x and does not use any witness w. We denote this procedure by $(a, \mathsf{st}) \overset{\$}{\leftarrow} \Sigma.P_1(x)$. Then it sends a to the verifier, and keeps st as its internal state.
2. V chooses a"challenge" $e \overset{\$}{\leftarrow} \{0,1\}^\lambda$ and sends e to P.
3. P generates a "response" z from st, witness w, and e. We denote this procedure by $z \overset{\$}{\leftarrow} \Sigma.P_3(\mathsf{st}, w, e)$. Then it sends z to V.
4. V verifies the transcript (a, e, z) and outputs $\top$ indicating acceptance or $\bot$ indicating rejection. We denote this procedure by $\top/\bot \overset{\$}{\leftarrow} \Sigma.V(x, a, e, z)$.

We require a delayed-witness Σ-protocol to satisfy the following property in addition to perfect completeness and statistical soundness.[15]

Special Honest-Verifier Zero-Knowledge. There exists a PPT simulator Sim_Σ such that we have

$$\{(a, z) : (a, \mathsf{st}) \overset{\$}{\leftarrow} \Sigma.P_1(x), z \overset{\$}{\leftarrow} \Sigma.P_3(\mathsf{st}, w, e)\}_{\lambda,x,w,e} \overset{comp}{\approx} \{(a, z) : (a, z) \overset{\$}{\leftarrow} \mathsf{Sim}_\Sigma(x, e)\}_{\lambda,x,w,e}$$

where $x \in L \cap \{0,1\}^\lambda$, $w \in R_L(x)$, and $e \in \{0,1\}^\lambda$.

[15] We do not require *special soundness*, which is often a default requirement of Σ-protocol.

Instantiations. An example of a delayed-witness Σ-protocol is a parallel repetition version of Blum's Graph Hamiltonicity protocol [Blu86]. In the protocol, we need a computationally hiding and perfectly binding non-interactive commitment scheme, which exists under the QLWE assumption as noted in Sect. 2.2. In summary, a delayed-input Σ-protocol for all **NP** languages exists under the QLWE assumption.

Quantum ϵ-Zero-Knowledge Proof and Argument. Here, we define quantum black-box ϵ-zero-knowledge proofs and arguments. The difference from the definition of quantum zero-knowledge in [BS20] are:

1. (**ϵ-Zero-Knowledge**) We allow the simulator to depend on a noticeable "accuracy parameter" ϵ, and allows its running time to polynomially depend on ϵ^{-1}, and
2. (**Black-Box Simulation**) the simulator is only given black-box access to a malicious verifier.

Definition 2.3 (Post-Quantum Black-Box ϵ-Zero-Knowledge Proof and Argument). *A post-quantum black-box ϵ-zero-knowledge proof (resp. argument) for an* **NP** *language L is an interactive proof (resp. argument) for L that satisfies the following property in addition to perfect completeness and statistical (resp. computational) soundness:*

Quantum Black-Box ϵ-Zero-Knowledge. There exists an oracle-aided QPT simulator Sim *such that for any non-uniform QPT malicious verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$ and any noticeable function $\epsilon(\lambda)$, we have*

$$\{\mathsf{OUT}_{V^*_\lambda}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_\epsilon \{\mathsf{OUT}_{V^*_\lambda}(\mathsf{Sim}^{V^*_\lambda(\rho_\lambda)}(x, 1^{\epsilon^{-1}}))\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, $w \in R_L(\lambda)$, and $\mathsf{OUT}_{V^*_\lambda}(\mathsf{Sim}^{V^*_\lambda(\rho_\lambda)}(x))$ is the state in the output register of V^*_λ after the simulated execution of V^*_λ by Sim.

Remark 2.1. In the above definition of quantum black-box ϵ-zero-knowledge, we do not consider an entanglement between auxiliary input of a malicious verifier and distinguisher unlike the original definition of quantum zero-knowledge by Watrous [Wat09]. However, in the full version we show that the above definition implies indistinguishability against a distinguisher that may get an entangled state to verifier's auxiliary input by taking advantage of black-box simulation.

Witness Indistinguishable Proof of Knowledge. The definition of witness indistinguishable proof of knowledge is given in the full version.

2.4 Quantum Rewinding Lemma

Watrous [Wat09] proved a lemma that enables us to amplify the success probability of a quantum algorithm under certain conditions. The following form of the lemma is based on that in [BS20, Lemma 2.1].

Lemma 2.1 **([Wat09,BS20]).** *There is an oracle-aided quantum algorithm* R *that gets as input the following:*

- *A quantum circuit* Q *that takes* n*-input qubits in register* Inp *and outputs a classical bit* b *(in a register outside* Inp*) and an* m *output qubits.*
- *An* n*-qubit state* ρ *in register* Inp*.*
- *A number* $T \in \mathbb{N}$ *in unary.*

$\mathsf{R}(1^T, \mathsf{Q}, \rho)$ *executes in time* $T \cdot |\mathsf{Q}|$ *and outputs a distribution over* m*-qubit states* $D_\rho := \mathsf{R}(1^T, \mathsf{Q}, \rho)$ *with the following guarantees.*

For an n*-qubit state* ρ*, denote by* Q_ρ *the conditional distribution of the output distribution* $\mathsf{Q}(\rho)$*, conditioned on* $b = 0$*, and denote by* $p(\rho)$ *the probability that* $b = 0$*. If there exist* $p_0, q \in (0, 1)$*,* $\gamma \in (0, \frac{1}{2})$ *such that:*

- *Amplification executes for enough time:* $T \geq \frac{\log(1/\gamma)}{4p_0(1-p_0)}$*,*
- *There is some minimal probability that* $b = 0$*: For every* n*-qubit state* ρ*,* $p_0 \leq p(\rho)$*,*
- $p(\rho)$ *is input-independent, up to* γ *distance: For every* n*-qubit state* ρ*,* $|p(\rho) - q| < \gamma$*, and*
- q *is closer to* $\frac{1}{2}$*:* $p_0(1 - p_0) \leq q(1 - q)$*,*

then for every n*-qubit state* ρ*,*

$$\mathsf{TD}(\mathsf{Q}_\rho, D_\rho) \leq 4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1 - p_0)}.$$

