Non-interactive Zero-Knowledge Arguments for QMA, with Preprocessing

Andrea Coladangelo(✉), Thomas Vidick, and Tina Zhang

Computing and Mathematical Sciences, Caltech, Pasadena, USA
acoladan@caltech.edu

Abstract. A non-interactive zero-knowledge (NIZK) proof system for a language $L \in \mathrm{NP}$ allows a prover (who is provided with an instance $x \in L$, and a witness w for x) to compute a *classical certificate* π for the claim that $x \in L$ such that π has the following properties: 1) π can be verified efficiently, and 2) π does not reveal any information about w, besides the fact that it exists (i.e. that $x \in L$). NIZK proof systems have recently been shown to exist for all languages in NP in the common reference string (CRS) model and under the learning with errors (LWE) assumption.

We initiate the study of NIZK *arguments* for languages in QMA. An argument system differs from a proof system in that the honest prover must be efficient, and that it is only sound against (quantum) polynomial-time provers. Our first main result is the following: if LWE is hard for quantum computers, then any language in QMA has an *NIZK argument with preprocessing*. The preprocessing in our argument system consists of (i) the generation of a CRS and (ii) a *single (instance-independent) quantum message* from verifier to prover. The instance-dependent phase of our argument system, meanwhile, involves only a single *classical* message from prover to verifier. Importantly, verification in our protocol is entirely classical, and the verifier needs not have quantum memory; its only quantum actions are in the preprocessing phase. NIZK proofs of (classical) knowledge are widely used in the construction of more advanced cryptographic protocols, and we expect the quantum analogue to likewise find a broad range of applications. In this respect, the fact that our protocol has an entirely classical verification phase is particularly appealing.

Our second contribution is to extend the notion of a classical *proof of knowledge* to the quantum setting. We introduce the notions of *arguments* and *proofs of quantum knowledge* (AoQK/PoQK), and we show that our non-interactive argument system satisfies the definition of an AoQK, which extends its domain of usefulness with respect to cryptographic applications. In particular, we explicitly construct an extractor which can recover a quantum witness from any prover who is successful in our protocol. We also show that any language in QMA has an (interactive) *proof of quantum knowledge*, again by exhibiting a particular proof system for all languages in QMA and constructing an extractor for it.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-56877-1_28) contains supplementary material, which is available to authorized users.

D. Micciancio and T. Ristenpart (Eds.): CRYPTO 2020, LNCS 12172, pp. 799–828, 2020.
https://doi.org/10.1007/978-3-030-56877-1_28

Keywords: Zero-knowledge · Non-interactive proof · Argument systems · QMA

1 Introduction

The paradigm of the interactive proof system is commonly studied in cryptography and in complexity theory. Intuitively speaking, an interactive proof system is a protocol in which an *unbounded* prover attempts to convince an *efficient* verifier that some problem instance x is in some language L. The verifier represents an entity less computationally powerful or less informed than the prover; the prover holds some knowledge that the verifier does not (namely, that $x \in L$), and the prover attempts to convince the verifier of this knowledge. We say that there is an interactive proof system *for a language* L if the following two conditions are satisfied. Firstly, for any $x \in L$, there must exist a prover (the 'honest' prover) which causes the (honest) verifier to accept in the protocol with high probability; and secondly, for any $x \notin L$, there is no prover which can cause the honest verifier to accept, except with some small probability. These two conditions are commonly referred to as the 'completeness' and 'soundness' conditions. We can also consider a relaxed soundness condition where, when $x \notin L$, we require only that it be computationally intractable (rather than impossible) to cause the verifier to accept. A protocol satisfying this relaxed soundness condition, and which has an efficient honest prover, is known as an interactive *argument* system.

Some interactive proof and argument systems satisfy a third property known as *zero-knowledge* [GMR85], which captures the informal notion that the verifier (even a dishonest verifier) 'learns no new information' from an interaction with the honest prover, except for the information that $x \in L$. This idea is formalised through a *simulator*, which has the same computational powers as the verifier V does, and can output transcripts that (for x such that $x \in L$) are indistinguishable from transcripts arising from interactions between V and the honest prover. As such, V intuitively 'learns nothing', because whatever it might have learned from a transcript it could equally have generated by itself. The property of zero-knowledge can be *perfect* (PZK), *statistical* (SZK) or *computational* (CZK). The difference between these three definitions is the extent to which simulated transcripts are indistinguishable from real ones. In a PZK protocol, the simulator's output distribution is *identical* to the distribution of transcripts that the honest prover and (potentially dishonest) verifier generate when $x \in L$. In SZK, the two distributions have negligible statistical distance, and in CZK, they are computationally indistinguishable. In this work we will primarily be concerned with CZK.

A *non-interactive* proof system (or argument system) is a protocol in which the prover and the verifier exchange only a single message that depends on the problem instance x. (In general, an instance-independent setup phase may be allowed in which the prover and verifier communicate, with each other or with a trusted third party, in order to establish shared state that is used during the protocol execution proper. We discuss this setup phase in more detail in the following paragraph.) Non-interactive zero-knowledge (NIZK) proofs and arguments have seen widespread application in classical cryptography, often in venues where their interactive counterparts would be impracticable—including, notably, in CCA-secure public-key cryptosystems [NY90,Sah99], digital signature schemes [BG90, CP92,BMW03], verifiable delegated computation [PHGR13] and, recently, a number of blockchain constructions [GGPR13,Com14,Lab17]. A particularly attractive feature of classical NIZK systems is that they can be amplified *in parallel* to achieve better security parameters [BDSMP91], which is in general not true of their interactive (private-coin) counterparts.

It is known [GO94] that NIZK proofs and arguments in the *standard model* (namely, the model where the only assumption is that adversarial entities are computationally efficient) exist only for languages in BPP. As such, in order to construct NIZK protocols for more interesting languages, it is customary to consider *extended* cryptographic models. Examples of these include the *common reference string* (CRS) model, in which the verifier and the prover are assumed to begin the protocol sharing access to a common string sampled from a specified distribution; and the *random oracle* (RO) model, in which prover and verifier have access to an efficiently evaluable function that behaves like a function sampled uniformly at random from the set of possible functions with some specified, and finite, domain and range. In these extended models, and under certain computational hardness assumptions, non-interactive computational zero-knowledge proof systems for all languages in NP are known. For instance, Blum, Santis, Micali and Persiano [BDSMP91] showed in 1990 that NIZK proofs for all languages in NP exist in the CRS model, assuming that the problem of quadratic residuosity is computationally intractable.

At this point, a natural question arises: what happens in the *quantum* setting? Ever since Shor's algorithm for factoring [Sho95] was published in 1995, it has been understood that the introduction of quantum computers would render a wide range of cryptographic protocols insecure. For example, quadratic residuosity is known to be solvable in polynomial time by quantum computers. Given that this is so, it is natural to ask the following question: in the presence of quantum adversaries, is it still possible to obtain proof systems for all languages in NP that are complete and sound, and if it is, in which extended models is it feasible? This question has been studied in recent years. For example, Unruh showed in [Unr15] that quantum-resistant NIZK proof systems for all languages in NP exist in the quantum random oracle (QRO) model, a quantum generalisation of the random oracle model. More recently, Peikert and Shiehian [PS19] achieved a more direct analogue of Blum et al.'s result, by showing that NIZK proofs for all languages in NP exist in the CRS model, assuming that learning with errors (LWE)—a problem believed to be difficult for quantum computers—is computationally intractable.[1]

However, the advent of large-scale quantum computers would not only render some cryptosystems insecure; it would also provide us with computational powers that extend those of our current classical machines, and give rise to new cryptographic tasks that were never considered in the classical literature. A second natural question which arises in the presence of quantum computers is the following: in which models is it possible to obtain a NIZK proof or argument system not only for all languages in NP, but for all languages in 'quantum NP' (i.e. QMA)? Loosely speaking, NIZK protocols for NP languages allow the prover to prove any statement that can be checked efficiently by a classical verifier who is given a classical witness. A NIZK protocol for QMA languages would, analogously, allow the prover to prove to the verifier (in a non-interactive, zero-knowledge way) the veracity of statements that require a quantum witness and quantum computing power to check. To our knowledge, the question of achieving NIZK protocols for QMA has not yet been studied. In 2016, Broadbent, Ji, Song and Watrous [BJSW16] exhibited a zero-knowledge proof system for QMA with an efficient honest prover, but their protocol requires both quantum and classical interaction.

In this work, our first contribution is to propose a non-interactive (computational) zero-knowledge argument system for all languages in QMA, based on the hardness of LWE, in which both verifier and prover are quantum polynomial time. The model we consider is the CRS (common reference string) model, augmented by a single message of (quantum) preprocessing. (The preprocessing consists of an instance-independent quantum message from the verifier to the prover.) The post-setup single message that the prover sends to the verifier, after it receives the witness, is classical; the post-setup verifier is also entirely classical; and, if we allow the prover and verifier to share EPR pairs *a priori*, as in a model previously considered by Kobayashi [Kob02], we can also make the verifier's preprocessing message classical. Like classical NIZK protocols, our protocol shows itself to be receptive to parallel repetition (see Sect. 2.3 of the supplementary material), which allows us to amplify soundness concurrently without affecting zero-knowledge. Our model and our assumptions are relatively standard ones which can be fruitfully compared with those which have been studied in the classical setting. As such, this result provides an early benchmark of the kinds of assumptions under which NIZK can be achieved for languages in QMA.

An example of an application in which the unique properties of our protocol might be useful is the setting of *verifiable delegated computation*, in which a prover (who is generally a server to whom a client, the verifier, has delegated

[1] Peikert and Shiehan construct, based on LWE, a NI(C)ZK proof system in the common *reference* string model, and a NI(S)ZK argument system in the common *random* string model. They do not explicitly consider the applications of either result to the quantum setting. We show, however, for our own purposes, that the latter of these results generalises to quantum adversaries. In other words, we show (in Sect. 1.3 of the Supplementary Material) that the Peikert-Shiehan NIZK *argument* system in the common *random* string model is adaptively sound against quantum adversaries and adaptively (quantum computational) zero-knowledge.

a quantum task) wishes to prove to the verifier a statement about a history state representing a certain computation. Suppose that the prover and the verifier complete the setup phase of our protocol when the delegation occurs. After the setup phase is complete, *the verifier does not need to preserve any quantum information*, meaning that it could perform the setup phase using borrowed quantum resources, and thereafter return to the classical world. When it receives the prover's single-message zero-knowledge proof, the verifier can verify its delegated computation without performing any additional quantum operations—a property that our protocol shares with protocols that have purely classical verification, such as Mahadev's classical-verifier argument system for QMA [Mah18]. An additional advantage of our protocol, however, is that the server can free the quantum memory associated with the verifier's computation *immediately* after the computation terminates, rather than holding the history state until the verifier is available to perform the verification.

