6.845 Quantum Complexity Theory - Final Project **Spring 2022**

Quantum Zero-Knowledge Proofs

Florian Juengermann, Taro Spirig

Abstract

We review recent results in the field of quantum zero-knowledge proofs. The main result we discuss is the zero-knowledge proof system for problems in the complexity class QMA developed by Broadbent et al. in [Bro+16]. We then discuss follow-up works towards Non-Interactive Zero-Knowledge (NIZK) proof systems for QMA focusing on the results shown in [CVZ20]. Finally, we present an adaptation of [Bro+16] into a classical zero-knowledge argument for QMA in which the verifier can be purely classical as shown in [VZ20].

1 Classical Zero-Knowledge

An interactive protocol is said to be zero-knowledge if on an accepting instance, the verifier cannot learn anything throughout the protocol apart from the validity of the proved statement. This property needs to hold whatever a cheating verifier tries to do. To show that a given protocol is zero-knowledge, one can show that there exists an efficient simulator simulating the protocol in the accepting instance without interacting with the prover. In other words, the efficient simulator produces a transcript which is indistinguishable from the transcript produced by a possibly cheating verifier interacting with the prover. It thus implies that the verifier cannot learn anything more than it could learn by itself without interacting with the powerful prover.

1.1 A Classical Example

One of the most well-known example of a classical zero-knowledge proof is the Goldreich-Micali-Widgerson graph isomorphism proof system [GMW91] (it was shown in the same seminal paper that all problems in NP have a zero-knowledge proof). The setup is as follows: the verifier and the prover posses two simple, n-vertex undirected graphs (G_0, G_1) and the verifier wants to know if the two graphs are isomorphic to each other. The steps of the interactive protocol for this problem are the following:

1. The prover chooses a permutation $\pi \in S_n$ at random and sends the graph $H = \pi(G_0)$ to the verifier.
2. The verifier then challenges the prover by choosing $a \in \{0, 1\}$ uniformly at random and sending it to the prover.
3. The prover sends back $\tau = \pi\sigma^a$ where $\sigma \in S_n$ is the permutation such that $\sigma(G_1) = G_0$ which is known to the prover in the accepting instance.
4. The verifier accepts if $\tau(G_a) = H$.

The intuition for this protocol is as follows. In the first step, the prover commits to one of the graphs G_0 or G_1 (in this implementation it always chooses G_0). In case the prover knowns the isomorphism σ, it can pass any possible challenges. If not, however, it will fail with probability $\frac{1}{2}$.
The efficient simulator constructed to prove the zero-knoweldge property of this protocol is as follows: the simulator guesses the cheating verifier challenge uniformly at random, i.e. it guesses $a \in \{0, 1\}$, and computes the graph $H = \tau(G_a)$ for a random permutation τ. If the cheating verifier issues the challenge a when sent the graph H, then the simulator can output τ which is the correct answer that the prover would also output. In this case, the simulator has succeeded to simulate the interactive protocol with an arbitrary verifier without interacting with the prover, as desired. On the other hand, if the cheating verifier does not issue a, we can rewind the simulator and try again. One can then show that the simulator will guess a correctly with probability $\frac{1}{2}$. It is thus clear that the simulator is indeed an efficient simulator of the interactive protocol in the accepting instance, which concludes the zero-knowledge proof.
The idea of rewinding in this example is one of the main tools for classical zero-knowledge proofs. In the quantum setting, however, the rewinding procedure fails. Indeed, if a cheating verifier first holds a useful

quantum state, then a simulator which fails to guess the verifier's challenge correctly on the first try might not be able to rewind to the beginning of the protocol as the useful information of the verifier's quantum state might be lost if for example it measured it before sending the challenge. Additionally, a quantum verifier has stronger algorithmic capabilities such as factoring efficiently using Shor's algorithm. The latter issue is less problematic as one can come up with problems difficult (or rather thought to be difficult) even for quantum computers. The former issue can also be solved using for example the so-called quantum rewinding lemma [Wat09]. We do not present nor show this lemma but refer the reader to [VW16] for more details, as in the following QMA zero-knowledge proof this lemma will not be used. Indeed, we will be able to go around the rewinding issue by enabling the efficient simulator to have access to the challenge of the verifier.

1.2 Quantum zero-knowledge

Before moving on to zero-knowledge proof systems for QMA, let us define quantum zero-knowledge proof systems formally. In a quantum zero-knowledge proof, one considers cheating verifiers W which efficiently implement a collection of channels $\{\Phi_x : x \in A\}$, for $A \subseteq \Sigma^*$ a set of binary strings, by interacting with the prover. The goal is then to show that in the YES case, every such channel Φ_x can be efficiently approximated by a simulator Q without interacting with the prover. In other words, the circuit $Q(x)$ implements an $\epsilon(|x|)$-approximation of Φ_x with respect to the diamond norm for each $x \in A$ and some negligible function $\epsilon : \mathbb{N} \to [0, 1]$. As in the classical case, this means that the cheating verifier could have implemented the channels Φ_x without interacting with the powerful prover, showing that it doesn't gain any knowledge by interacting with it.