Moreover, $\mathsf{R}(1^T, \mathsf{Q}, \rho)$ *works in the following manner: It uses* Q *for only implementing oracles that perform the unitary part of* Q *and its inverse, acts on* Inp *only through these oracles, and the output of* R *is the state in the output register of* Q *after the simulated execution. We note that* R *may directly act on* Q*'s internal registers other than* Inp*.*

Remark 2.2. The final claim of the lemma ("Moreover...") is not explicitly stated in previous works. In the description of R in [Wat09], the first qubit of Inp is designated to output b, and thus the above requirement is not satisfied. However, this can be easily avoided by just letting Q output b in a register outside Inp as required above. Then one can see that R acts on the input register only through Q as seen from the description of R in [Wat09] (with the above modification in mind). Looking ahead, this is needed to show our ϵ-zero-knowledge simulators are black-box.

3 Extraction Lemma

In this section, we prove our main technical lemma, which we call the *extraction lemma*. Before giving a formal statement, we give an intuitive explanation. Suppose that we have a two-stage quantum algorithm $\mathcal{A} = (\mathcal{A}_{\mathsf{com}}, \mathcal{A}_{\mathsf{open}})$ that works as follows. $\mathcal{A}_{\mathsf{com}}$ is given pp of a commitment scheme and generates a commitment

com, and passes a quantum state ρ_{st} in its internal register to $\mathcal{A}_{\mathsf{open}}$. $\mathcal{A}_{\mathsf{open}}$ is given the internal state ρ_{st}, and outputs a message-randomness pair (m, r) (which is not necessarily a valid opening to com) along with a classical output out, and let ρ'_{st} be its internal state after the execution. We call a successive execution of $\mathcal{A}_{\mathsf{com}}$ and $\mathcal{A}_{\mathsf{open}}$ a real experiment. On the other hand, we consider an *extraction experiment* where an "extractor" Ext runs on input ρ_{st} in between $\mathcal{A}_{\mathsf{com}}$ and $\mathcal{A}_{\mathsf{open}}$ to "extract" a committed message m_{Ext} while generating a simulated $\mathcal{A}$'s internal state ρ_{Ext}. Then we run $\mathcal{A}_{\mathsf{open}}$ with the internal state ρ_{Ext} instead of ρ_{st} to complete the extraction experiment. Roughly, the extraction lemma claims that if the commitment scheme is strong collapse-binding (resp. statistically binding), then there exists an extractor Ext such that we have $m = m_{\mathsf{Ext}}$ with high probability and distributions of $(m, r, \mathsf{out}, \rho'_{\mathsf{st}})$ in real and extraction experiments are computationally (resp. statistically) indistinguishable *conditioned on that* (m, r) *is a valid opening to* com.

The formal statement is given below.

Definition 3.1 (Extraction Experiments). *Let* $\mathsf{Com} = (\mathsf{Setup}, \mathsf{Commit})$ *be a commitment scheme with message space* $\mathcal{M}$*, randomness space* $\mathcal{R}$*, commitment space* $\mathcal{COM}$*, and a public parameter space* $\mathcal{PP}$*. Let* $\mathcal{A} = \{\mathcal{A}_{\mathsf{com},\lambda}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ *be a sequence of two-stage non-uniform QPT algorithms with the following syntax:*

$\mathcal{A}_{\mathsf{com},\lambda}(\mathsf{pp}; \rho_\lambda) \rightarrow (\mathsf{com}, \rho_{\mathsf{st}})$*: It takes as input* $\mathsf{pp} \in \mathcal{PP}$ *and an advice* ρ_λ*, and outputs* $\mathsf{com} \in \mathcal{COM}$ *and a quantum state* ρ_{st} *in register* $\mathbf{ST}$*.*

$\mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}}) \rightarrow (m, r, \mathsf{out}, \rho'_{\mathsf{st}})$*: It takes as input a quantum state* ρ_{st} *in register* $\mathbf{ST}$*, and outputs* $m \in \mathcal{M}$*,* $r \in \mathcal{R}$*, a classical string* out*, and a quantum state* ρ'_{st} *in register* $\mathbf{ST}$*.*

Let Ext *be a QPT algorithm and* δ *be a function in* λ*. Then we define following experiments:*

```
$\underline{\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com}, \mathcal{A}](\lambda)}$
$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$,
$(\mathsf{com}, \rho_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{com},\lambda}(\mathsf{pp}; \rho_\lambda)$,

$(m, r, \mathsf{out}, \rho'_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}})$,
If $\mathsf{Commit}(\mathsf{pp}, m; r) \neq \mathsf{com}$,
    Output $\bot$
Else Output $(\mathsf{pp}, \mathsf{com}, m, r, \mathsf{out}, \rho'_{\mathsf{st}})$.
```

```
$\underline{\mathsf{Exp}_{\mathsf{ext}}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)}$
$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$,
$(\mathsf{com}, \rho_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{com},\lambda}(\mathsf{pp}; \rho_\lambda)$,
$(m_{\mathsf{Ext}}, \rho_{\mathsf{Ext}}) \xleftarrow{\$} \mathsf{Ext}(1^\lambda, 1^{\delta^{-1}}, \mathsf{pp}, \mathsf{com}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_{\mathsf{st}})$,
$(m, r, \mathsf{out}, \rho'_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{Ext}})$,
If $\mathsf{Commit}(\mathsf{pp}, m; r) \neq \mathsf{com} \vee m \neq m_{\mathsf{Ext}}$,
    Output $\bot$
Else Output $(\mathsf{pp}, \mathsf{com}, m, r, \mathsf{out}, \rho'_{\mathsf{st}})$.
```