Our second contribution is to show that our protocol also satisfies a notion of *argument of quantum knowledge*. In the classical setting, some proof systems and argument systems for NP languages satisfy a stronger notion of soundness wherein a witness can be *extracted* from any prover P who convinces the verifier to accept with high probability. More formally, in such a setting, there is an *extractor* machine which—given black-box access to any P who convinces the verifier to accept with high probability (on the input x)—is able to efficiently compute a witness w that testifies that the problem instance x is in the language L. Such protocols are known as *proofs* and *arguments of knowledge* (PoK and AoK). Intuitively speaking, the notion of PoK/AoK is a framework for describing situations where the prover is not necessarily more powerful, but only *better informed*, than the verifier. In these situations, the prover possesses knowledge (the witness w, which could represent a password or some other form of private information) that the verifier does not; and the prover wishes to convince the verifier, possibly in a zero-knowledge way (i.e. without revealing sensitive information), that it indeed 'knows' or 'possesses' the witness w (so that it might, for example, be granted access to its password-protected files, or cash a quantum cheque). The idea of a machine 'knowing' some witness w is formalised by the existence of the extractor.

Until now, the witness w has always been classical, and the notion of a proof of *quantum* knowledge (PoQK) has not been formally defined or studied. In this paper, we formulate a definition for a PoQK that is analogous to the classical definition of a PoK,[2] and we exhibit a protocol that is an (interactive) PoQK for any language in QMA.[3] We also introduce the notion of an *argument of quantum knowledge* (AoQK), and we prove that our NIZK protocol for QMA is (under this definition) a zero-knowledge argument of quantum knowledge. We present our definitions of PoQK and AoQK in Sect. 2.4.

There are two main difficulties in extending the classical notion of a PoK to the quantum setting. The first is that we must precisely specify how the extractor

[2] This definition is joint work with Broadbent and Grilo.

[3] This result is also obtained in independent and concurrent work by Broadbent and Grilo [BG19].

should be permitted to interact with the successful (quantum) prover. For this, we borrow the formalism of quantum interactive machines that Unruh [Unr12] uses in defining quantum proofs of *classical* knowledge. The second difficulty is to give an appropriate definition of success for the extractor. In the classical setting, the NP relation R which defines the set of witnesses w for a problem instance x is binary: a string w is either a witness or it is not. In the quantum setting, on the other hand—unlike in the classical case, in which any witness is as good as any other—different witnesses might be accepted with different probabilities by some verification circuit Q under consideration. In other words, some witnesses may be of better 'quality' than others. In addition, because QMA is a probabilistic class, the choice of Q (which is analogous to the choice of the NP relation R) is more obviously ambiguous than it is in the classical case. Different (and equally valid) choices of verifiers Q for a particular language $L \in$ QMA might have different probabilities of accepting a candidate witness ρ on a particular instance x. In our definition, we define a 'QMA relation' with respect to a fixed choice of verifying circuit (family) Q; we define the 'quality' of a candidate witness ρ for x to be the probability that Q accepts (x, ρ); and we require that the successful extractor returns a witness whose quality lies strictly above the soundness parameter for the QMA relation.

The Interactive Protocol from [BJSW16]

Our protocol is inspired by the protocol exhibited in [BJSW16], which gives a zero-knowledge (interactive) proof system for any language in QMA. The [BJSW16] protocol can be summarized as follows. (For a more detailed exposition, see Sect. 2.2.)

1. The verifier and the prover begin with an instance x of some interesting problem, the latter of which is represented by a (promise) language $L = (L_{yes}, L_{no}) \in$ QMA. The prover wishes to prove to the verifier that $x \in L_{yes}$. The first step is to map x to an instance H of the QMA-complete *local Clifford Hamiltonian problem.* In the case that x is a yes instance, i.e. $x \in L_{yes}$, the prover, who receives a witness state $|\Phi\rangle$ for x as auxiliary input, performs the efficient transformation that turns the witness $|\Phi\rangle$ for x into a witness $|\Psi\rangle$ for H. (The chief property that witnesses $|\Psi\rangle$ for H have is that $\langle\Psi| H |\Psi\rangle$ is *small*—smaller than a certain threshold—which, rephrased in physics terminology, means that $|\Psi\rangle$ has *low energy with respect to* H.) The prover then sends an *encoding* of $|\Psi\rangle$ to the verifier (under a specified quantum authentication code which doubly functions as an encryption scheme). The prover also *commits* to the secret key of the authentication code.
2. The Clifford Hamiltonian H to which x has been mapped can be written as a sum of polynomially many terms of the form $C^* |0^k\rangle \langle 0^k| C$, where C is a Clifford unitary. (This is the origin of the name 'Clifford Hamiltonian'.) The verifier chooses a string r uniformly at random. r plays a role analogous to that of the verifier's choice of edge to check in the 3-colouring zero-knowledge protocol introduced by [GMR85]: intuitively, r determines the verifier's challenge to the prover. Each r corresponds to one of the terms $C_r^* |0^k\rangle \langle 0^k| C_r$ of the Clifford Hamiltonian.

The verifier then measures the term $C_r^* \left|0^k\right\rangle \left\langle 0^k\right| C_r$ on the encoded witness (this can be done 'homomorphically' through the encoding). The outcome z obtained by the verifier can be thought of as an encoding of the true measurement outcome, the latter of which should be *small* (i.e. correspond to low energy) if $|\Psi\rangle$ is a true witness. The verifier sends z (its measurement outcomes) and r (its choice of Hamiltonian term) back the prover.

3. Finally, using a zero-knowledge NP proof system,[4] the prover provides an (interactive) ZK proof for the following NP statement: there *exists* an opening to its earlier (perfectly binding) commitment such that, if the verifier had the opened encoding keys, it *would* accept. This is an NP statement because the witness string is the encoding keys. Proving that the verifier 'would accept' amounts to proving that the verifier's measurement outcomes z, decoded under the keys which were committed to earlier, would correspond to a low-energy outcome. Because the proof that the prover provides is zero-knowledge, the verifier learns nothing substantial from this exchange, but it becomes convinced that it should accept.

In the protocol from [BJSW16], it is critical to soundness that the prover sends the encoding of the witness to the verifier *before* the verifier chooses r. The zero-knowledge property holds because the encoding that the prover applies to the witness state functions like an authenticated encryption scheme: its encryption-like properties prevent the verifier from learning anything substantial about the witness while handling the encoded state, and its authentication code–like properties ensure that the verifier cannot deviate very far from its honest behaviour.

Our Non-interactive Protocol

We wish to make the protocol from [BJSW16] *non-interactive*. To start with, we can replace the prover's proof in step 3 with a NIZK proof in the CRS model. NIZK proofs for all languages in NP have recently been shown to exist [CLW19,PS19] based on the hardness of LWE only, and we prove that the Peikert-Shiehian construction from [PS19] remains secure (i.e. quantum computationally sound and zero-knowledge) against quantum adversaries, assuming that LWE is quantum computationally intractable. However, the more substantial obstacle to making the [BJSW16] protocol non-interactive is the following: in order to do away with the verifier's message in step 2, it seems that the prover would have to somehow *predict* z (the verifier's measurement outcomes) and send a NIZK proof corresponding to this z. Unfortunately, in order for the authentication code to work, the number of possible outcomes z has to be exponentially large (and thus the prover cannot provide a NIZK proof of consistency for each possible outcome). Even allowing for an instance-independent preprocessing step between the verifier and the prover, it is unclear how this impasse could be resolved.

Our first main idea is to use *quantum teleportation*. We add an instance-independent preprocessing step in which the verifier creates a number of EPR

[4] It is known that there are quantumly sound and quantumly zero-knowledge proof systems for NP: see [Wat09].

pairs and sends half of each to the prover. We then have the verifier (prematurely) make her measurement from step 2 *during the preprocessing step* (and hence *independently of the instance!*), and send the measurement outcomes z to the prover. Once x is revealed, the prover *teleports* the encoded witness to the verifier, and sends the verifier the teleportation outcomes d, along with a commitment to his encoding keys. The prover then provides an NIZK proof of an opening to the committed keys such that d, z and the encoding keys are consistent with a low-energy outcome. The hope is that, because the prover's and the verifier's actions commute (at least when the prover is honest), this protocol will be, in some sense, equivalent to one where the prover firstly teleports the witness, *then* the verifier makes the measurements, and finally the prover sends an NIZK proof. This latter protocol would be essentially equivalent to the [BJSW16] protocol.

There are three main issues with this strategy:

1. In the preprocessing step, the verifier does not yet know what the instance x (and hence what the Clifford Hamiltonian) is. Thus, she cannot measure the term $C_r^* \left|0^k\right\rangle \left\langle 0^k\right| C_r$, as she would have done in what we have called step 2 of the protocol from [BJSW16].
2. The second issue is that the verifier cannot communicate her choice of r in the preprocessing step in the clear. If she does, the prover will easily be able to cheat by teleporting a state that passes the check for the rth Hamiltonian term, but that would not pass the check for any other term.
3. The third issue is a bit more subtle. If the prover knows the verifier's measurement outcomes z before he teleports the witness state to the verifier, he can misreport the teleportation outcomes d, and make a clever choice of d such that d, z and the committed keys are consistent with a low-energy outcome even when he does not possess a genuine witness.

The first issue is resolved by considering the (instance-independent) verifying circuit Q *for the* QMA *language* L (recall that Q takes as input both an instance x and a witness state), and mapping Q itself to a Clifford Hamiltonian $H(Q)$. (For comparison, in the protocol from [BJSW16], it is the circuit $Q(x, \cdot)$ which is mapped to a Clifford Hamiltonian.) In the instance-dependent step, the prover will be asked to teleport a "history state" corresponding to the execution of the circuit Q on input $(x, |\Psi\rangle)$, where $|\Psi\rangle$ is a witness for the instance x. In the preprocessing step, the verifier will measure a uniformly random term from $H(Q)$, and will also perform a special measurement (with some probability) which is meant to certify that the prover put the correct instance x into Q when it was creating the history state. Of course, the verifier does not know x at the time of this measurement, but she will know x at the point where she needs to verify the prover's NIZK proof.