Definition 1.1. Let $A \subseteq \Sigma^*$ be a set of binary strings and let $P(x)$ be a prover in an interactive game for each $x \in A$. It is said that P is a quantum statistical zero-knowledge prover on the set A if, for every collection of channels $\{\Phi_x : x \in A\}$ that is efficiently implementable through an interaction with P, one has that the collection of channels is also efficiently ϵ-approximable for some choice of a negligible function $\epsilon : \mathbb{N} \to [0, 1]$.

The complexity class of quantum zero knowledge systems $\mathrm{QSZK}_{a,b}(m)$ contains all the promise problem $A = (A_{yes}, A_{no})$ for which there exists a quantum interactive proof system in $\mathrm{QIP}_{a,b}(m)$ whose prover is a quantum statistical zero-knowledge prover on the set A_{yes}.

As an alternative to statistical zero-knowledge proof systems, we can define the notion of computational zero-knowledge proof systems, where the simulator approximates the collection of channels $\{\Phi_x : x \in A\}$ in a computational sense instead of an information theoretic one, i.e. two channels are indistinguishable if no polynomial-size quantum circuit can efficiently distinguish them instead of considering the diamond norm to distinguish them.

2 Zero-knowledge proof system for QMA

As [GMW91] showed that all problems in NP admit a zero-knowledge proof system that is even secure against quantum cheating verifiers [Wat09], it is then natural to consider zero-knowledge proof systems for the class QMA. It was shown in [Bro+16] that all problems in the class QMA admit a zero-knowledge proof system. In the result it is assumed that an unconditionally binding and quantum computationally concealing commitment scheme exists. Instead of using the quantum rewinding lemma, [Bro+16] show that a variant of the local Hamiltonian problem, the local Clifford-Hamiltonian problem, is QMA complete and allows to construct a zero-knowledge proof system. Their protocol uses three key ideas:

- The witness is encrypted with a quantum one-time pad that works in harmony with Clifford measurements. Specifically, the verifier can measure a local Clifford operation on the *encrypted* witness.
- The witness state is encoded with quantum error correcting codes and scrambled together with additional trap qubits. In this way, the cheating verifier is forced to measure the correct qubits, otherwise the prover will reject and not provide further information.

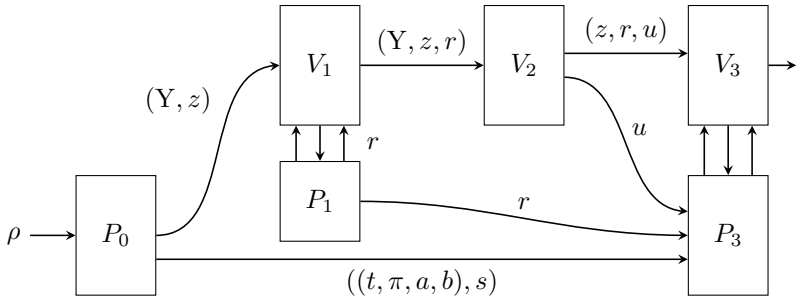


Figure 1: The [Bro+16] Protocol.

- In case the verifier behaves as expected, the prover convinces the verifier of the validity of its measurement through a classical zero-knowledge NP protocol.

.

2.1 Local Clifford-Hamiltonian Problem

In the classical local Hamiltonian problem, the input consists of m k-local Hamiltonian operators acting on n qubits together with real numbers a and b with $a < b$. The task is to decide whether the minimum eigenvalue λ_{min} of the sum of the Hamiltonians is $\leq a$ (YES case) or $\geq b$ (NO case). For a promise gap $b - a$ that is at least inverse polynomial, the problem is QMA-complete for any $k \geq 2$ [KR03; KKR06]. When restricting to an exponentially small a, [Bra11; GN16] showed QMA-completeness for $k = 4$ and $k = 3$ respectively.
For the zero-knowledge proof system of QMA, [Bro+16] introduce an additional restriction. Every k-local term H_i has to be a rank 1 projector of the form

$$H_i = C_i^\dagger \left|0^k\right\rangle\!\left\langle 0^k\right| C_i, \tag{1}$$

where $C_i \in \mathrm{span}\{H, S, CNOT\}$ is an element of the Clifford group. It is not obvious that this local Clifford-Hamiltonian problem is still QMA complete as the Clifford group does not form a set of universal quantum gates. However, [Bro+16] show completeness for $k = 5$ and exponentially small a with an inverse polynomial gap $b - a$. The proof follows the same construction as in the original 5-local Hamiltonian construction by Kitaev [Kit+02] showing that all terms in

$$H_{transcript} = H_{init} + H_{out} + H_{clock} + H_{prop} \tag{2}$$

can be implemented with Clifford-Hamiltonian projectors.