Lemma 3.1 (Extraction Lemma). *For any strong collapse-binding commitment scheme* $\mathsf{Com} = (\mathsf{Setup}, \mathsf{Commit})$*, there exists a QPT algorithm* Ext *such that for any noticeable function* $\delta(\lambda)$ *and* $\mathcal{A} = \{\mathcal{A}_{\mathsf{com},\lambda}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ *as in Definition 3.1, we have*

$$\{\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com}, \mathcal{A}](\lambda)\}_{\lambda\in\mathbb{N}} \overset{comp}{\approx}_{\delta} \{\mathsf{Exp}_{\mathsf{ext}}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)\}_{\lambda\in\mathbb{N}}.$$

If Com *is statistically binding instead of strong collapse-binding, we have*

$$\{\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com},\mathcal{A}](\lambda)\}_{\lambda\in\mathbb{N}} \overset{stat}{\approx}_{\delta} \{\mathsf{Exp}_{\mathsf{ext}}[\mathsf{Com},\mathcal{A},\mathsf{Ext}](\lambda,\delta)\}_{\lambda\in\mathbb{N}}.$$

Moreover, $\mathsf{Ext}(1^{\lambda},1^{\delta^{-1}},\mathsf{pp},\mathsf{com},\mathcal{A}_{\mathsf{open},\lambda},\rho_{\mathsf{st}})$ *works in the following manner: It uses* $\mathcal{A}_{\mathsf{open},\lambda}$ *for only implementing oracles that perform unitary part of* $\mathcal{A}_{\mathsf{open},\lambda}$ *and its inverse, and acts on* $\mathbf{ST}$ *only through black-box access to the oracles. The second output* ρ_{Ext} *of* Ext *is the state in* $\mathbf{ST}$ *after the execution. We note that* Ext *may directly act on internal registers of* $\mathcal{A}_{\mathsf{open},\lambda}$ *other than* $\mathbf{ST}$*.*

The above lemma abstracts our technical core, which is extraction of the verifier's committed challenge without collapsing verifier's internal state too much. (One can think of $\mathcal{A}$ in the above lemma as the verifier and ρ_{st} and ρ'_{st} as verifier's internal states before and after opening the commitment, respectively, in our constant round ϵ-zero-knowledge proofs/arguments.) Since the intuition of the proof is already explained in Sect. 1.2, we defer the proof to the full version.

4 Post-quantum ϵ-Zero-Knowledge Proof and Argument

In this section, we prove the following theorems.

Theorem 4.1. *If the QLWE assumption holds, then there exists a* 5*-round post-quantum black-box* ϵ*-zero-knowledge proof for all* $\mathbf{NP}$ *languages.*

Theorem 4.2. *If a collapsing hash function exists, then there exists a* 5*-round post-quantum black-box* ϵ*-zero-knowledge proof for all* $\mathbf{NP}$ *languages.*

Theorem 4.3. *If post-quantunm OWF exists, then there exists a 9-round post-quantum black-box* ϵ*-zero-knowledge argument for all* $\mathbf{NP}$ *languages.*

In the rest of this section, we prove Theorem 4.1 and 4.2. The proof of Theorem 4.3 is given in the full version.

4.1 Construction

Our construction is the same as the Golderich-Kahan protocol [GK96] except that we instantiate the verifier's commitment with a strong collapse-binding commitment and we rely on a post-quantum delayed-witness Σ-protocol. Specifically, our construction is built on the following ingredients:

- A commitment scheme ($\mathsf{CBCom.Setup}$, $\mathsf{CBCom.Commit}$) that is statistical hiding and strong collapse-binding with message space $\{0,1\}^{\lambda}$ and randomness space $\mathcal{R}$. As noted in Sect. 2.2, such a commitment scheme exists under the QLWE assumption.
- A delayed-witness Σ-protocol $(\Sigma.P_1,\Sigma.P_3,\Sigma.V)$ for an $\mathbf{NP}$ language L as defined in Definition 2.2. As noted in Sect. 2.3, such a protocol exists under the QLWE assumption.

Protocol 1

Common Input: An instance $x \in L \cap \{0,1\}^\lambda$ for security parameter $\lambda \in \mathbb{N}$.
P**'s Private Input:** A classical witness $w \in R_L(x)$ for x.

1. V**'s Commitment to Challenge:**
   (a) P computes $\mathsf{pp} \overset{\$}{\leftarrow} \mathsf{CBCom.Setup}(1^\lambda)$ and sends pp to V.
   (b) V chooses $e \overset{\$}{\leftarrow} \{0,1\}^\lambda$ and $r \overset{\$}{\leftarrow} \mathcal{R}$, computes $\mathsf{com} \overset{\$}{\leftarrow} \mathsf{CBCom.Commit}(\mathsf{pp}, e; r)$, and sends com to P.
2. Σ**-Protocol Execution:**
   (a) P generates $(a, \mathsf{st}) \overset{\$}{\leftarrow} \Sigma.P_1(x)$ and sends a to V.
   (b) V sends (e, r) to P.
   (c) P aborts if $\mathsf{CBCom.Commit}(\mathsf{pp}, e; r) \neq \mathsf{com}$.
   Otherwise, it generates $z \overset{\$}{\leftarrow} \Sigma.P_3(\mathsf{st}, w, e)$ and sends z to V.
   (d) V outputs $\Sigma.V(x, a, e, z)$.

Fig. 1. Constant-round post-quantum ϵ-zero-knowledge proof for $L \in \mathbf{NP}$

Then our construction of post-quantum black-box ϵ-zero-knowledge proof is given in Fig. 1.

The completeness of the protocol clearly follows from that of the underlying Σ-protocol. In Sect. 4.2 and 4.3, we prove that this protocol satisfies statistical soundness and quantum black-box ϵ-zero-knowledge. Then we obtain Theorem 4.1.