Our second main idea, which addresses the second and the third issues above (at the price of downgrading our proof system to an argument system), is to have the prover *compute his NIZK proof homomorphically.* During the preprocessing step, we have the verifier send the prover a (computationally hiding) commitment σ to her choice of r; and, in addition, we ask the verifier to send the prover a *homomorphic encryption* of r, of the randomness s used to commit to σ, and of her measurement outcomes z. At the beginning of the instance-dependent step, the prover receives a witness $|\Psi\rangle$ for the instance x. During the instance-dependent step, and after having received the verifier's ciphertexts in the preprocessing step, we ask the prover firstly to commit to some choice of encoding keys, and then to teleport to the verifier (an encoding of) the history state corresponding to the execution of Q on input $(x, |\Psi\rangle)$. Let d be the outcome of the teleportation measurements. After the prover has committed to his encoding keys, we ask the prover to homomorphically encrypt d and his encoding keys, and homomorphically run the following circuit: check that r, s is a valid opening to σ, and (using the properties of the authentication code) check also that the verifier performed the honest measurement during preprocessing. If all the checks pass, then the prover *homomorphically* computes an NIZK proof that there exist encoding keys consistent with his commitment such that these keys, together with r, z, d, indicate that the verifier's measurement result was a low-energy outcome. The homomorphic encryption safeguards the verifier against a malicious prover who may attempt to take advantage of knowing r, or of the freedom to cleverly choose d, in order to pass in the protocol without holding a genuine witness.

In summary, the structure of our protocol is as follows. Let Q be a QMA verification circuit, and let $H(Q)$ be the Clifford Hamiltonian obtained from Q by performing a circuit-to-Clifford-Hamiltonian reduction.

1. *(preprocessing step)* The verifier creates a (sufficiently large) number of EPR pairs, and divides them into 'her halves' and 'the prover's halves'. She interprets her halves as the qubits making up (an encoding of) a history state generated from an evaluation of the circuit Q. Then, the verifier samples r (her 'challenge') uniformly at random, and according to its value, does one of two things: either she measures a uniformly random term of $H(Q)$ on 'her halves' of the EPR pairs, or she makes a special measurement (on her halves of the EPR pairs) whose results will allow her later to verify that the circuit Q was evaluated on the correct instance x. Following this, the verifier samples a public-key, secret-key pair (pk, sk) for a homomorphic encryption scheme. She sends the prover:
   (a) pk;
   (b) the 'prover's halves' of the EPR pairs;
   (c) a commitment to her choice of challenge r;
   (d) homomorphic encryptions of
      i. r,
      ii. the randomness s used in the commitment, and
      iii. the measurement outcomes z.
2. *(instance-dependent step)* Upon receiving x, and a witness $|\Psi\rangle$, the prover computes the appropriate history state, and samples encoding keys. Then, he teleports an encoding of the history state to the verifier using the half EPR pairs that he previously received from her. Notice that the verifier has already measured the other half of the EPR pairs on her side during the

preprocessing step: hence the encoded history state is not being physically teleported. Nonetheless, because the measurements of the verifier and the prover commute, the net effect in terms of measurement outcome statistics is the same. Let d be the teleportation measurement outcomes. The prover sends to the verifier:

(a) d;
(b) a commitment σ to his encoding keys;
(c) a homomorphic encryption of a NIZK proof (homomorphically computed) of the existence of an opening to σ such that the opened keys, together with d, z, r, are consistent with a low-energy outcome.

Upon receiving d, σ, and an encrypted proof $\tilde{\pi}$ from the prover, the verifier decrypts $\tilde{\pi}$ to obtain π, and checks that π is a valid proof and that it is consistent with d and σ (i.e the d and σ from steps (a) and (b) are the same that appear in the statement being proven).

Analysis

Our protocol is a non-interactive, zero-knowledge argument system in the CRS model with a one-message preprocessing step. It is straightforward to see that the protocol satisfies completeness.

Intuitively, soundness follows from the fact that the encryptions the prover receives in the preprocessing step should be indistinguishable (assuming the prover is computationally bounded) from encryptions of the zero string. As such, the encryptions of z, r, s (and the commitment to r) cannot possibly be helping the prover in guessing r or in selecting a false teleportation measurement outcome d' which makes z, r, d' and the authentication keys consistent with a low-energy outcome. Soundness then essentially reduces to soundness of the protocol in [BJSW16].

The zero-knowledge property follows largely from the properties of the protocol in [BJSW16] that allowed Broadbent, Ji, Song and Watrous to achieve zero-knowledge. One key difference is that, in order to avoid rewinding the (quantum) verifier, the authors of [BJSW16] use the properties of an *interactive coin-flipping protocol* to allow the efficient simulator to recover the string r (recall that r determines the verifier's challenge) with probability 1. (The traditional alternative to this strategy is to have the simulator guess r, and rewind the verifier if it guessed incorrectly in order to guess again. This is typical in classical proofs of zero-knowledge [GMR85]. However, because quantum rewinding [Wat09] is more delicate, the authors of [BJSW16] avoid it for simplicity.) As our protocol is non-interactive, we are unable to take the same approach. Instead, we ask the verifier to choose r and commit to it using a commitment scheme with a property we call *extractability*. Intuitively, extractability means that the commitment scheme *takes a public key determined by the CRS*. We then show that the simulator can efficiently recover r from the verifier's commitment by taking advantage of the CRS. For an LWE-based extractable commitment scheme, see Sect. 1.2 of the Supplementary Material.

Another subtlety, unique to homomorphic encryption, is that the verifier may learn something about the homomorphic computations performed by the prover (and hence possibly about the encoding keys) by looking at the *encryption randomness* in the encryption (of an NIZK proof) that the prover sends the verifier. (Recall that the verifier possesses the decryption key *sk* for the homomorphic encryption scheme.) This leads us to require the use of a fully homomorphic encryption scheme which satisfies the property of *circuit privacy*. For a definition of this property, see Sect. 1.2 of the Supplementary Material.

Remark 1. The technique we proposed to remove interaction from the protocol of [BJSW16] is based on two main ingredients: the use of quantum teleportation, which allows the verifier to *anticipate* her measurements of the state she receives from the prover in the instance-dependent step, and the use of classical homomorphic encryption to allow the prover to demonstrate (homomorphically) that he has performed a certain computation correctly. These two ingredients work in tandem to ensure that the soundness and the zero-knowledge property of the [BJSW16] protocol are preserved. We believe that this technique could find use more broadly. In particular, it may be applicable as a general (soundness and zero-knowledge preserving) transformation to any interactive proof system for QMA with an efficient honest prover. We leave a more thorough investigation of this as a direction for future work.

A Non-interactive Argument of Quantum Knowledge

One desirable feature of our non-interactive argument system is that it is also an *argument of quantum knowledge*. As we mentioned earlier, one of our contributions is to generalize the definitions of PoKs and AoKs for NP-relations to definitions of PoKs and AoKs for *QMA relations*. In the latter setting, the prover wishes to convince the verifier that he 'knows' or 'possesses' the quantum witness for an instance of a QMA problem. In order to show that our protocol satisfies this additional property, we need to exhibit an extractor that, for any yes instance x, and given quantum oracle access to any prover that is accepted with high probability in our protocol, outputs a quantum state which is a witness for x. In Sect. 6, we explicitly construct such an extractor K for our non-interactive protocol. The intuition is the following. K (the extractor) has oracle access to a prover P^*, and it simulates an execution of the protocol between P^* and the honest verifier V. We show that, if P^* is accepted in our protocol with sufficiently high probability, then it must teleport to V (and hence to K) the *encoding* $\tilde{\rho}$ of a witness state, and a commitment σ to the encoding keys. If K knew the encoding keys, it would be able to decode $\tilde{\rho}$, but it is not clear *a priori* how K could obtain such keys. Crucially, the same feature of our protocol that allows the *zero-knowledge* simulator to extract r from the verifier's commitment to r also plays in K's favour: when K simulates an execution of the protocol, it samples a common reference string which is given to both V and P^*, and in our protocol, the CRS contains a public key which P^* uses to make his commitment. As such, in order to extract a witness from P^*, the extractor samples a CRS containing a public key pk for which it knows the corresponding secret key sk, and provides this particular CRS as input to P^*. Then, when K receives $\tilde{\rho}$ and σ from P^*, it is able to extract the committed keys from σ, and use these to decode $\tilde{\rho}$.

An Interactive Proof of Quantum Knowledge

Our non-interactive protocol is an *argument system*, which means that it is sound only against computationally bounded provers. In Sect. 7, we introduce a separate but complementary result to our NIZK argument (of knowledge) for QMA by showing that the zero-knowledge proof system for QMA exhibited in [BJSW16] (with some minor modifications) is also a *proof of quantum knowledge*.

2 Preliminaries

2.1 Notation

For an integer $\ell \geq 1$, $[\ell]$ denotes the set $\{1, \ldots, \ell\}$. We use $\mathrm{poly}(n)$ and $\mathrm{negl}(n)$ to denote an arbitrary polynomial and negligible function of n respectively (a negligible function f is any computable function such that $f(n)q(n) \rightarrow_{n\rightarrow\infty} 0$ for all polynomials q). For an integer $d \geq 1$, $\mathrm{D}(\mathbb{C}^d)$ denotes the set of density matrices on $\mathbb{C}^d$, i.e. positive semidefinite ρ on $\mathbb{C}^d$ such that $\mathrm{Tr}(\rho) = 1$. For a set S and an element $s \in S$, we write $s \overset{\$}{\leftarrow} S$ to mean that s is sampled uniformly at random from S. For an integer l, we denote by $\{0,1\}^{\leq l}$ the set of binary strings of length at most l. We use the notation S_N to denote the set of all permutations of a set of N elements.

We use the terminology PPT for *probabilistic polynomial time* and QPT for *quantum polynomial time* to describe algorithms.

2.2 The [BJSW16] Protocol

The following exposition is taken from [VZ19]. For an introduction to the Local Hamiltonian problem, and the associated notation, we refer the reader to the Supplementary Material.

In [BJSW16], Broadbent, Ji, Song and Watrous describe a protocol involving a quantum polynomial-time verifier and an unbounded prover, interacting quantumly, which constitutes a zero-knowledge proof system for languages in QMA. (Although it is sound against arbitrary provers, the system in fact only requires an honest prover who is provided with a single witness state to perform quantum polynomial-time computations.) We summarise the steps of their protocol below. For details and fuller explanations, we refer the reader to [BJSW16, Section 3].

Notation. Let L be any language in QMA. For a definition of the *k-local Clifford Hamiltonian problem*, see [BJSW16, Section 2] (this is the defined analogously to the k-local Hamiltonian problem, except that the Hamiltonian instance consists of Clifford terms, as introduced in the previous subsection). The k-local Clifford Hamiltonian problem (with exponentially small ground state energy) is QMA-complete for $k = 5$; therefore, for all possible inputs x, there exists a 5-local Clifford Hamiltonian H (which can be computed efficiently from x) whose terms are all operators of the form $C^* |0^k\rangle \langle 0^k| C$ for some Clifford operator C, and such that

- if $x \in L$, the ground energy of H is $\leq 2^{-p}$,
- if $x \notin L$, the ground energy of H is $\geq \frac{1}{q}$,

for some positive integers p and q which are bounded above by polynomials in $|x|$.

Parties. The proof system involves a *verifier*, who implements a quantum polynomial-time procedure; a *prover*, who is unbounded, but who is only required by the protocol to implement a quantum polynomial-time procedure. The verifier and the prover communicate quantumly.