2.2 Description of the Proof System

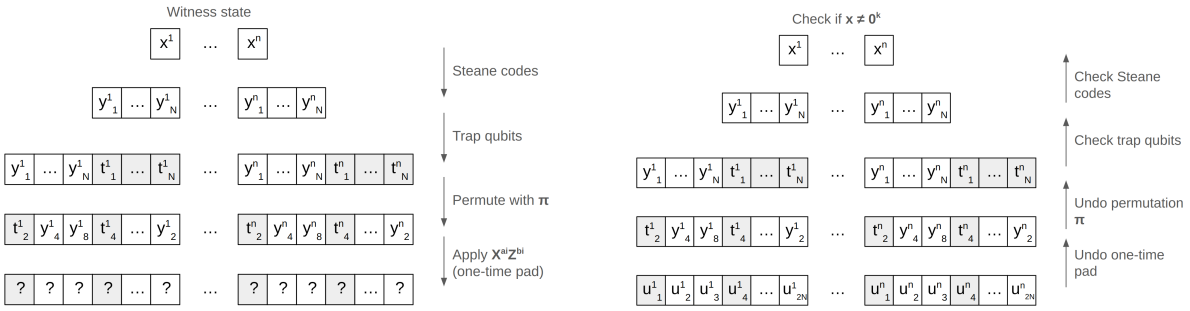
Given a general QMA problem, the completeness from the previous section allows us to phrase it as a local Clifford-Hamiltonian problem with m k-local Clifford operations $C_1, \ldots, C_m$ acting on n qubits. In the YES case, the prover wants to convince the verifier that they possess a low energy witness state ρ, without giving any information on ρ away. We describe the proof system of [Bro+16] in three steps and refer to Figure 1 for a visualization.

2.2.1 Witness Encoding (Prover)

Let us assume the prover holds a witness ρ in the registers $\mathrm{X}_1, \ldots, \mathrm{X}_n$. We define N as a polynomially bounded even power of 7. The prover encodes the witness in the following way (see Figure 2a for a visualization):

- Every qubit X_i is encoded by N qubits $\mathrm{Y}_1^i, \ldots, \mathrm{Y}_N^i$ by concatenating Steane codes. The Steane code is a quantum error correcting CSS code that is based on the classical (7,4,3) Hamming code. This results in the state

$$(\mathrm{Y}_1^1, \ldots, \mathrm{Y}_N^1), \ldots, (\mathrm{Y}_1^n, \ldots, \mathrm{Y}_N^n) \tag{3}$$



(a) encoding (b) check

Figure 2: Overview of the witness encoding (a) and the check (b).

- For each of the n N-tuples, we append N trap qubits. Each trap qubit is a uniformly random selection out of $\{|0\rangle, |+\rangle, |\circlearrowleft\rangle\}$. The prover stores the information about the values of the trap qubits in $t = t_1 \cdots t_n$ with $t_i \in \{0, +, \circlearrowleft\}^N$. The state becomes

$$(\mathrm{Y}_1^1, \ldots, \mathrm{Y}_{2N}^1), \ldots, (\mathrm{Y}_1^n, \ldots, \mathrm{Y}_{2N}^n) \tag{4}$$

- A single permutation $\pi \in S_{2N}$ is chosen uniformly at random and π is applied to each of the n $2N$-tuples individually.

- The quantum one-time pad is applied. For that, we choose $a = a_1 \cdots a_n$, $b = b_1 \cdots b_n$ with $a_i, b_i \in \{0,1\}^{2N}$ uniformly at random and apply $X^{a_i} Z^{b_i}$ to $(\mathrm{Y}_1^i, \ldots, \mathrm{Y}_{2N}^i)$ for all $i \in [n]$.

The prover then sends Y together with the commitment

$$z = \mathrm{commit}((\pi, a, b), s), \tag{5}$$

which encodes the chosen permutation π and the strings a, b defining which Pauli matrices were applied in the one-time pad. The string s is a random string chosen by the prover that is needed for the commitment. Intuitively, the verifier will not learn anything from this state as it cannot undo the one-time pad because we assume that the commitment z conceals the input a and b. Furthermore, the prover will later be able to check if the verifier reports honest measurements by looking at the trap qubits.

2.2.2 Random Challenge (Verifier)

The verifier wants to check whether the witness state encoded in Y has low energy on the sum of the Clifford-Hamiltonians. Specifically, that means the state needs to have low energy for every term H_i. The verifier thus chooses an index $r \in [m]$ uniformly at random to confirm that the witness is a low-energy state of the term $H_r = C_r^\dagger \left|0^k\right\rangle\!\!\left\langle 0^k\right| C_r$. Let $i_1, \ldots, i_k$ be the k qubits that C_r acts on non-trivially. Then, the verifier applies C_r on the $2N$ k-tuples

$$(\mathrm{Y}_1^{i_1}, \ldots, \mathrm{Y}_1^{i_k}), \ldots, (\mathrm{Y}_{2N}^{i_1}, \ldots, \mathrm{Y}_{2N}^{i_k}), \tag{6}$$

and measures the result in the standard basis. The verifier sends the outcomes $u_{i_1}, \ldots, u_{i_k} \in \{0,1\}^{2N}$ to the prover and expects the prover to demonstrate that these outcomes correspond to a low energy state.
While this is the intended behavior, we later want to show the zero-knowledge property for any cheating verifier. Therefore, we add an additional step to ensure the index r is chosen uniformly at random. The idea is to use a *coin flipping protocol* [Blu83] that ensures randomness:

1. The prover commits a random $y_i \in \{0,1\}$.
2. The verifier chooses $z_i \in \{0,1\}$.
3. The prover reveals y_i and both parties agree that the random coin is $r_i = y_i \oplus z_i$.

By repeating this for $\lceil \log m \rceil$ times, the parties can agree on a truly random string r. [DL09] showed that this coin flipping protocol is quantum-secure for a quantum-secure commitment scheme.