4.2 Statistical Soundness

This is essentially the same as the proof in [GK96], but we give a proof for completeness.

For $x \notin L$ an unbounded-time cheating prover P^*, we consider the following sequence of hybrids. We denote by win_i the event that P^* wins in Hyb_i.

Hyb_1: This is the original game. That is,
1. P^* sends pp to V.
2. V chooses $e \overset{\$}{\leftarrow} \{0,1\}^\lambda$ and $r \overset{\$}{\leftarrow} \mathcal{R}$, computes $\mathsf{com} \overset{\$}{\leftarrow} \mathsf{CBCom.Commit}(\mathsf{pp}, e; r)$, and sends com to P^*.
3. P^* sends a to V.
4. V sends (e, r) to P^*
5. P^* sends z to V.

We say that P^* wins if we have $\Sigma.V(x, a, e, z) = \top$.

Hyb_2: This hybrid is identical to the previous one except that in Step 4, V uniformly chooses r' such that $\mathsf{com} = \mathsf{CBCom.Commit}(\mathsf{pp}, e; r')$ and sends (e, r') to P^* instead of (e, r). We note that this procedure may be inefficient. This is just a conceptual change and thus we have $\Pr[\mathsf{win}_1] = \Pr[\mathsf{win}_2]$.

Hyb_3: This hybrid is identical to the previous one except that in Step 2, V sends $\mathsf{com} \xleftarrow{\$} \mathsf{CBCom.Commit}(\mathsf{pp}, 0^\ell; r)$ and the generation of e is delayed to Step 4. Since no information of r is given to P^* due to the modification made in Hyb_2, by the statistical hiding property of CBCom, we have $|\Pr[\mathsf{win}_3] - \Pr[\mathsf{win}_2]| = \mathsf{negl}(\lambda)$.

Now, it is easy to prove $\Pr[\mathsf{win}_3] = \mathsf{negl}(\lambda)$ by reducing it to the statistical soundness of the Σ-protocol. Namely, we consider a cheating prover $\Sigma.P^*$ against the Σ-protocol that works as follows.

1. $\Sigma.P^*$ runs P^* to get the first message pp.
2. $\Sigma.P^*$ computes $\mathsf{com} \xleftarrow{\$} \mathsf{CBCom.Commit}(\mathsf{pp}, 0^\ell; r)$, sends com to P^*, and gets the third message a. Then $\Sigma.P^*$ sends a to its own external challenger as the first message of the Σ-protocol.
3. Upon receiving a challenge e from the external challenger, $\Sigma.P^*$ uniformly chooses r' such that $\mathsf{com} = \mathsf{CBCom.Commit}(\mathsf{pp}, e; r')$, sends (e, r') to P^*, and gets the P^*'s final message z. Then $\Sigma.P^*$ sends z to the external challenger.

It is easy to see that $\Sigma.P^*$ perfectly simulates the environment in Hyb_3 for P^*. Therefore, $\Sigma.P^*$'s winning probability is equal to $\Pr[\mathsf{win}_3]$. On the other hand, by soundness of the Σ-protocol, $\Sigma.P^*$'s winning probability is $\mathsf{negl}(\lambda)$. Therefore we have $\Pr[\mathsf{win}_3] = \mathsf{negl}(\lambda)$.

Combining the above, we have $\Pr[\mathsf{win}_1] = \mathsf{negl}(\lambda)$, which means that the protocol satisfies the statistical soundness.

4.3 Quantum Black-Box ϵ-Zero-Knowledge

Structure of the Proof. A high-level structure of our proof is similar to that of [BS20]. Specifically, we first construct simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ that simulate the "aborting case" and "non-aborting case", respectively. More precisely, $\mathsf{Sim}_{\mathsf{a}}$ correctly simulates the verifier's view if the verifier aborts and otherwise returns a failure symbol Fail and $\mathsf{Sim}_{\mathsf{na}}$ correctly simulates the verifier's view if the verifier does not abort and otherwise returns a failure symbol Fail. Then we consider a combined simulator $\mathsf{Sim}_{\mathsf{comb}}$ that runs either of $\mathsf{Sim}_{\mathsf{a}}$ or $\mathsf{Sim}_{\mathsf{na}}$ with equal probability. Then $\mathsf{Sim}_{\mathsf{comb}}$ correctly simulates the verifier's view conditioned on that the output is not Fail, and it returns Fail with probability almost 1/2. By applying the Watrous' quantum rewinding lemma (Lemma 2.1) to $\mathsf{Sim}_{\mathsf{comb}}$, we can convert it to a full-fledged simulator.

Though the above high-level structure is similar to [BS20], the analyses of simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ are completely different from [BS20] since we consider different protocols. While the analysis of $\mathsf{Sim}_{\mathsf{a}}$ is easy, the analysis of $\mathsf{Sim}_{\mathsf{na}}$ is a little more complicated as it requires the extraction lemma (Lemma 3.1), which was developed in Sect. 3.

Proof of Quantum Black-Box ϵ-Zero-Knowledge. For clarity of exposition, we first show the quantum ϵ-zero-knowledge property ignoring that the simulator should be black-box. That is, we give the full description of the malicious verifier and its quantum advice as part of the simulator's input instead of only the oracle access to the verifier. At the end of the proof, we explain that the simulator is indeed black-box.

In quantum ϵ-zero-knowledge, we need to show a simulator Sim that takes an accuracy parameter $1^{\epsilon^{-1}}$ as part of its input. We assume $\epsilon(\lambda) = o(1)$ without loss of generality since the other case trivially follows from this case. Without loss of generality, we can assume that a malicious verifier V^* does not terminate the protocol before the prover aborts since it does not gain anything by declaring the termination. We say that V^* aborts if it fails to provide a valid opening (e, r) to com in Step 2b (i.e., the prover aborts in Step 2c).