Inputs

1. Input to the verifier: (a) The Hamiltonian H. (b) A quantum computationally concealing, perfectly binding (classical) commitment protocol. In this section, we refer to the commitment algorithm from this protocol as commit; $\mathsf{commit}(\mu, s)$ takes as input a message μ and a random string s and produces

Auth.Enc:

Parameters: $N(\cdot)$, a polynomially bounded function in $|x|$. (N functions as a security parameter.)

Input: An m-qubit state ρ.

The prover firstly applies a concatenated Steane code (which maps every one qubit to $N(|x|)$ qubits) to each qubit in ρ. (For details on the concatenated Steane code, see [BJSW16, Appendix A.6]. It will be important to Broadbent et al.'s purposes, and ours, that this code admits transversal applications of Clifford operations.) It then executes the following steps:

(a) Concatenate N trap qubits to the end of each logical qubit (alternatively, to the end of each N-qubit block) in the result of applying the concatenated Steane code to ρ. Each trap qubit is initialised uniformly at random to one of $|0\rangle, |+\rangle, |+_y\rangle$. ($|+_y\rangle$ here refers to the state $\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$.) Denote the string that records the choices of trap qubits for all m logical qubits by $t = t_1, \ldots, t_N \in \{|0\rangle, |+\rangle, |+_y\rangle\}^{mN}$.

(b) Permute each $2N$-tuple of qubits in the result of (a) according to a uniformly random permutation $\pi \in S_{2N}$. (Note that the same permutation π is applied to every $2N$-tuple.)

(c) Apply a Pauli one-time pad $X^a Z^b$, for uniformly random $a, b \in \{0,1\}^{2mN}$, to the entire $2mN$-qubit state.

Fig. 1. The authentication code

a commitment string z. (c) A proof system for NP sound against arbitrary quantum provers.

2. Input to the prover: (a) The Hamiltonian H. (b) The n-qubit quantum state ρ, where ρ is a ground state of the Hamiltonian H. (c) A quantum computationally concealing, perfectly binding (classical) commitment protocol. (d) A proof system for NP sound against arbitrary quantum provers.

Protocol

1. *The prover's encoding step.* The prover applies the following encoding to the witness state ρ.
   We refer to t, π, a, b as 'the authentication keys' or 'the encoding keys'.
   The prover's encoding applied to ρ is denoted by $E(\rho)$, and the procedure E is fully determined by the encoding key (t, π, a, b) which the prover chose to use. At this point, the prover sends the state $E(\rho)$ to the verifier, along with a commitment (using some perfectly binding, computationally concealing classical commitment protocol) to the tuple (π, a, b). (A commitment to the sequence of trap qubits t is unnecessary because, in a sense, the trap qubits exist only to check the verifier.) Let the prover's commitment string be denoted z.
2. *Coin-flipping protocol.* The prover and the verifier execute a coin-flipping protocol, choosing a string r of fixed length uniformly at random. This random string r determines a local Hamiltonian term $H_r = C_r^* |0^k\rangle \langle 0^k| C_r$ that is to be tested. (This step can be implemented [DL09] using the same classical commitment protocol that the prover employed in the previous step.)
3. *Verifier's challenge.* The verifier applies the Clifford C_r transversally to the qubits on which the k-local Hamiltonian term H_r acts nontrivially, and measures them in the standard basis. It then sends the measurement results $u_{i_1}, \ldots, u_{i_k}$ which it obtained to the prover. (Each u_i is a $2N$-bit string, and $i_1, \ldots, i_k$ are the indices of the logical qubits on which the term H_r acts nontrivially.)
4. *Prover's response.* The prover receives the verifier's measurement results u, and firstly checks whether they cause a predicate $\tilde{Q}(t, \pi, a, b, r, u)$ to be satisfied. (We will explain the predicate $\tilde{Q}$ in more detail shortly. Intuitively, $\tilde{Q}$ is satisfied if and only if both verifier *and* prover behaved honestly. Note that we have used the notation $\tilde{Q}$ to represent this predicate, while the authors of [BJSW16] simply call it Q. We add the tilde in order to differentiate their predicate from our predicate Q, the latter of which we define in Definition 13.)
   If $\tilde{Q}$ is not satisfied, the prover aborts, causing the verifier to reject. If $\tilde{Q}$ is satisfied, then the prover proves to the verifier, using an NP zero-knowledge protocol, that there exists randomness s_P and an encoding key (t, π, a, b) such that $z = \mathsf{commit}((\pi, a, b), s_P)$ and $\tilde{Q}(t, \pi, a, b, r, u) = 1$.

Here $\tilde{Q}$ represents the prover's check after it has update the one-time pad keys based on the Clifford C_r, and reversed the effects of the one-time pad keys. We refer the reader to [BJSW16] for a formal definition of $\tilde{Q}$.

2.3 Argument Systems

Interactive Quantum Machines. The definitions of *interactive quantum machines*, their *executions* and *oracle access* to an interactive quantum machine are taken from [Unr12], and are omitted from this version due to space constraints.

Oracle Access to an Interactive Quantum Machine. We say that a quantum algorithm A has oracle access to an interactive quantum machine M (and we write this as A^M, or sometimes $A^{|M\rangle}$ to emphasize that M is a quantum machine and that oracle access includes the ability to apply the inverse of M) to mean the following. Besides the security parameter and its own classical input x, we allow A to execute the quantum circuit $M_{\mu x}$ specifying M, and its inverse (these act on the an "internal" register S and on a "network" register N of M). Moreover, we allow A to provide and read messages from M (formally, we allow A to act freely on the network register N). We do not allow A to act on the internal register S of M, except via $M_{\mu x}$ or its inverse.

Argument Systems with Setup. First we define the kinds of relations that underlie our argument systems. Classically, a relation over finite sets $\mathcal{X} \times \mathcal{Y}$ is a subset $R \subseteq \mathcal{X} \times \mathcal{Y}$. An NP relation $R = \{(x, w) : V_{|x|}(x, w) = 1\}$ has the additional property that given any $x \in \mathcal{X}$ and $w \in \mathcal{Y}$, the claim that $(x, w) \in R$ can be verified by a uniformly generated family of circuits $V = \{V_n\}$ (the "verifier").

In the quantum case the "input" x (the first argument to the relation) remains classical, but the "witness" w (the second argument) can be a quantum state $|\psi\rangle$. Before we give our definition of a QMA relation we introduce some notation. Fix a uniformly generated family of polynomial-size quantum circuits $Q = \{Q_n\}_{n\in\mathbb{N}}$ such that for every n, Q_n takes as input a string $x \in \{0,1\}^n$ and a quantum state σ on $p(n)$ qubits (for some polynomial $p(n)$) and returns a single bit as output. For any $0 \le \gamma \le 1$ define

$$R_{Q,\gamma} = \bigcup_{n\in\mathbb{N}} \left\{(x,\sigma) \in \{0,1\}^n \times \mathrm{D}(\mathbb{C}^{p(n)}) \,\middle|\, \Pr(Q_n(x,\sigma) = 1) \ge \gamma\right\}$$

and

$$N_{Q,\gamma} = \bigcup_{n\in\mathbb{N}} \left\{x \in \{0,1\}^n \,\middle|\, \forall\sigma \in \mathrm{D}(\mathbb{C}^{p(n)}),\ \Pr(Q_n(x,\sigma) = 1) < \gamma\right\} .$$

Note the presence of the parameter γ, that quantifies the expected success probability for the verifier; γ can be thought of as a measure of the "quality" of a witness $|\psi\rangle$ (or mixture theoreof, as represented by the density matrix σ) that is sufficient for the witness to be acceptable with respect to the relation R.

Definition 1 (QMA relation). *A QMA* relation *is specified by triple (Q, α, β) where $Q = \{Q_n\}_{n\in\mathbb{N}}$ is a uniformly generated family of quantum circuits such* *that for every n, Q_n takes as input a string $x \in \{0,1\}^n$ and a quantum state $|\psi\rangle$ on $p(n)$ qubits and returns a single bit, and $\alpha, \beta : \mathbb{N} \to [0,1]$ are such that $\alpha(n) - \beta(n) \geq 1/p(n)$ for some polynomial p and all $n \in \mathbb{N}$. The QMA relation associated with (Q, α, β) is the pair of sets $R_{Q,\alpha}$ and $N_{Q,\beta}$.*

We say that a language $L = (L_{yes}, L_{no})$ is specified by a *QMA* relation (Q, α, β) *if*

$$L_{yes} \subseteq \bigcup_{n \in \mathbb{N}} \{x \in \{0,1\}^n | \exists \sigma \in \mathrm{D}(\mathbb{C}^{p(n)}) \ s.t. \ (x, \sigma) \in R_{Q,\alpha}\}, \tag{1}$$

and $L_{no} \subseteq N_{Q,\beta}$.

Note that in contrast to an NP relation, we define a QMA relation using two sets: the first set, $R_{Q,\alpha}$, is the set of (instance, witness) pairs that are deemed to form part of the relation. The second set, $N_{Q,\beta}$, is the set of instances that are deemed to be such that they are in relation to no witness. Some instances may lie in neither (the projection of) $R_{Q,\alpha}$ or $N_{Q,\beta}$; this is analogous to the necessity for a "promise" between the completeness and soundness parameters α and β in the definition of the class QMA, that do not appear in the definition of NP. In particular, note that, whenever $\alpha - \beta > 1/\operatorname{poly}(n)$, a language L that is specified by (Q, α, β) lies in QMA. Conversely, any language in QMA is specified by some QMA relation (of course such relation is not unique).

Definition 2 (protocol with setup). *A* protocol with setup *is a triple of interactive machines (S, P, V) with the following properties:*

1. *$S = \{S_{\mu n}\}_{\mu \in \mathbb{N}}$ depends on the security parameter μ and an instance size n, takes no input and returns a classical output in the message registers N_{SP} and N_{SV}. When the output in both registers is the same, we refer to it as "common reference string".*
2. *Each of P and V has two phases: $P = (P_1, P_2)$ and $V = (V_1, V_2)$. $P_1 = \{P_{1,\mu n}\}$ and $V_1 = \{V_{1,\mu n}\}$ are interactive machines that depend on the security parameter μ and an instance size parameter n, take a classical message input in register N_{SP} and N_{SV} respectively and return a quantum message as output in registers $\mathsf{N}_{P_1P_2}$ and $\mathsf{N}_{V_1V_2}$ respectively. $P_2 = \{P_{2,\mu n}\}$ and $V_2 = \{V_{2,\mu n}\}$ are interactive machines that depend on the security parameter μ and an input size n. V_2 takes as input the output of V_1, in register $\mathsf{N}_{V_1V_2}$, as well as an instance x such that $|x| = n$. P_2 takes as input the output of P_1, in register $\mathsf{N}_{P_1P_2}$, an instance x such that $|x| = n$, and a quantum state ρ. V_2 returns a single bit $b \in \{0,1\}$ as output, and P_2 returns no output. If $b = 1$ then we say that V accepts, and otherwise we say that it rejects.*

We refer to the first phase of P and V as the *preprocessing phase*, and to the second phase as the *instance-dependent phase*.