2.2.3 Check and Response (Prover)

Once the prover receives the challenge u, it first checks whether it is the outcome of an honest measurement of the verifier (see Figure 2b). If so, the prover tries to convince the verifier that the outcomes correspond to a low energy state. The intuition for the first step is that we want to ensure that a cheating verifier measures the agreed-upon Clifford projection H_r on the qubits it received. For that, we evaluate a predicate R_r which depends on t, π, u *after* undoing the one-time pad. First, we split up each u_i into the parts $y_i, z_i \in \{0,1\}^N$ which correspond to the original qubits and to the trap qubits:

$$\pi(y_i z_i) = u_i \tag{7}$$

We define R_r to take value 1 if and only if all the three following conditions are met:

(1) Every y_i corresponds to a valid code word from the Steane code used to encode the witness.

(2) At least one y_i corresponds to an encoding of a $|1\rangle$ state.

(3) The trap qubit measurements do not contradict t: $\langle z_{i_1} \cdots z_{i_k}| \, C_r^{\otimes N} \, |t_{i_1} \cdots t_{i_k}\rangle \neq 0$

Conditions (1) and (3) check that u corresponds to a valid measurement. Condition (2) shows that our witness has a low energy on H_r as measuring 0^k would indicate a high energy state.
To remove the one-time pad, we observe that there is a unique solution $c_1, \ldots, c_n, d_1, \ldots, d_n \in \{0,1\}^{2N}$ that corresponds to removing the one-time pad *after* applying the Clifford operations:

$$C_r^{\otimes 2N}(X^{a_1} Z^{b_1} \otimes \cdots \otimes X^{a_n} Z^{b_n}) = \alpha(X^{c_1} Z^{d_1} \otimes \cdots \otimes X^{c_n} Z^{d_n}) C_r^{\otimes 2N} \tag{8}$$

with $\alpha \in \{1, i, -1, -i\}$. For Clifford operations, it is easy to compute c and d from a and b. So the prover evaluates the predicate

$$Q_r(t, \pi, u, a, b) = R_r(t, \pi, u \oplus c_{i_1} \cdots c_{i_k}) \tag{9}$$

If the predicate evaluates to zero, the prover aborts the protocol, otherwise, it uses a classical NP zero-knowledge protocol to convince the verifier that there exist a string s, and variables t, π, a, b with $\text{commit}((\pi, a, b), s) = z$ and $Q_r(t, \pi, u, a, b) = 1$. This is a valid NP statement as the predicate Q_r and the commitment can be efficiently checked with classical computation for a given witness (s, t, π, a, b) and the known values z and u.

2.3 Completeness and soundness

For completeness, we need to show that in the YES case, an honest verifier will accept with high probability. Conditions (1) and (3) from subsubsection 2.2.3 pass with certainity as u corresponds to an honest measurement. Condition (2) fails if the verifier measures high energy on the term C_r. As the minimum eigenvalue λ_{min} is exponentially small in this case, the condition passes with high probability.
For the soundness of the proof system, we show that a dishonest prover cannot fool an honest verifier in the NO case. Through the NP protocol, the verifier makes sure the commitment z corresponds to a valid permutation π and one-time pad (a, b). However, when interacting with a dishonest prover, we cannot assume that the qubits (before adding the trap qubits)

$$(Y_1^1, \ldots, Y_N^1), \ldots, (Y_1^n, \ldots, Y_N^n) \tag{10}$$

correspond to valid Steane codes. Instead, we assume them to be in an arbitrary state ξ. [Bro+16] show that there is a mapping function Ξ[1] that maps a $n \times N$ qubit state to an n qubit state $\rho = \Xi(\xi)$ while preserving its trace. We know that in the NO case, for any n qubit state ρ, there is at least one Hamiltonian term H_r which will yield a high energy when acting on ρ. The trace-conservation condition implies that ξ has high energy and the check (2) from subsubsection 2.2.3 fails with non-negligible probability. The coin-flipping protocol yields r with non-negligible probability, so the rejection probability is bounded away from 0.

[1] We use a slightly simplified notation $\Xi = \Xi_N^{\otimes n}$ compared to [Bro+16]

2.4 Zero-knowledge Property

We now want to prove that the interactive protocol described above is zero-knowledge. We therefore show that there exists a simulator which can efficiently approximate any channel that a cheating verifier might implement during the interactive protocol even if the simulator does not have access to the witness ρ. The protocol for a cheating verifier is similar to the honest verifier protocol described above apart from the fact that a cheating verifier might use an additional quantum register Z_0 as input and output the result of the channel it implements in a register Z_3. We denote with Z_0 and Z_1 the input and output registers of the first step performed by the verifier, i.e. the random challenge, which we denote by V_1'. Similarly we denote by Z_1 and Z_2 the input and output registers of the second step performed by the verifier, i.e. the check, which we denote by V_2'. Now the idea of the proof is to first simplify the protocol by simulating certain parts of the protocol and by eliminating the commitment that the prover sends in the first step of the protocol. It is important to note that these simplifications only make sense as we are considering the accepting instance where the verifier tries to cheats to learn more than it should. The simplifications we make are obviously not giving an equivalent protocol in all instances, e.g. in the soundness case shown above. Finally we directly show that there exists an efficient simulator to the simplified protocol.