First, we construct a simulator $\mathsf{Sim}_{\mathsf{comb}}$, which returns a special symbol Fail with probability roughly $1/2$ but almost correctly simulates the output of V^*_λ conditioned on that it does not return Fail. The simulator $\mathsf{Sim}_{\mathsf{comb}}$ uses simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ as sub-protocols:

```
$\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$:
    1. Choose $\mathsf{mode} \xleftarrow{\$} \{\mathsf{a}, \mathsf{na}\}$.
    2. Run $\mathsf{Sim}_{\mathsf{mode}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$.
    3. Output what $\mathsf{Sim}_{\mathsf{mode}}$ outputs.
$\mathsf{Sim}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$:[16]
    1. Set $V^*_\lambda$'s internal state to $\rho_\lambda$.
    2. Compute $\mathsf{pp} \xleftarrow{\$} \mathsf{CBCom.Setup}(1^\lambda)$ and send $\mathsf{pp}$ to $V^*_\lambda$.
    3. $V^*_\lambda$ returns $\mathsf{com}$.
    4. Compute $(a, \mathsf{st}) \xleftarrow{\$} \Sigma.P_1(x)$ and send $a$ to $V^*_\lambda$.
    5. $V^*_\lambda$ returns $(e, r)$.
    6. Return $\mathsf{Fail}$ and abort if $\mathsf{CBCom.Commit}(\mathsf{pp}, e; r) = \mathsf{com}$.
       Otherwise, let $V^*_\lambda$ output the final output notifying that the prover aborts.
    7. The final output of $V^*_\lambda$ is treated as the output $\mathsf{Sim}_{\mathsf{a}}$.
$\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$:
    1. Set $V^*_\lambda$'s internal state to $\rho_\lambda$.
    2. Compute $\mathsf{pp} \xleftarrow{\$} \mathsf{CBCom.Setup}(1^\lambda)$ and send $\mathsf{pp}$ to $V^*_\lambda$.
    3. $V^*_\lambda$ returns $\mathsf{com}$. Let $\rho_{\mathsf{st}}$ be the internal state of $V^*_\lambda$ at this point.
    4. Compute $(e_{\mathsf{Ext}}, \rho_{\mathsf{Ext}}) \xleftarrow{\$} \mathsf{Ext}(1^\lambda, 1^{\delta^{-1}}, \mathsf{pp}, \mathsf{com}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_{\mathsf{st}})$ where $\mathsf{Ext}$ is as
       in Lemma 3.1 for the commitment scheme $\mathsf{CBCom}$, $\delta := \frac{\epsilon^2}{3600 \log^4(\lambda)}$, and
       $\mathcal{A} = (\mathcal{A}_{\mathsf{com},\lambda}, \mathcal{A}_{\mathsf{open},\lambda})$ as defined below:
        $\mathcal{A}_{\mathsf{com},\lambda}(\mathsf{pp}; \rho_\lambda)$: It sets $V^*_\lambda$'s internal state to $\rho_\lambda$ and sends $\mathsf{pp}$ to $V^*_\lambda$. Let
            $\mathsf{com}$ be the response by $V^*_\lambda$ and $\rho_{\mathsf{st}}$ be the internal state of $V^*_\lambda$ at this
            point. It outputs $(\mathsf{com}, \rho_{\mathsf{st}})$.
        $\mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}})$: It generates $(a, \mathsf{st}) \xleftarrow{\$} \Sigma.P_1(x)$,[17] sets $V^*_\lambda$'s internal state
            to $\rho_{\mathsf{st}}$, and sends $a$ to $V^*_\lambda$. Let $(e, r)$ be the response by $V^*_\lambda$ and let
```

[16] Though $\mathsf{Sim}_{\mathsf{a}}$ does not depend on ϵ, we include $1^{\epsilon^{-1}}$ in the input for notational uniformity.

[17] We note that we consider x to be hardwired into $\mathcal{A}_{\mathsf{open},\lambda}$. We also note that though $\mathcal{A}_{\mathsf{open},\lambda}$ does not take explicit randomness, it can generate randomness by say, applying Hadamard on its working register and then measuring it.

ρ'_{st} be the internal state of V^*_λ at this point. It outputs $(e, r, \mathsf{out} := (a, \mathsf{st}), \rho'_{\mathsf{st}})$.

Here, we remark that V^*_λ's internal register corresponds to $\mathbf{ST}$ and e corresponds to m in the notation of Lemma 3.1.

5. Set the verifier's internal state to ρ_{Ext}.
6. Compute $(a, z) \overset{\$}{\leftarrow} \mathsf{Sim}_\Sigma(x, e_{\mathsf{Ext}})$ and send a to V^*_λ.
7. V^*_λ returns (e, r).
8. Return Fail and abort if $e \neq e_{\mathsf{Ext}}$ or $\mathsf{CBCom.Commit}(\mathsf{pp}, e; r) \neq \mathsf{com}$. Otherwise, send z to V^*_λ.
9. The final output of V^*_λ is treated as the output $\mathsf{Sim}_{\mathsf{na}}$.

Intuitively, $\mathsf{Sim}_{\mathsf{a}}$ (resp. $\mathsf{Sim}_{\mathsf{na}}$) is a simulator that simulates the verifier's view in the case that verifier aborts (resp. does not abort).

More formally, we prove the following lemmas.

Lemma 4.1 **($\mathsf{Sim}_{\mathsf{a}}$ simulates the aborting case).** *For any non-uniform QPT malicious verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $\mathsf{OUT}_{V^*_{\mathsf{a}}}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)$ be the V^*_λ's final output that is replaced with* Fail *if V^*_λ does not abort. Then we have*

$$\{\mathsf{OUT}_{V^*_{\mathsf{a}}}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \equiv \{\mathsf{Sim}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}.$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

Proof. Since $\mathsf{Sim}_{\mathsf{a}}$ perfectly simulates the real execution for V^*_λ when it aborts, Lemma 4.1 immediately follows.