Definition 3 (argument system with completeness c and soundness s). *Let (Q, α, β) be a QMA relation and $s, c : \mathbb{N} \to [0,1]$. An* argument system *(with setup) for (Q, α, β), with completeness c and soundness s, is a protocol with setup (S, P, V) such that S, P, V are quantum polynomial-time and, in addition, the following hold:*

1. *(Completeness) For all* $(x, \rho) \in R_{Q,\alpha}$*, for all integer* μ*, the execution* $(S, P(x, \rho), V(x))$ *returns* 1 *with probability at least* $c(\mu)$*.*
2. *(Soundness) For all* $x \in N_{Q,\beta}$*, all integer* μ *and all polynomial-time* P^* *the execution* $(S, P^*(x), V(x))$ *returns* 1 *with probability at most* $s(\mu) + negl(\mu)$*.*

When the second phase of a protocol with setup (S, P, V) consists of a single message from P to V we refer to it as a *non-interactive* protocol with setup. If it is a an argument system with setup, we refer to it as a *non-interactive* argument system with setup. When the first phase involves some communication between P and V, we specify that it is a non-interactive argument system with setup *and preprocessing*. When S outputs a common reference string (as defined in 2), we refer to it as an argument system *with CRS setup* (possibly with preprocessing).

Note that Definition 3 requires that the execution $(S, P(x, \rho), V(x))$ returns 1 with probability at least $c(\mu)$. In the case of sequential or parallel repetition of a protocol, it may not be possible for the prover to succeed with a single copy of the witness ρ as input. In this case we may considering relaxing the definition as follows.

Definition 4 (Completeness of argument system with setup—alternative definition). *Let* Q^q *be the circuit that runs* Q *on* q *registers, and accepts if all executions accept. There exists a polynomial* $q > 0$*, such that for all* $(x, \rho) \in R_{Q^q,\alpha}$*, for all integers* μ*, the execution* $(S, P(x, \rho), V(x))$ *returns* 1 *with probability at least* $c(\mu)$*.*

We will clarify, whenever we refer to an argument system with setup, which definition we refer to.

Finally, we define the notion of *adaptive soundness*, which captures security against adversaries that are allowed to choose the common instance x *after* having carried out the preprocessing phase.

Definition 5 (Adaptive soundness). *An argument with setup* (S, P, V) *for a QMA relation* (Q, α, β) *has* adaptive soundness $s(\mu)$ *if for every QPT algorithm* $P^* = \{(P^*_{1,\mu n}, P^*_{2,\mu n})\}$*, for all* μ*,*

$$\Pr_{\substack{(\sigma_{PV})\leftarrow(S_{\mu n}, P^*_{1,\mu n}, V_{1,\mu n}),\\ (x,\tau)\leftarrow P^*_{2,\mu n}(\sigma_P)}} \big(x \in N_{Q,\beta} \wedge (P^*_{2,\mu n}(x, \tau), V_{2,\mu n}(x, \sigma_V)) = 1\big) \leq s(\mu) + negl(\mu).$$

The terminology that follows Definition 3 is modified in the natural way in the case of adaptive soundness.

2.4 Proofs and Arguments of Quantum Knowledge

The content of this subsection, as it pertains to *proofs of quantum knowledge*, was written in collaboration with Broadbent and Grilo, and appears with slight differences in [BG19].

A *Proof of Knowledge (PoK)* is an interactive proof system for some relation R such that if the verifier accepts some input x with high enough probability, then she is "convinced" that the prover "knows" some witness w such that $(x, w) \in R$. This notion is formalized by requiring the existence of an efficient *extractor* K that is able to return a witness for x when given oracle access to the prover (including the ability to rewind its actions, in the classical case).

Definition 6 (Classical Proof of Knowledge). *Let $R \subseteq \mathcal{X} \times \mathcal{Y}$ be a relation. A proof system (P, V) for R is a Proof of Knowledge for R with knowledge error κ if there exists a polynomial $p > 0$ and a polynomial-time machine K such that for any classical interactive machine P^*, any $\mu \in \mathbb{N}$, any polynomial $l > 0$, any instance $x \in \{0,1\}^n$ for $n = \mathrm{poly}(\mu)$ and any string y: if the execution $(P^*(x, y), V(x))$ returns 1 with probability $\varepsilon > \kappa(\mu)$, we have*

$$\Pr\left(\left(x, K^{P^*(x,y)}(x)\right) \in R\right) \geq p\left(\varepsilon - \kappa(\mu), \frac{1}{\mu}\right) - negl(\mu)$$

In this definition, y corresponds to the side information that P^* has, possibly including some w such that $(x, w) \in R$.

PoKs were originally defined only considering classical adversaries, and this notion was first studied in the quantum setting by Unruh [Unr12]. The first issue that arises in the quantum setting is to formalize the type of query that the extractor K is able to make. In order to do so, we assume that P^* always performs a fixed unitary operation U when invoked. Notice that this can be assumed without loss of generality since (i) we can always consider a purification of P^*, (ii) all measurements can be performed coherently, and (iii) P^* can keep track of the round of communication in some internal register and U can implicitly control on this value. Then, the quantum extractor K has oracle access to P^* in the sense that it may perform U and $U^\dagger$ on the message register and private register of P^*, but has no direct access to the latter. We denote the extractor K with such oracle access to P^* by $K^{|P^*(x,\rho)\rangle}$, where ρ is some (quantum) side information held by P^*.

Definition 7 (Quantum Proof of (Classical) Knowledge). *Let $R \subseteq \mathcal{X} \times \mathcal{Y}$ be a relation. A proof system (P, V) for R is a Quantum Proof of Knowledge for R with knowledge error κ if there exists a polynomial $p > 0$ and a quantum polynomial-time machine K such that for any quantum interactive machine P^*, any $\mu \in \mathbb{N}$, any polynomial $l > 0$, any instance $x \in \{0,1\}^n$ for $n = \mathrm{poly}(\mu)$ and any state ρ: if the execution $(P^*(x, \rho), V(x))$ returns 1 with probability $\varepsilon > \kappa(\mu)$, we have*

$$\Pr\left(\left(x, K^{|P^*(x,\rho)\rangle}(x)\right) \in R\right) \geq p\left(\varepsilon - \kappa(\mu), \frac{1}{\mu}\right).$$

Remark 2. In the fully classical case of 6, the extractor could repeat the procedure in sequence polynomially many times in order to increase the probability of a successful extraction (which, in Definitions 6 and 7, is allowed to be inverse-polynomially small in the security parameter). This is not known to be possible for a general quantum P^*, since the final measurement to extract the witness could possibly disturb the internal state of P^*, making it impossible to simulate the side information that P^* had originally in the subsequent simulations.

We finally move on to the full quantum setting, where we want a *Proof of Quantum Knowledge* (PoQK). Intuitively, at the end of the protocol, we would like the verifier to be 'convinced' that the prover 'has' a *quantum witness* for the input x. The main difference from Quantum Proofs of (classical) Knowledge is that in the case of QMA relations, as defined in Sect. 2.3, the notion of a witness is not as unambiguous as in the case of NP relations. We introduce a parameter q which quantifies the probability that the witness returned by the extractor makes the verifying circuit accept. We refer to this parameter as the "quality" of the PoQK. We also allow the extractor K to return a special symbol "$\perp$" in a designated portion of the output register, and we require that either the extractor returns "$\perp$" or it returns a witness of a certain quality. Formally, we assume that the output of the extractor is measured according to $\{|\perp\rangle\langle\perp|, I-|\perp\rangle\langle\perp|\}$. We ask that the outcome of this measurement be the latter with at least inverse-polynomial probability, and that, conditioned on the latter outcome, the post-measurement state be a witness (of a certain quality).

Definition 8 (Proof of Quantum Knowledge). *Let (Q, α, β) be a QMA relation. A proof system (P, V) is a Proof of Quantum Knowledge for (Q, α, β) with knowledge error κ and quality $q > \beta$, if there exists a polynomial $p > 0$ and a quantum polynomial-time machine K such that for any quantum interactive machine P^*, any $\mu \in \mathbb{N}$, any polynomial $l > 0$, any instance $x \in \{0,1\}^n$ for $n = \mathrm{poly}(\mu)$ and any state ρ: if the execution $(P^*(x, \rho), V(x))$ returns 1 with probability $\varepsilon > \kappa(\mu)$, we have, letting $\sigma = \frac{(I-|\perp\rangle\langle\perp|)K^{|P^*(x,\rho)\rangle}(x)(I-|\perp\rangle\langle\perp|)}{\mathrm{Tr}[(I-|\perp\rangle\langle\perp|)K^{|P^*(x,\rho)\rangle}(x)]}$,*

$$\mathrm{Tr}[(I-|\perp\rangle\langle\perp|)K^{|P^*(x,\rho)\rangle}(x)] > p\left(\varepsilon - \kappa(\mu), \frac{1}{\mu}\right) - negl(\mu), \text{ and } (x, \sigma) \in R_{Q,q(|x|,\varepsilon)}.$$

The intuition behind the last equation is that we want the probability that the extractor K does not output '$\perp$' to be at least p, and we want the state conditioned on not outputting $\perp$ to be a good enough witness.

Remark 3. Note that quality of the witness returned by the extractor K in Definition 8 may be lower than the quality of the witness used by the prover to produce the proof. We suspect that this loss is inherent. Consider the following simple example. Suppose the prover is given a witness ρ that has quality $0 < c < 1$ with respect to some QMA verification procedure. The prover uses ρ in a protocol that executes one of two tests, each with probability 1/2: (i) an "energy test" that is designed to check ρ, and (ii) a "structure test" that is designed to check some property of the prover's strategy.

Now consider two provers, P_1 and P_2, each of which succeeds in this protocol with probability $c' = (1+c)/2$. P_1 is given a witness of quality c and plays optimally in the structure test. P_2 is given a witness of quality 1 and purposefully succeeds in the structure test with probability c only. Then because of the existence of P_1, it would be unreasonable to expect that the extractor can extract witnesses of quality $> c$ from provers that succeed with probability $\leq c'$. This means that running P_2 on a witness returned by the extractor will succeed with probability $c < c'$ only.

We also define *arguments* of quantum knowledge (with a setup). The main difference is that the proof system is replaced by an argument system with setup. Moreover, the extractor is allowed to create the setup as they wish (they can "impersonate" the setup procedure S).