2.4.1 Simulating Parts of the Protocol

We show that we can simulate the coin flipping protocol and the classical zero-knowledge proof in the last part of the protocol.
As shown in [DL09], we can replace the coin-flipping protocol by a simulator S_1 that takes as input the quantum register Z_0, the encoded witness Y, the commitment z together with a true random string r. S_1 outputs the quantum register Z_1 and forwards the random string r to the prover.
As we assume that the last step of the protocol is a classical zero-knowledge proof system, there must exist an equivalent simulator in the accepting instance. In other words, we replace the last interaction between the verifier and the prover by a simulator S_3 which takes as input the quantum register Z_2 and the results of the predicate Q. It then outputs the register Z_3. We assume here that the simulator S_3 also behaves as the verifier would when the prover aborts the protocol, i.e. when the output of the predicate Q is zero. We also note that as Q does not take s as an input we can get rid of it as soon as the commitment has been sent to the simulator S_1.

2.4.2 Eliminating the Commitment

As we assume the commitment scheme to be quantum computationally concealing and that the commitment is never revealed in the process, the cheating verifier has no way of knowing whether (π, a, b) is actually not chosen at random. We can thus assume here that the commitment is a fixed set of (π_0, a_0, b_0) with random s. We can thus consider the commitment to (π_0, a_0, b_0), the simulator S_1 and the verifier's circuit V_2' together as one single efficiently implementable cheating verifier circuit V'. It takes as input the register Z_0, the encoded witness Y and the random string r. It ouputs the register Z_1 and the measurement u.

2.4.3 The Simulation

As the last step of the protocol only consists in the simulation S_3, we only need to show that the channel implemented by V' is efficiently simulable. We construct the following simulator: The simulator first queries the random generator for the string r (it does not matter if the string is produced before or during the protocol) and then produces the state ρ_r which passes the energy measurement test (given the rth term of the Hamiltonian, one can easily construct a state which will have low energy on the k-qubits it acts on). Then it goes through the encoding procedure and simulates anything the cheating verifier would do. The difference between the true channel and the simulated one is that the simulator uses the easy to construct state ρ_r instead of the true witness ρ. In the next step we outline the proof that thanks to the encoding, those channels become computationally indistinguishable. To show this, property for a random string r, we prove it for every fixed value of r. Specifically, we show the slightly more general statement that the channels implemented by V' taking as input the state ξ_0 or ξ_1, which have respective probability p_0 and p_1 to pass

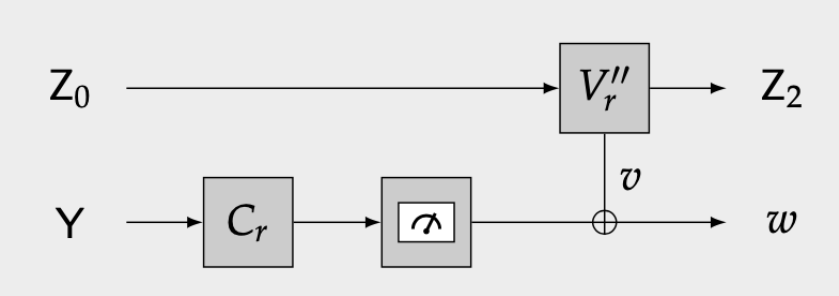


Figure 3: The twirling simplification leads to a simpler circuit where the verifier only interacts with the prover by XOR-ing a string v to the prover measurement of the state Y.

the test of the protocol, will be statistically indistinguishable if $|p_0 - p_1|$ is negligible. As ρ and ρ_r both only fail with negligible probability, this will give us the result.

2.4.4 Twirling

We first show that we can simplify the verifier's work V'. Indeed, the one-time pad labelled by (a, b) that the prover acts with on Y is chosen at random. As we have seen in the protocol, the prover needs to remove the one-time pad to compute the predicate Q after it receives the measurement outcome u of V'. This is done by finding the unique (c, d) given by equation (8) and applying $u \oplus c$. We can thus rewrite the application of the one-time pad labeled by (a, b) by the equivalent operator labelled by (c, d) as in equation (8). We then take the average over randomly chosen c and d instead of choosing one tuple (c, d). The prover first acts on Y with C_r then with an average over Z^d. Averaging over Z^d chosen uniformly at random puts all of the states in Y in the standard basis. The rest of the operations consist of the average over c of $V'(I \otimes C_r^\dagger)(I \otimes X^c)(\mathrm{Z}_0 \otimes \mathrm{Y}')$, where Z_0 denotes the state in the register Z_0 and Y' is the encoded witness in the standard basis, followed by XOR-ing the measurement outcome of V' with c. As the application of the random superposition of the Paulis X^c correspond to some bit flips of states in Y' and V' is forced to output a classical string, we can rewrite this whole process in an alternative way by acting on the state Y with C_r and then measure it in the standard basis. We then define a quantum instrument V'' which takes as input the register Z_0 only and outputs a string v which is XOR-ed to the output of the measurement. We represent this simplification in Figure 3.