Lemma 4.2 **($\mathsf{Sim}_{\mathsf{na}}$ simulates the non-aborting case).** *For any non-uniform QPT malicious verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle$ (x) be the V^*_λ's final output that is replaced with* Fail *if V^*_λ aborts. Then we have*

$$\{\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_\delta \{\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

Proof. Here, we analyze $\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$. In the following, we consider hybrid simulators $\mathsf{Sim}_{\mathsf{na},i}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ for $i = 1, 2, 3$. We remark that they also take the witness w as input unlike $\mathsf{Sim}_{\mathsf{na}}$.

$\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$: This simulator works similarly to $\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ except that it generates $(a, \mathsf{st}) \overset{\$}{\leftarrow} \Sigma.P_1(x)$ and $z \overset{\$}{\leftarrow} \Sigma.P_3(\mathsf{st}, w, e_{\mathsf{Ext}})$ instead of $(a, z) \overset{\$}{\leftarrow} \mathsf{Sim}_\Sigma(x, e_{\mathsf{Ext}})$ in Step 6.

By the special honest-verifier zero-knowledge property of the Σ-protocol, we have

$$\{\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \overset{comp}{\approx} \{\{\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

$\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$: This simulator works similarly to $\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ except that the generation of z is delayed until Step 8 and it is generated as $z \overset{\$}{\leftarrow} \Sigma.P_3(\mathsf{st}, w, e)$ instead of $z \overset{\$}{\leftarrow} \Sigma.P_3(\mathsf{st}, w, e_{\mathsf{Ext}})$.

The modification does not affect the output distribution since it outputs Fail if $e \neq e_{\mathsf{Ext}}$ and if $e = e_{\mathsf{Ext}}$, then this simulator works in exactly the same way as the previous one. Therefore we have

$$\{\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \equiv \{\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

$\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$: This simulator works similarly to $\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ except that Step 4 and 5 are deleted and the check of $e \neq e_{\mathsf{Ext}}$ in Step 8 is omitted. That is, it outputs Fail in Step 8 if and only if we have $\mathsf{CBCom.Commit}(\mathsf{pp}, e; r) \neq \mathsf{com}$.

We note that e_{Ext} and ρ_{Ext} are no longer used at all and thus need not be generated.

We can see that Step 3 is exactly the same as executing $(\mathsf{com}, \rho_{\mathsf{st}}) \overset{\$}{\leftarrow} \mathcal{A}_{\mathsf{com},\lambda}(\mathsf{pp}; \rho_\lambda)$ and Step 6 and 7 of previous and this experiments are exactly the same as executing $(e, r, \mathsf{out} = (a, \mathsf{st}), \rho'_{\mathsf{st}}) \overset{\$}{\leftarrow} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{Ext}})$ and $(e, r, \mathsf{out} = (a, \mathsf{st}), \rho'_{\mathsf{st}}) \overset{\$}{\leftarrow} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}})$, respectively where we define ρ'_{st} in simulated experiments as V^*_λ's internal state after Step 7. Moreover, the rest of execution of the simulators can be done given $(\mathsf{pp}, \mathsf{com}, e, r, \mathsf{out} = (a, \mathsf{st}), \rho'_{\mathsf{st}})$. Therefore, by a straightforward reduction to Lemma 3.1, we have

$$\{\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \overset{comp}{\approx}_\delta \{\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

We can see that $\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ perfectly simulates the real execution for V^*_λ and outputs V^*_λ's output conditioned on that V^*_λ does not abort, and just outputs Fail otherwise. Therefore, we have

$$\{\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \equiv \{\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$. Combining the above, Lemma 4.2 is proven.

By combining Lemmas 4.1 and 4.2, we can prove the following lemma.

Lemma 4.3 ($\mathsf{Sim}_{\mathsf{comb}}$ simulates V^*_λ's output with probability almost 1/2). *For any non-uniform QPT malicious verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $p^{\mathsf{suc}}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ be the probability that $\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ does not return* Fail *and $D_{\mathsf{sim},\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ be a conditional distribution of* $\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$*, conditioned on that it does not return* Fail. *There exists a negligible function* negl *such that for any $x = \{x_\lambda \in L \cap \{0,1\}^\lambda\}_{\lambda\in\mathbb{N}}$, we have*

$$\left|p^{\mathsf{suc}}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) - 1/2\right| \leq \delta/2 + \mathsf{negl}(\lambda). \quad (1)$$

Moreover, we have

$$\{\mathsf{OUT}_{V^*}\langle P(w), V_\lambda^*(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_{4\delta} \{D_{\mathsf{sim,comb}}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)\}_{\lambda,x,w} \quad (2)$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, *and* $w \in R_L(x)$.

Proof. (sketch.) Intuition of the proof is very easy: By Lemma 4.1 and 4.2, Sim_a and Sim_na almost simulate the real output distribution of V_λ^* conditioned on that V_λ^* aborts and does not abort, respectively. Therefore, if we randomly guess if V_λ^* aborts and runs either of Sim_a and Sim_na that successfully works for the guessed case, the output distribution is close to the real output distribution of V_λ^* conditioned on that the guess is correct, which happens with probability almost $1/2$.

Indeed, the actual proof is based on the above idea, but for obtaining concrete bounds as in Eq. 1 and 2, we need some tedious calculations. We give a full proof in the full version since the proof is easy and very similar to that in [BS20] (once we obtain Lemma 4.1 and 4.2).