Definition 9 (Quantum Argument of (Classical) Knowledge). *Let $R \subseteq \mathcal{X} \times \mathcal{Y}$ be a relation. An argument system with setup $\Pi = (S, P, V)$ for R is a Quantum Argument of Knowledge with setup for R with knowledge error κ if there exists a polynomial $p > 0$ and a quantum polynomial-time machine K such that for any quantum polynomial-time interactive machine P^*, any $\mu \in \mathbb{N}$, any polynomial $l > 0$, any instance $x \in \{0,1\}^n$ for $n = \mathrm{poly}(\mu)$ and any state ρ: if the execution $(S, P^*(x,\rho), V(x))$ returns 1 with probability $\varepsilon > \kappa(\mu)$, we have*

$$\Pr\left(\left(x, K^{|P^*(x,\rho)\rangle}(x)\right) \in R\right) \geq p\left(\varepsilon - \kappa(\mu), \frac{1}{\mu}\right) - negl(\mu) .$$

Definition 10 (Argument of Quantum Knowledge). *Let (Q, α, β) be a QMA relation. An argument system with setup $\Pi = (S, P, V)$ is an Argument of Quantum Knowledge with setup for (Q, α, β) with knowledge error κ and quality $q > \beta$ if there exists a polynomial $p > 0$ and a quantum polynomial-time interactive machine K such that for any quantum polynomial-time interactive machine P^*, any $\mu \in \mathbb{N}$, any polynomial $l > 0$, any instance $x \in \{0,1\}^n$ for $n = \mathrm{poly}(\mu)$ and any state ρ: if the execution $(S, P^*(x,\rho), V(x))$ returns 1 with probability $\varepsilon > \kappa(\mu)$, we have, letting $\sigma = \frac{(I-|\bot\rangle\langle\bot|)K^{|P^*(x,\rho)\rangle}(x)(I-|\bot\rangle\langle\bot|)}{\mathrm{Tr}[(I-|\bot\rangle\langle\bot|)K^{|P^*(x,\rho)\rangle}(x)]}$,*

$$\mathrm{Tr}[(I - |\bot\rangle\langle\bot|)K^{|P^*(x,\rho)\rangle}(x)] > p\left(\varepsilon - \kappa(\mu), \frac{1}{\mu}\right) - negl(\mu), \text{ and } (x, \sigma) \in R_{Q,q(|x|,\varepsilon)}.$$

As for the several possible specializations to the definition of Argument of Quantum Knowledge with setup based on the properties of the underlying argument system (NIZK, CRS setup, preprocessing etc.), we naturally apply the terminology introduced in Sect. 2.3, and in Sect. 1.3 of the Supplementary Material.

Reducing the Knowledge Error Sequentially. One of the most natural properties of Proofs of Knowledge that one investigates in the classical setting is reducing the knowledge error by sequential repetition. Classically, it is well-known that the knowledge error drops exponentially fast in the number of sequential repetitions [BG92]. Just like in the classical case, sequential repetition of a proof of quantum knowledge reduces the knowledge error exponentially fast. This is an immediate consequence of the proof of a lemma from Unruh [Unr12] for the case of quantum Proofs of (classical) Knowledge. We refer the reader to the Supplementary Material for a formal statement.

3 The Protocol

3.1 Notation and Predicates

For a circuit Q_n, we denote by $H(Q_n)$ the local Clifford Hamiltonian obtained by performing the circuit-to-Clifford-Hamiltonian reduction from [BJSW16, Section 2]. In the rest of this section, Q_n will always be taken from a family $Q = \{Q_n\}_{n\in\mathbb{N}}$, where Q specifies a QMA relation (Q, α, β), and we will let the r-th term of the Clifford Hamiltonian $H(Q_n)$ be $C_r^* \left|0^k\right\rangle \left\langle 0^k\right| C_r$. So,

$$H(Q_n) = \sum_{r=1}^{m} C_r^* \left|0^k\right\rangle \left\langle 0^k\right| C_r, \tag{2}$$

where each C_r is a k-local Clifford unitary. (Following [BJSW16], we use the short-hand $|0^k\rangle\langle 0^k|$ to denote a projector which is $|0\rangle \langle 0|$ on at most k qubits and identity everywhere else. As shown in [BJSW16], we can take $k = 5$ without loss of generality.)

We denote by $\mathcal{H}_{\mathsf{clock}} \otimes \mathcal{H}_{\mathsf{instance}} \otimes \mathcal{H}_{\mathsf{witness}}$ the Hilbert space that $H(Q_n)$ acts on. For notational convenience, we assume in the rest of this section that $\mathcal{H}_{\mathsf{instance}}$ is n qubits, that is, $\mathcal{H}_{\mathsf{instance}} = \mathbb{C}^{2^n}$.

For clarity and notational convenience, we define predicates R_r and Q below, which we will refer to in our description of our protocol.

Remark 4. Predicates Q and R_r are defined with respect to a fixed problem instance x and a fixed Clifford Hamiltonian H, where

$$H = \sum_{r=1}^{m} C_r^* \left|0^k\right\rangle \left\langle 0^k\right| C_r$$

for some m that is polynomial in n.

Definition 11 (Definition of R_r). *As in Sect. 2.2, we write $\mathcal{D}_N$ to represent the set of all valid (classical) N-bit codewords of a particular error-correcting code. We will generally refer to this error-correcting code as 'the concatenated Steane code'. (This code is the same concatenated Steane code which is outlined in [BJSW16, Appendix A.6].) We may write $\mathcal{D}_N = \mathcal{D}_N^0 \cup \mathcal{D}_N^1$, where $\mathcal{D}_N^0$ is the set of all codewords that encode 0, and $\mathcal{D}_N^1$ is defined analogously.*

We assume that r takes values in $[m+1]$, where m is the number of terms in the Clifford Hamiltonian H. Our R_r is defined differently when $r \in [m]$ and when $r = m + 1$.

1. *If $r \in [m]$: Let $u_{i_1}, \ldots, u_{i_k} \in \{0,1\}^{2N}$, $\pi \in S_{2N}$, and $t_{i_1}, \ldots, t_{i_k} \in \{0, +, +_y\}^N$. For each $i \in \{i_1, \ldots, i_k\}$, define strings p_i, q_i in $\{0,1\}^N$ such that $\pi(p_i \| q_i) = u_i$ (alternatively: $\pi^{-1}(u_i) = p_i \| q_i$). We define a predicate $\tilde{R}_r(t, \pi, u)$ that takes value 1 if and only if the following two conditions are met:*

(a) $p_i \in \mathcal{D}_N$ *for every* $i \in \{i_1, \ldots, i_k\}$, *and* $p_i \in \mathcal{D}_N^1$ *for at least one index* $i \in \{i_1, \ldots, i_k\}$. *(*$\mathcal{D}_N = \mathcal{D}_N^0 \cup \mathcal{D}_N^1$ *is the set of all valid classical* N*-bit codewords of the concatenated Steane code).*

(b) $\langle q_{i_1} \cdots q_{i_k} | C_r^{\otimes n} | t_{i_1} \cdots t_{i_k} \rangle \neq 0$.

Here $|t_{i_1} \cdots t_{i_k}\rangle$ *is the state of* kN *qubits obtained by tensoring* $|0\rangle, |+\rangle$ *and* $|+_y\rangle$ *in the natural way. Then, we define* $R_r(t, \pi, u) = \tilde{R}_r(t, \pi, u)$.

2. If $r = m + 1$, *then we set* $R_r = R_{m+1}$, *where* R_{m+1} *is defined below (Definition 12).*

Definition 12 (Definition of R_{m+1}). *Let* $u = u_{\mathsf{clock}_1}, u_{\mathsf{instance}_1}, \ldots, u_{\mathsf{instance}_n}$ *be a string in* $\{0,1\}^{2N(n+1)}$.

Remark 5. Each u_{label}, *for* $\mathsf{label} \in \{\mathsf{clock}_1, \mathsf{instance}_1, \ldots, \mathsf{instance}_n\}$, *is a* $2N$*-bit string, and intuitively represents the result of measuring the* logical *qubit with an index specified by* label*. (For notational convenience in the exposition below, we replace the iterator* label *by the iterator* i*.) For example,* u_{clock_1} *is the string that results from measuring the first* logical *qubit of the clock register. The logical clock register consists of many logical qubits, and each logical qubit is encoded in* $2N$ *physical qubits as a result of applying the authentication code described in Fig. 1.*

For $\pi \in S_{2N}$, *and for each* $i \in \{\mathsf{clock}_1, \mathsf{instance}_1, \ldots, \mathsf{instance}_n\}$, *define strings* p_i, q_i *in* $\{0,1\}^N$ *such that* $\pi(p_i \| q_i) = u_i$ *(alternatively:* $\pi^{-1}(u_i) = p_i \| q_i$*). The predicate* $R_{m+1}(t, \pi, u)$ *takes the value 1 if and only if the following two conditions (1. and 2.) are met:*

1. Either

$p_{\mathsf{clock}_1} \in \mathcal{D}_N^1$ *(this corresponds to the first qubit of the clock register, expressed in unary, being in state* 1*, i.e. the clock register is* not *at time* 0*),*

or

For every $i \in \{\mathsf{instance}_1, \ldots, \mathsf{instance}_n\}$, $p_i \in \mathcal{D}_N^{x_i}$.

2. $\langle q_{\mathsf{clock}_1} \, q_{\mathsf{instance}_1} \cdots q_{\mathsf{instance}_n} | t_{\mathsf{clock}_1} \, t_{\mathsf{instance}_1} \cdots t_{\mathsf{instance}_n} \rangle \neq 0$.

We now define our predicate Q in terms of the R_r defined in Definition 11.