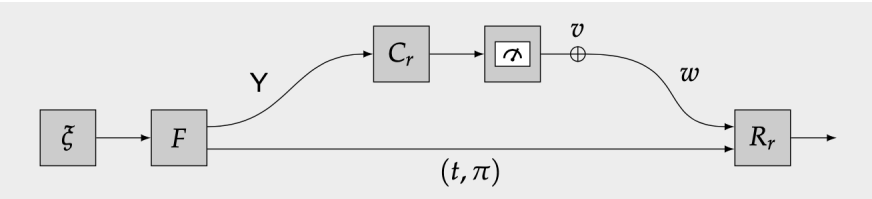


Figure 4: The simplified circuit we consider to show that the simulator can simulate any XOR attack by the cheating verifier.

2.4.5 XOR attacks

As Z_2 does not depend on the encoded witness Y, it is efficiently simulable and we only consider the output w. By the simplification made in the previous step, we can now consider the circuit shown in Figure 4, i.e. without loss of generality any attack from a cheating verifier will be an XOR attack. Note that w is not accessible to the verifier but is the input to the prover's predicate R_r. So, what is left to show is that if ξ_0 and ξ_1 are close with respect to the accepting probability, then any fixed choice of v cannot make the outcome of R_r distinguishable.

Let K be the minimum number of ones of a non-zero codeword in the concatenated Steane code. We look at three cases depending on the number of 1-bits in v.

Case 1: $|v|_1 = 0$ we can check the two conditions (1) and (3) from section 2.2.3 easily. The trap qubits z_i and the data qubits y_i are not modified so all the checks will pass. For condition (2), the probability of having at least one y_i that encodes a 1 is exactly the probability of ξ passing the challenge. So the statement holds in this case.

Case 2: $0 < |v|_1 < K$ The number of changed bits is not enough to change the encoding of a code word. So for any ξ, we do not change the number of y_i which encode a 1, thus the probability of passing condition (2) is unchanged. A bit flip might change the probabilities of passing conditions (1) and (3), however, not in a way that depends on the input ξ. This means that if ξ_0 and ξ_1 have an indistinguishable acceptance probability, their acceptance probability may change but in the same way, thus staying indistinguishable.

Case 3: $|v|_1 \geq K$ In this case it is possible to change a code word such that we might be able to separate ξ_0 and ξ_1. But as the permutation π is random, with high probability, v will also modify some trap qubits. While we do not fail with certainty if we modify a trap bit, [Bro+16] show that the probability of passing condition (3) decreases exponentially with K. So in this case, the probability of outputting 1 is negligible for all inputs ξ and we cannot distinguish ξ_0 and ξ_1.

3 Non-interactive zero-knowledge arguments and proofs for QMA

A natural challenge regarding an interactive proof protocol is to reduce the number of communication rounds between the prover and the verifier. This leads to the definition of a non-interactive protocol. A non-interactive zero-knowledge (NIZK) proof system is a zero-knowledge protocol where the prover only sends one message to the verifier for an instance x of the problem. To make it feasible, an instance-independent preprocessing phase where the verifier and the prover communicate with each other is allowed. NIZK proof systems have seen a wide number of applications in classical cryptography where the corresponding interactive protocol would not be implementable, notably in blockchain applications. As we have seen in the zero-knowledge protocol presented above, one of the main ingredients of the protocol is the random challenge that the verifier forces the prover to answer. Without any interaction such a challenge is hard to implement. To circumvent that issue, one usually considers a common shared string (CRS) model where both the verifier and the prover have access to some string sampled from a given distribution (not necessarily uniformly random). It has been recently shown in [PS19] that all languages in NP have NIZK proofs in the CRS model which are safe against quantum attacks assuming learning with errors (LWE)[Reg09] is quantum computationally hard. It is then natural to ask the question whether all languages in QMA also have NIZK proofs. Several advances have been made to answer this question in [CVZ20; BG20; Ala+20; BS19] with varying settings and assumptions. We present a high-level description of [CVZ20] as it is a natural follow-up to [Bro+16]. The result in that work is a NIZK argument based on the CRS model augmented by a message of quantum preprocessing. An argument is a relaxed version of a proof system where the prover is efficient, i.e. in the soundness case an efficient dishonest prover cannot cause the verifier to accept and in the completeness case, the honest prover is efficient given a witness. In [BG20], Broadbent and Grilo describe a statistical NIZK proof system with preprocessing for QMA which is based on a simpler protocol than the one in [Bro+16]. The main disadvantage of this powerful result is that their protocol is not based on the CRS model which is desirable especially in application. Instead, in their model, the trusted randomness is asymmetric for the verifier and the prover.