Then, we convert $\mathsf{Sim}_\mathsf{comb}$ to a full-fledged simulator that does not return Fail by using the quantum rewinding lemma (Lemma 2.1). Namely, we let Q be a quantum algorithm that takes ρ_λ as input and outputs $\mathsf{Sim}_\mathsf{comb}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)$ where $b := 0$ if and only if it does not return Fail, $p_0 := \frac{1}{4}$, $q := \frac{1}{2}$, $\gamma := \delta$, and $T := 2\log(1/\delta)$. Then it is easy to check that the conditions for Lemma 2.1 is satisfied by Eq. 1 in Lemma 4.3 (for sufficiently large λ). Then by using Lemma 2.1, we can see that $\mathsf{R}(1^T, \mathsf{Q}, \rho_\lambda)$ runs in time $T \cdot |\mathsf{Q}| = \mathsf{poly}(\lambda)$ and its output (seen as a mixed state) has a trace distance bounded by $4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1-p_0)}$ from $D_{\mathsf{sim,comb}}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)$. Since we have $\gamma = \delta = \frac{\epsilon^2}{3600\log^4(\lambda)} = 1/\mathsf{poly}(\lambda)$, we have $4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1-p_0)} < 30\sqrt{\gamma}\log^2(\lambda) = \frac{\epsilon}{2}$ for sufficiently large λ where we used $\log(1/\gamma) = \log(\mathsf{poly}(\lambda)) = o(\log^2(\lambda))$. Thus, by combining the above and Eq. 2 in Lemma 4.3, if we define $\mathsf{Sim}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda) := \mathsf{R}(1^T, \mathsf{Q}, \rho_\lambda)$, then we have

$$\mathsf{OUT}_{V^*}\langle P(w), V_\lambda^*(\rho_\lambda)\rangle(x) \overset{comp}{\approx}_{\frac{\epsilon}{2}+4\delta} \mathsf{Sim}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda).$$

We can conclude the proof of quantum ϵ-zero-knowledge by noting that we have $\frac{\epsilon}{2} + 4\delta < \epsilon$ since we have $\delta = \frac{\epsilon^2}{3600\log^4(\lambda)} < \frac{\epsilon}{8}$.

Black-Box Simulation. Here, we explain that the simulator Sim constructed as above only needs black-box access to the verifier. What we need to show are that Sim applies the unitary part $U_{V_\lambda^*}$ of V_λ^* and its inverse $U_{V_\lambda^*}^\dagger$ only as oracles and Sim does not directly act on V_λ^*'s internal register. There are two parts of the construction of Sim that are not obviously black-box. The first is Step 4 and 5 of Sim_na where it runs the extraction algorithm Ext of Lemma 3.1, and the second is the conversion from $\mathsf{Sim}_\mathsf{comb}$ to Sim using R in Lemma 2.1. In the following, we explain that both steps can be implemented by black-box access to the verifier.

1. By Lemma 3.1, Ext uses the unitary part of $\mathcal{A}_{\mathsf{open},\lambda}$ and its inverse only in a black-box manner, and they can be implemented by black-box access to $U_{V^*_\lambda}$ and $U^\dagger_{V^*_\lambda}$. Moreover, since register $\mathbf{ST}$ in the notation of Lemma 3.1 corresponds to the internal register of V^*_λ in our context, the lemma ensures that Ext does not directly act on it. Also, $\mathsf{Sim}_{\mathsf{na}}$ need not explicitly set V^*_λ's internal register to ρ_{Ext} in Step 5 if we do the above black-box simulation since a state in the register automatically becomes ρ_{Ext} after the execution as stated in Lemma 3.1. Therefore, this step can be implemented by black-box access to V^*_λ.
2. Given the above observation, we now know that both $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ only need black-box access to V^*_λ. This means that Q only needs black-box access to V^*_λ. Since R only uses Q as oracles that perform the unitary part of Q and its inverse as stated in Lemma 2.1 and they can be implemented by black-box access to V^*_λ, R uses $U_{V^*_\lambda}$ and $U^\dagger_{V^*_\lambda}$ only as oracles. Moreover, since the register Inp in Lemma 2.1 corresponds to the internal register of V^*_λ in our context, R does not directly act on it.

By the above observations, we can see that the simulator Sim only needs black-box access to V^*_λ.

4.4 Instantiation from Collapsing Hash Function

Our construction in Fig. 1 is based on two building blocks: a statistically hiding and strong collapse-binding commitment scheme and a delayed-witness Σ-protocol. Though the former can be instantiated by a collapsing hash function, we do not know how to instantiate the latter by a collapsing hash function since it needs non-interactive commitment that is not known to be implied by collapsing hash functions. However, we can just use a 4-round version of a delayed-witness Σ-protocol where the first message "commitment" in the Σ-protocol is instantiated based on Naor's commitments [Nao91] instead of a non-interactive one. Since Naor's commitments can be instantiated under any OWF and collapsing hash function is trivially also one-way, we can instantiate the 4-round version of a delayed-witness Σ-protocol based on a collapsing hash function. We can prove security of the construction based on 4-round version of a delayed-witness Σ-protocol in essentially the same manner as the security proofs in Sect. 4.2 and 4.3. We also note that this does not increase the number of rounds of our construction. Based on these observations, we obtain Theorem 4.2.

Acknowledgement. NHC's research is support by the U.S. Department of Defense and NIST through the Hartree Postdoctoral Fellowship at QuICS and by NSF through IUCRC Planning Grant Indiana University: Center for Quantum Technologies (CQT) under award number 2052730. KMC's research is partially supported by MOST, Taiwan, under Grant no. MOST 109-2223-E-001-001-MY3 and Executive Yuan Data Safety and Talent Cultivation Project (ASKPQ-109-DSTCP).