Definition 13 (Definition of Q). *Let* $d = (x_1, \ldots, x_{2Np(n)}, y_1, \ldots, y_{2Np(n)})$ *be a string in* $\{0,1\}^{4Np(n)}$, *for some polynomial* $p(n)$ *of* n*. Define*

$$\begin{aligned}\mathbb{P}_{m+1} =& |0\rangle\langle 0|_{\mathsf{clock}_1} \otimes \big(I - |x\rangle\langle x|\big)_{\mathsf{instance}} \otimes I_{\mathsf{witness}} \\ &+ (I - |0\rangle\langle 0|)_{\mathsf{clock}_1} \otimes I_{\mathsf{instance}} \otimes I_{\mathsf{witness}}\end{aligned}$$

where $|x\rangle\langle x|$ *is a shorthand for the projector onto the standard-basis bitstring* $\langle x \rangle$, *and*

$$C_{m+1} = I_{\mathsf{clock}} \otimes I_{\mathsf{instance}} \otimes I_{\mathsf{witness}}.$$

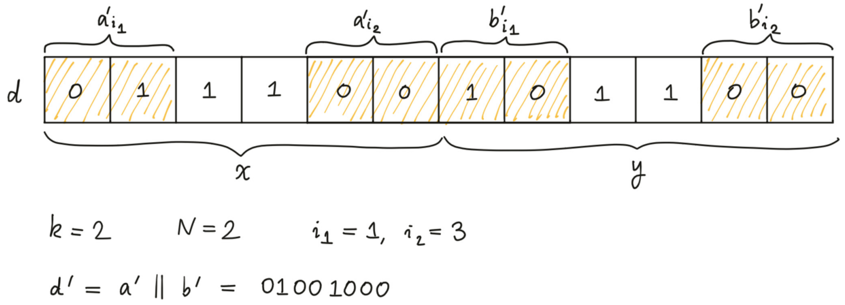
For $r \in [m+1]$, *define*

$$\mathbb{P}_r = \begin{cases} C_r^* \, |0^k\rangle\langle 0^k| \, C_r & r \in [m] \\ \mathbb{P}_{m+1} & r = m+1 \end{cases}$$

Let $i_1, .., i_k$ be the indices of the qubits on which $\mathbb{P}_r$ acts non-trivially. Let

$$\begin{aligned} d' &= (a', b') \\ &= (a'_{i_1}, \dots, a'_{i_k}, b'_{i_1}, \dots, b'_{i_k}) \\ &= (x_{2Ni_1+1}, \dots, x_{2Ni_1+2N}, \dots, x_{2Ni_k+1}, \dots, x_{2Ni_k+2N}, \\ &\quad y_{2Ni_1+1}, \dots, y_{2Ni_1+2N}, \dots, y_{2Ni_k+1}, \dots, y_{2Ni_k+2N}) \end{aligned}$$

be a string in $\{0,1\}^{4Nk}$. (The example below, wherein $k = 2, N = 2, i_1 = 1, i_2 = 3$, and $d' = (a', b') = 01001000$, may clarify the notation.)



Let $e_{i_1}, \dots, e_{i_k}$ be the unique strings such that

$$\begin{aligned} C_r^{\otimes 2N}(X^{(a \oplus a')_{i_1}} Z^{(b \oplus b')_{i_1}} \otimes \cdots \otimes X^{(a \oplus a')_{i_k}} Z^{(b \oplus b')_{i_k}}) \\ = \alpha(X^{e_{i_1}} Z^{f_{i_1}} \otimes \cdots \otimes X^{e_{i_k}} Z^{f_{i_k}}) C_r^{\otimes 2N} \end{aligned} \tag{3}$$

for some $\alpha \in \{1, i, -1, -i\}$ and some $f_{i_1}, \dots, f_{i_k} \in \{0,1\}^{2N}$. (It is possible to efficiently compute $e = e_{i_1}, \dots, e_{i_k}$ and $f = f_{i_1}, \dots, f_{i_k}$ given a, b and C_r.)

Predicate Q is defined as follows:

$$Q(t, \pi, a, b, r, z, d) = R_r(t, \pi, z \oplus e_{i_1} \cdots e_{i_k}).$$

3.2 The Protocol

Parties. The argument system involves

1. A (QPT) *verifier V*,
2. A (QPT) *prover P*, and
3. A (classical PPT) *setup machine S*.

The verifier sends a single quantum message to the prover in the preprocessing phase of the protocol, and the prover sends the verifier a single classical message in the instance-dependent phase of the protocol. S sends an identical classical message to both the prover and the verifier during the preprocessing phase.

Inputs. (Unless otherwise stated, all inputs are common to all three parties.)

1. Preprocessing stage:
   (a) An instance size parameter n and a security parameter μ.

(b) A QMA relation (Q, α, β).
(c) The Clifford Hamiltonian $H(Q_n)$ (see Eq. (2)).
(d) Other parameters:
  i. $c(n)$, an upper bound on the number of qubits in a witness state;
  ii. $p(n)$, an upper bound on the number of qubits in a history state corresponding to an execution of Q_n on a witness state of length $c(n)$ and an instance of size n;
  iii. $m = \text{poly}(n)$, the number of terms in the Clifford Hamiltonian (Eq. (2));
  iv. $N = \text{poly}(n)$, the number of physical qubits per logical qubit in the Steane code introduced in Sect. 2.2.
(e) A perfectly binding, quantum computationally concealing (classical) commitment scheme $\mathsf{Com} = (\mathsf{Com.gen}, \mathsf{Com.commit}, \mathsf{Com.reveal}, \mathsf{Com.verify}, \mathsf{Com.recover})$, of the form described in Sect. 1.2 of the Supplementary Material.
(f) A NIZK argument system with setup for NP, such as the one described in Sect. 1.3 of the Supplementary Material. We denote this argument system by a triple $(\mathsf{NIZK}.S, \mathsf{NIZK}.P, \mathsf{NIZK}.V)$.
(g) A levelled fully homomorphic encryption scheme with circuit privacy, $\mathsf{FHE} = (\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval}, \mathsf{FHE.Refresh})$.

2. Instance-dependent stage:
(a) The instance x.
(b) Auxiliary (private) input to the prover: a witness ρ for the claim that $x \in R_{Q,\alpha}$.

Protocol
Refer to Fig. 2 for a diagram charting the essential structure of the protocol below.

3.3 Preprocessing Step

(S) S samples $\gamma \leftarrow \mathsf{NIZK}.S(1^{q(n)}, 1^{\mu})$, for a sufficiently large polynomial $q > 0$, in addition to $(pk_P, sk_P), (pk_V, sk_V) \leftarrow \mathsf{Com.gen}(1^{\mu})$ (the two pairs are sampled independently). S then outputs (γ, pk_P, pk_V).

(i) V samples $r' \xleftarrow{\$} [m+n]$. Sets $r = r'$ if $r \leq m$, and $r = m+1$ otherwise. Computes $\sigma \leftarrow \mathsf{Com.commit}(pk_V; r)$. Let s_V be the randomness used.

(ii) V creates $2Np(n)$ EPR pairs ($2N$ for each qubit of the history state). The second qubit of each EPR pair will be sent to P in step (iv). We refer to the remaining qubits (i.e. the first qubit in each EPR pair) as "V's qubits".

(iii) V interprets her $2Np(n)$ qubits as the encoding (according to the authentication code of Fig. 1) of a history state for $H(Q_n)$.
- If $r \in [m]$: V applies C_r transversally to the subset of her $2Np(n)$ qubits which encode the k logical qubits on which C_r acts. Measures those qubits in the computational basis, obtaining an outcome string z.
- If $r = m+1$: V measures the subset of the $2Np(n)$ qubits corresponding to the first qubit of $\mathcal{H}_{\mathsf{clock}}$ and all the qubits of $\mathcal{H}_{\mathsf{instance}}$ in the computational basis, obtaining an outcome string z.

(iv) V samples $(pk_E, sk_E) \leftarrow \mathsf{FHE.Gen}(1^n)$. V sends to P:
- $\alpha \leftarrow \mathsf{FHE.Enc}(pk_E, (r, s_V, z))$.
- pk_E and σ.

3.4 Instance-Dependent Step

- **Prover's message:**
  1. P computes the history state corresponding to an evaluation of the circuit Q_n on the input $|x\rangle \otimes |\psi\rangle$. This is the state $|\Psi\rangle = \sum_{t=0}^{T} |t\rangle_{\mathsf{clock}} \otimes \Pi_{j=1}^{t} U_j \big(|x\rangle \otimes |\psi\rangle \otimes |0\rangle^{\otimes n}\big)$ for some unitary U_j, which can be computed efficiently. P computes $|\tilde{\Psi}\rangle \leftarrow \mathsf{Auth.Enc}(|\Psi\rangle)$ according to the authentication scheme of Fig. 1. Let the sampled authentication keys be:
     (a) $a = a_1, .., a_{p(n)}$, $b = b_1, .., b_{p(n)}$ for $a_1, .., a_{p(n)}, b_1, .., b_{p(n)} \in \{0,1\}^{2N}$,
     (b) $\pi \in S_{2N}$,
     (c) $t = t_1, .., t_{p(n)}$ where $t_1, .., t_{p(n)} \in \{0, +, +_y\}^N$.
     P samples commitment randomness s_P, and computes
     $\sigma_{\mathsf{keys}} \leftarrow \mathsf{Com.commit}(pk, (t, \pi, a, b), s_P)$.
  2. P teleports the state ρ to V using his halves of the $2Np(n)$ shared EPR pairs received in step (iv) of the preprocessing step. Let $d = (x_1, \ldots, x_{2Np(n)}, y_1, \ldots, y_{2Np(n)}) \in \{0,1\}^{4Np(n)}$ be the Bell basis measurement outcomes obtained during the teleportation.
  3. P computes $\beta \leftarrow \mathsf{FHE.Enc}\big(pk_E, (d, \sigma, \sigma_{\mathsf{keys}}, (t, \pi, a, b), s_P)\big)$, where σ is the commitment received in step (iv) of the preprocessing step. P homomorphically evaluates the following circuit C using β and the ciphertext α that it received from the verifier. (Recall that α is an encryption of (r, s_V, z).)

     > C takes as input $d, \sigma, r, s_V, z, \sigma_{\mathsf{keys}}, t, \pi, a, b, s_P$. It checks that (r, s_V) is a valid opening for σ, and that $Q(t, \pi, a, b, r, z, d) = 1$, where Q is defined in Definition 13. If its checks pass, using γ it computes an NIZK argument for the existence of an opening to σ_{keys} such that the opened value (t, π, a, b) satisfies $Q(t, \pi, a, b, r, z, d) = 1$. If its checks do not pass, it outputs "$\perp$".

  4. Let $\tilde{\pi}$ be the encrypted proof that P obtains in step 4. P computes $\tilde{\pi}' \leftarrow \mathsf{FHE.Refresh}(\tilde{\pi})$. Sends d, σ_{keys} and $\tilde{\pi}'$ to V.
- **Verifier's check:** V decrypts $\tilde{\pi}'$, and executes $\mathsf{NIZK.V}$ to check the decrypted proof. It checks that the d received from P is the same d that appears in the statement being proven.

Theorem 1. *Assuming that LWE is intractable for quantum polynomial-time (QPT) adversaries, every language in* QMA *has an adaptively zero-knowledge non-interactive argument system with CRS setup and preprocessing (where completeness is according to Definition 4) with* negl *adaptive soundness. Moreover, the preprocessing phase consists of a single quantum message from the verifier to the prover.*

We refer to the combination of the protocols of Sects. 3.3 and 3.4 as "the protocol".

To show Theorem 1 we start with an arbitrary language $L \in \text{QMA}$. Using standard amplification techniques, for any polynomial t there is a family of polynomial-size verification circuits Q such that L is the language associated with the QMA relation $(Q, 1-2^{-t}, 2^{-t})$ as in Definition 1. We show that the protocol associated to this relation is an NIZK argument with setup for $(Q, 1-2^{-t}, 2^{-t})$. Completeness is easy to verify, as for any $(x, \rho) \in R_{Q,1-2^{-t}}$ the prover described in Sect. 3.4 is accepted with probability negligibly close to 1, given access to ρ. In Sect. 4 we prove soundness inverse polynomially close to 1, and in Sect. 2.3 of the Supplementary Material we show how soundness can be amplified in parallel to any 2^{-p} for polynomial p (provided t is taken large enough compared to p). After parallel amplification, completeness holds only if we allow the prover to receive polynomially many copies of the witness (as in Definition 4). Finally, in Sect. 5 we prove the zero-knowledge property.