3.1 NIZK argument in the CRS model

The idea in [CVZ20] is to consider the protocol in [Bro+16] and use quantum teleportation and cryptographic techniques to transform it into a non-interactive protocol. As mentioned above, [PS19] showed that there exist a quantum secure NIZK proof for all NP problems. We can thus use this protocol to transform the NP protocol of the last step in the protocol of Broadbent et al. into a NIZK proof system. The main challenge is

that to make the protocol non-interactive, we somehow need to get rid of the communication of the check and response phase (subsubsection 2.2.3) in the protocol. However, the prover cannot guess the outcome of the verifier's measurement nor produce all possible measurements. This is where quantum teleportation comes into play. With that, the verifier can do its measurement even before the instance of the problem has been set as will become clearer below. One might ask how an instance-independent measurement might even look like? The Clifford Hamiltonian (1) was constructed from a k-local Hamiltonian which was in turn constructed from an instance specific history state of the QMA quantum circuit Q. One can then instead consider the Hamiltonian $H(Q)$ constructed from the quantum circuit Q without specifying the instance. The prover is then asked to send a witness state which corresponds to a history state of the circuit Q on an instance x. The verifier then can simply measure one term from the Hamiltonian $H(Q)$ without knowing the instance. The prover's instance dependent witness will still have the same completeness and soundness properties since the Hamiltonian $H(Q)$ will have a low energy only on uncorrupted accepting history states. The malicious prover could however ignore the instance x to provide a low energy state. With some probability, the verifier will therefore have to make a measurement that ensures that the prover is indeed considering the circuit Q on the correct instance x to construct the witness. In summary, the two steps of the protocol are the following:

1. *Preprocessing*: The verifier start with a large number of EPR pairs and samples unformly at random a challenge r. The verifier then measures one of the following:

   - $H(Q)$ on half of the EPR pairs,
   - the special measurement to verify that the prover evaluates Q on the correct instance x later on.

   The verifier then creates encoding keys for a homomorphic encryption and sends the following to the prover:

   - the public key,
   - the untouched half of the EPR pairs,
   - a commitment to r, i.e. commit(r,s) for some random string s,
   - a homomorphic encryption of: r, s and the verifier measurement outcome u.

2. The prover receives an instance x as well as a witness $\langle\Psi\rangle$ and computes a history state. The prover then samples encoding keys and sends an encoding of the history state to the verifier using quantum teleportation applied to the half of the EPR pairs it has received. The prover send to the verifier:

   - the teleportation measurement outcome d,
   - a commitment σ to the encoding keys,
   - a homomorphic encryption of a homomorphically computed NIZK proof in the CRS model of the existence of an opening of σ such that the opened keys together with d, z, r are consistent with a low-energy measurement.

Completeness for this protocol follows from the fact that in the case where both the prover and the verifier are honest, their actions commute and hence the verifier's measurement in the preprocessing phase could have been made after the prover's commitment which is then equivalent to the Broadbent et al. protocol. Soundness comes intuitively from the homomorphic encryptions in the protocol. The two homomorphic encryptions are indeed needed such that the prover can't learn r nor the measurement outcome u of the verifier prior to constructing the witness. If it would have access to these two quantities it could easily construct in the first case a state which would have low energy on the rth term of the Hamiltonian only or in the second case choose the string d such that d, z, r are consistent with a low-energy measurement. The zero-knowledge proof follows from the zero-knowledge proof of Broadbent et al. except for the fact that the commitment to the challenge r was constructed through an interactive coin-flipping protocol, such that an efficient simulator could have access to its value with certainty. In the current setting, the verifier chooses r alone and the simulator would need to guess its value which would lead to another route to analyze the protocol (we would need to rewind the protocol as in the classical setting discussed in the first section). Instead Coladangelo, Vidick, and Zhang force the verifier to choose r and commit to it by taking a public

key determined by the CRS. They then show that in this way a simulator can efficiently recover r using the CRS. The protocol is thus indeed a NIZK argument in the CRS model with a one-message preprocessing step.

4 Classical Zero-Knowledge Arguments for QMA

As shown by Vidick and Zhang [VZ20], the proof system we present from [Bro+16] can be modified to allow for classical verifiers. By doing so, however, the verifier is no longer sound against any quantum prover, but only sound against quantum polynomial-time provers in the NO-case.
An interesting application for the near future is that it would allow actors without access to quantum hardware to verify that a quantum computer has a secret state that fulfills certain properties.
To see how to transform Broadbent et al.'s proof system described in section 2 into a classical argument system, we observe that almost all communication and computation on the verifier's side is already classical. Only the initial message from the prover to the verifier containing the encrypted witness ρ is quantum. Upon receiving the state, the verifier is supposed to immediately measure it and the rest of the protocol is purely classical.
To eliminate the measurement on the verifier side, [VZ20] uses the measurement protocol introduced by [Mah18]. This protocol allows a classical verifier to receive honest measurements from the prover's quantum state. To apply the results from this protocol, [VZ20] uses the 2-local XZ Hamiltonian problem instead of the 5-local Clifford Hamiltonian problem used in [Bro+16]. To make the verifier sound while not having access to the witness, they split up the coin-flipping protocol into two stages.

4.1 Changes to the Protocol

The protocol works very similarly to the one described in section 2. We briefly describe the key differences in the following sections.

2-local XZ Hamiltonian problem The 2-local XZ Hamiltonian problem is a variant of the k-local Hamiltonian problem in which only the Pauli observables σ_X, σ_Z are used. Specifically, the input contains $H = \sum_{i=0}^{m} d_i H_i$ where d_i is a polynomially bounded rational weighting factor and H_i can be represented as the tensor product of identity matrices with at most two Pauli matrices σ_X or σ_Z. [BL08] showed that this version of the problem is QMA-complete for an inverse-polynomial promise gap.
Otherwise, the first step of the protocol is unchanged. The witness state is encoded with concatenated Steane codes, scrambled with a set of trap qubits[2] and encrypted with a quantum one-time pad. As the verifier is classical, it is not possible to send the encrypted witness. Instead, the prover only sends the commitment $z = \text{commit}((\pi, a, b), s)$.