References

[ACGH20] Alagic, G., Childs, A.M., Grilo, A.B., Hung, S.-H.: Non-interactive classical verification of quantum computation. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part III. LNCS, vol. 12552, pp. 153–180. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64381-2_6

[AL20] Ananth, P., La Placa, R.L.: Secure quantum extraction protocols. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part III. LNCS, vol. 12552, pp. 123–152. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64381-2_5

[AR06] Aharon, N., Regev, O.: Witness-preserving Amplification of QMA (lecture note) (2006). https://cims.nyu.edu/regev/teaching/quantum_fall_2005/ln/qma.pdf

[BC90] Brassard, G., Crepeau, C.: Sorting out zero-knowledge. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 181–191. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-46885-4_20

[BCY91] Brassard, G., Crépeau, C., Yung, M.: Constant-round perfect zero-knowledge computationally convincing protocols. Theor. Comput. Sci. **84**(1), 23–52 (1991)

[BG20] Broadbent, A., Grilo, A.B.: QMA-hardness of consistency of local density matrices with applications to quantum zero-knowledge. In: 61st FOCS, pp. 196–205 (2020)

[BJSW20] Broadbent, A., Ji, Z., Song, F., Watrous, J.: Zero-knowledge proof systems for QMA. SIAM J. Comput. **49**(2), 245–283 (2020)

[BKP18] Bitansky, N., Kalai, Y.T., Paneth, O.: Multi-collision resistance: a paradigm for keyless hash functions. In: 50th ACM STOC, pp. 671–684 (2018)

[BKP19] Bitansky, N., Khurana, D., Paneth, O.: Weak zero-knowledge beyond the black-box barrier. In: 51st ACM STOC, pp. 1091–1102 (2019)

[BL02] Barak, B., Lindell, Y.: Strict polynomial-time in simulation and extraction. In: 34th ACM STOC, pp. 484–493 (2002)

[BLP13] Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: 45th ACM STOC, pp. 575–584 (2013)

[Blu86] Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians, pp. 1444–1451 (1986)

[BP12] Bitansky, N., Paneth, O.: Point obfuscation and 3-round zero-knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 190–208. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_11

[Bra18] Brakerski, Z.: Quantum FHE (almost) as secure as classical. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 67–95. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96878-0_3

[BS20] Bitansky, N., Shmueli, O.: Post-quantum zero knowledge in constant rounds. In: 52nd ACM STOC, pp. 269–279 (2020)

[BY20] Brakerski, Z., Yuen, H.: Quantum Garbled Circuits. arXiv:2006.01085 (2020)

[CCLY21] Chia, N.-H., Chung, K.-M., Liu, Q., Yamakawa, T.: On the Impossibility of Post-Quantum Black-Box Zero-Knowledge in Constant Rounds. arXiv:2103.11244 (2021)

[CCY20] Chia, N.-H., Chung, K.-M., Yamakawa, T.: Classical verification of quantum computations with efficient verifier. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part III. LNCS, vol. 12552, pp. 181–206. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64381-2_7

[CLP15] Chung, K.-M., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 66–92. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_4

[CVZ20] Coladangelo, A., Vidick, T., Zhang, T.: Non-interactive zero-knowledge arguments for QMA, with preprocessing. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 799–828. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56877-1_28

[DFS04] Damgård, I., Fehr, S., Salvail, L.: Zero-knowledge proofs and string commitments withstanding quantum attacks. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 254–272. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_16

[DNRS03] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. J. ACM **50**(6), 852–921 (2003)

[DNS04] Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. J. ACM **51**(6), 851–898 (2004)

[FGJ18] Fleischhacker, N., Goyal, V., Jain, A.: On the existence of three round zero-knowledge proofs. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 3–33. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_1

[FS90] Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 526–544. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_46

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

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

[GMW91] Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. J. ACM **38**(3), 691–729 (1991)

[Gol01] Goldreich, O.: The Foundations of Cryptography - Volume 1: Basic Techniques. Cambridge University Press, Cambridge (2001)

[Gol04] Goldreich, O.: The Foundations of Cryptography - Volume 2: Basic Applications. Cambridge University Press, Cambridge (2004)

[Gra97] Graaf, J.V.D.: Towards a formal definition of security for quantum protocols. PhD thesis, University of Montreal, Montreal, Canada (1997)

[HILL99] Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)

[JKKR17] Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 158–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_6

[Kob03] Kobayashi, H.: Non-interactive quantum perfect and statistical zero-knowledge. In: Ibaraki, T., Katoh, N., Ono, H. (eds.) ISAAC 2003. LNCS, vol. 2906, pp. 178–188. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-24587-2_20

[Mah18a] Mahadev, U.: Classical homomorphic encryption for quantum circuits. In: 59th FOCS, pp. 332–338 (2018)

[Mah18b] Mahadev, U.: Classical verification of quantum computations. In: 59th FOCS, pp. 259–267 (2018)

[Nao91] Naor, M.: Bit commitment using pseudorandomness. J. Cryptol. **4**(2), 151–158 (1991)

[NWZ09] Nagaj, D., Wocjan, P., Zhang, Y.: Fast Amplification of QMA. arXiv:0904.1549 (2009)

[Pas03] Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 160–176. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_10

[Pei09] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: 41st ACM STOC, pp. 333–342 (2009)

[PS19] Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 89–114. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_4

[PW08] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: 40th ACM STOC, pp. 187–196 (2008)

[PW09] Pass, R., Wee, H.: Black-box constructions of two-party protocols from one-way functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 403–418. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_24

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

[Shm20] Shmueli, O.: Multi-theorem (Malicious) Designated-Verifier NIZK for QMA. arXiv:2007.12923 (2020)

[SV03] Sahai, A., Vadhan, S.P.: A complete problem for statistical zero knowledge. J. ACM **50**(2), 196–249 (2003)

[Unr12] Unruh, D.: Quantum proofs of knowledge. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 135–152. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_10

[Unr16a] Unruh, D.: Collapse-binding quantum commitments without random oracles. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 166–195. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_6

[Unr16b] Unruh, D.: Computationally binding quantum commitments. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 497–527. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_18

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

[Zha19] Zhandry, M.: Quantum lightning never strikes the same state twice. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 408–438. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17659-4_14