4 Soundness

In this section we prove soundness of our protocol from Sect. 3.2. This is captured by the following lemma.

Lemma 2. *Assume that LWE is intractable for quantum polynomial-time (QPT) adversaries. Let* (Q, α, β) *be a QMA relation. Then the non-interactive protocol with setup and preprocessing for* (Q, α, β) *described in Sect. 3.2 has negligible adaptive soundness.*

We give an overview of the proof of Lemma 2 in the next subsection.

4.1 Overview

The structure of the proof is as follows. We show through a sequence of hybrids that it is possible to transform an execution of our protocol on some instance x, into an execution of the protocol from [BJSW16] on a specific local Clifford Hamiltonian derived from x. We show that this transformation can at most negligibly decrease the optimal acceptance probability of the prover. Thus, soundness of our protocol reduces to soundness of the protocol from [BJSW16]. The main steps in our sequence of hybrids are the following:

- Remove the encryption of V's choice of r, randomness s_V and measurement outcomes z sent in step (iv) of the preprocessing step.
- Replace the step where P teleports the encoded witness to V through shared EPR pairs (step 2 in Sect. 3.4) with one where P directly sends the qubits of the encoded witness to V.
- Remove the portion of the CRS corresponding to the NIZK argument, and replace the NIZK argument sent by the prover in step 4 of Sect. 3.4 with a ZK proof.

In Sect. 2.3 of the Supplementary Material, we amplify soundness by repeating the protocol in parallel. One can check that our proof goes through unchanged for the case of adaptive soundness as well. In particular, the key is that the NIZK proof system for NP employed in our protocol is adaptively sound.

5 Zero-Knowledge Property

Lemma 3. *Assume that LWE is intractable for quantum polynomial-time (QPT) adversaries. Let L be a language in QMA, let $x \in \{0,1\}^*$ be a problem instance, and let $V^* = \{V^*_{\mu n}\}$ be an arbitrary QPT verifier for the protocol of Sect. 3. There exists a QPT simulator $S = \{S_{\mu n}\}$ such that, for any μ, n and yes-instance $x \in L$ with $|x| = n$, and for any auxiliary quantum input Z_0 to the verifier, the distribution of V^*'s final output after its interaction with the honest prover P in the protocol is quantum computationally indistinguishable from S's output distribution on auxiliary input Z_0.*

Furthermore, the simulator S only requires knowledge of the instance x after the preprocessing phase has been executed (simulated) with V^*. As such, the zero-knowledge property holds in the adaptive setting.

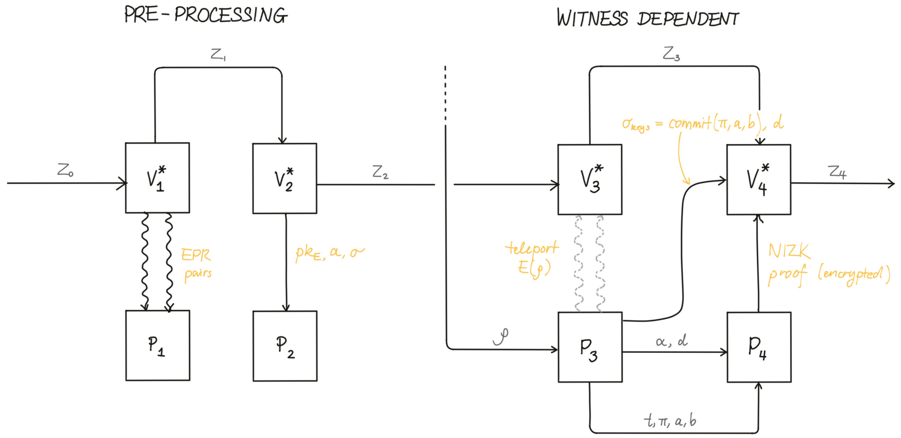


Fig. 2. Diagram representing the original protocol execution between the honest prover P and a cheating verifier V^*. For visual clarity, the prover and the (cheating) verifier have been split into parts $\{P_i\}$ and $\{V_i^*\}$ with $i \in \{1,2,3,4\}$, respectively, where parts 1 and 2 execute the preprocessing phase of the protocol, and parts 3 and 4 execute the instance-dependent phase of the protocol. Communications between verifier and prover are labelled in orange; internal communications on either side are labelled in grey. In the two subsequent diagrams, we will omit the auxiliary input Z_0 that the cheating verifier receives, as well as the internal communications Z_1, Z_2, Z_3 between the different parts of the cheating verifier.

Due to space constraints, we provide the proof of Lemma 3 in Sect. 3 of our supplementary material. In order to show that our protocol is (adaptively) zero-knowledge, we proceed through the following hybrid argument, in which we make a series of replacements, and show at each stage that the verifier's final output after the replacement is made is (computationally or statistically) indistinguishable from its output before. Figure 2 is a diagram that numbers the stages of the prover and the verifier in the original protocol. For convenience, we use the numbering scheme presented in that figure.

1. In the original protocol, P_4 offers an encryption (under a homomorphic encryption scheme FHE) of a non-interactive NP proof π, which has been computed homomorphically, to the last component of the potentially cheating verifier, V_4^*. We replace the encryption of the genuine proof π with the encryption of a *simulated* proof π'. π' is indistinguishable from π because the proof system is zero-knowledge. We use the circuit privacy property of FHE to show that the encryption of π' is also indistinguishable from the encryption of π.
2. Step 1 allows us (details of how are provided in supplementary material) to replace the commitment to encoding keys that P_3 sends to V_4^* with a commitment to a fixed string, which the verifier could generate by itself.
3. After the replacement in step 2 has been made, we are then able to replace the genuine witness ρ which the honest P_3 receives with a *simulated* witness that can be efficiently prepared without knowledge of the real witness. Arguing that the verifier's final output after this replacement is (statistically) indistinguishable from its output before is perhaps the most involved step in the proof, and involves in particular making use of the *extractability* property of the commitment scheme (see Sect. 1.2 of the supplementary material) that the verifier uses to commit to its challenge r in order to argue that the simulator can efficiently recover r and then construct a simulated witness which passes only the challenge determined by r.

6 NIZK Argument of Quantum Knowledge with Preprocessing for QMA

In this section we show that for any QMA relation the NIZK argument system with CRS setup and preprocessing described in Sect. 3 is also a NIZK Argument of Quantum Knowledge with CRS setup and preprocessing (as defined in Sect. 2.4). The intuition for this is simple. From the proof of soundness of the protocol from [BJSW16], to which soundness of our argument system reduces, we are able to infer that any prover which is accepted in our protocol with high probability must be teleporting to the verifier *an encoding* of a low-energy witness state for the given instance of the 5-local Clifford Hamiltonian problem. Then, all that an extractor (given oracle access to such a prover) has to do in order to output a good witness is:

– Simulate an honest verifier so as to receive (by teleportation) such an encoded witness from the prover,

– Find a way to recover the committed encoding keys and use them to decode the received state.

We formalize this sketch in Sect. 4 of the Supplementary Material.

7 Proofs of Quantum Knowledge for QMA

The interactive protocol that we show is a proof of quantum knowledge for languages in QMA is identical to the protocol from [BJSW16], as recalled in Sect. 2.2, except for one modification: at the same time as the prover sends the encoded state $E(\rho)$ and the commitment σ to the verifier (end of step 1 of the protocol), the prover also sends a classical zero-knowledge PoK of an opening to the commitment. More precisely, define a relation R such that $R(\sigma, z) = 1$ if z is a valid opening for the commitment σ. V and P engage in a ZK PoK protocol for the relation R on common input σ, as defined in Definition 6. If the verifier rejects in this protocol, then the verifier outputs "reject" for the whole protocol; otherwise the verifier proceeds to the next phase.

Informally, the extractor K first takes the quantum state ρ^* sent by P^* in the first message. It then executes an extractor K' for an opening to the commitment sent in the first message, that must exist by the quantum proof of knowledge property for the sub-protocol. If K' succeeds in recovering the committed keys, K decodes the state received in the first message using these keys and returns the decoded state. Otherwise, K returns an abort symbol "$\perp$". We formalize this sketch in Sect. 5 of the Supplementary Material.

References

[BDSMP91] Blum, M., De Santis, A., Micali, S., Persiano, G.: Noninteractive zero-knowledge. SIAM J. Comput. **20**(6), 1084–1118 (1991)

[BG90] Bellare, M., Goldwasser, S.: New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 194–211. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_19

[BG92] Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 390–420. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_28

[BG19] Broadbent, A., Grilo, A.B.: Zero-knowledge for QMA from locally simulatable proofs (2019)

[BJSW16] Broadbent, A., Ji, Z., Song, F., Watrous, J.: Zero-knowledge proof systems for QMA. In: 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), pp. 31–40. IEEE (2016)

[BMW03] Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_38

[CLW19] Canetti, R., Lombardi, A., Wichs, D.: Fiat-Shamir: from practice to theory, part II (2019)

[Com14] Electric Coin Company. Zcash Cryptocurrency (2014)

[CP92] Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_7

[DL09] Damgaard, I., Lunemann, C.: Quantum-secure coin-flipping and applications. arXiv e-prints, arXiv:0903.3118, March 2009

[GGPR13] Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_37

[GMR85] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, STOC 1985, pp. 291–304. ACM, New York (1985)

[GO94] Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptol. **7**(1), 1–32 (1994). https://doi.org/10.1007/BF00195207

[Kob02] Kobayashi, H.: Non-interactive quantum statistical and perfect zero-knowledge. arXiv e-prints, quant-ph/0207158, July 2002

[Lab17] O(1) Labs. Coda Cryptocurrency (2017)

[Mah18] Mahadev, U.: Classical verification of quantum computations. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS), pp. 259–267. IEEE (2018)

[NY90] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing, STOC 1990, pp. 427–437. ACM, New York (1990)

[PHGR13] Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: nearly practical verifiable computation. In: 2013 IEEE Symposium on Security and Privacy, pp. 238–252. IEEE (2013)

[PS19] Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. IACR Cryptology ePrint Archive 2019:158 (2019)

[Sah99] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, FOCS 1999, p. 543. IEEE Computer Society, Washington, DC (1999)

[Sho95] Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. arXiv e-prints, quant-ph/9508027, August 1995

[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

[Unr15] Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 755–784. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_25

[VZ19] Vidick, T., Zhang, T.: Classical zero-knowledge arguments for quantum computations. arXiv e-prints, arXiv:1902.05217, February 2019

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