Splitted Coin Flipping While the prover has committed to his encoding parameters, it has not committed to anything related to the witness ρ. So if the verifier were to send the challenge r now, the dishonest prover could easily construct a low energy state ρ_r. Instead, the verifier keeps its part of the coin flipping protocol r_v secret and only shares a commitment $c = \text{commit}(r_v, s')$. Then, the prover sends its part of the coin flipping protocol r_p to the verifier. The resulting randomness $r = r_p \oplus r_v$ is now fixed but only known to the verifier.

Measurement Protocol [Mah18] We briefly summarize the Mahadev's measurement protocol and refer the reader to [Mah18] for a more detailed description. The key idea of the protocol is the method for the prover to commit to a quantum witness with a classical commitment. For that, the verifier prepares $2nN$ functions η_{κ_i} which are either one-to-one or two-to-one. It then sends the functions encoded by the keys κ_i to the prover. Based on the intractability assumption of the LWE problem [Reg09], a quantum polyonmial time prover cannot tell apart whether κ_i describes a one-to-one or two-to-one function. The verifier keeps a trapdoor for each function that allows it to invert each one of them. Now, the prover applies the functions to a superposition state involving the witness, measures the function output and sends the results back to

[2] Here we choose the trap qubits $t \in \{0, +\}^N$ rather than $t \in \{0, +, \circlearrowleft\}^N$.

the verifier. This step can be seen as a commitment to the witness. The verifier does one of the following two options with probabilty $\frac{1}{2}$ each:

- *Test round.* The verifier requests a standard basis measurement of the witness and checks that it is consistent with the previous commitment.
- *Hadamard round.* The verifier requests a Hadamard basis measurement on the witness. It then uses the trapdoors to invert the commitment together with the reported outcomes from the prover to compute the measurement outcome of the witness state.

Now, the verifier has the measurement outcomes like in the original protocol from section 2 and we can continue as before. The verifier sends its measurements to the prover, this time together with the secret trapdoors and its random string r_v. The prover verifies that everything is consistent and then finally convinces the verifier that the measurement outcomes correspond to a low-energy state with a classical zero-knowledge NP protocol. So this argument system works with completely classical verifiers.

References

[Blu83] Manuel Blum. "Coin flipping by telephone a protocol for solving impossible problems". In: *SIGACT News* 15.1 (1983), pp. 23–27. DOI: 10.1145/1008908.1008911. URL: https://doi.org/10.1145/1008908.1008911.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. "Proofs That Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems". In: *J. ACM* 38.3 (July 1991), pp. 690–728. ISSN: 0004-5411. DOI: 10.1145/116825.116852. URL: https://doi.org/10.1145/116825.116852.

[Kit+02] Alexei Yu Kitaev et al. *Classical and quantum computation*. 47. American Mathematical Soc., 2002.

[KR03] Julia Kempe and Oded Regev. "3-local Hamiltonian is QMA-complete". In: *Quantum Inf. Comput.* 3.3 (2003), pp. 258–264. DOI: 10.26421/QIC3.3-7. URL: https://doi.org/10.26421/QIC3.3-7.

[KKR06] Julia Kempe, Alexei Kitaev, and Oded Regev. "The complexity of the local Hamiltonian problem". In: *Siam journal on computing* 35.5 (2006), pp. 1070–1097.

[BL08] Jacob D Biamonte and Peter J Love. "Realizable Hamiltonians for universal adiabatic quantum computers". In: *Physical Review A* 78.1 (2008), p. 012352.

[DL09] Ivan Damgård and Carolin Lunemann. "Quantum-Secure Coin-Flipping and Applications". In: *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*. Ed. by Mitsuru Matsui. Vol. 5912. Lecture Notes in Computer Science. Springer, 2009, pp. 52–69. DOI: 10.1007/978-3-642-10366-7_4. URL: https://doi.org/10.1007/978-3-642-10366-7%5C_4.

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40.

[Wat09] John Watrous. "Zero-knowledge against quantum attacks". In: *SIAM Journal on Computing* 39.1 (2009), pp. 25–58.

[Bra11] Sergey Bravyi. "Efficient algorithm for a quantum analogue of 2-SAT". In: *Contemporary Mathematics* 536 (2011), pp. 33–48.

[Bro+16] Anne Broadbent et al. "Zero-knowledge proof systems for QMA". In: *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2016, pp. 31–40.

[GN16] David Gosset and Daniel Nagaj. "Quantum 3-SAT is QMA _1-complete". In: *SIAM Journal on Computing* 45.3 (2016), pp. 1080–1128.

[VW16] Thomas Vidick and John Watrous. "Quantum Proofs". In: *Foundations and Trends® in Theoretical Computer Science* 11.1-2 (2016), pp. 1–215. ISSN: 1551-305X. DOI: 10.1561/0400000068. URL: http://dx.doi.org/10.1561/0400000068.

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

[BS19] Nir Bitansky and Omri Shmueli. *Post-quantum Zero Knowledge in Constant Rounds*. Cryptology ePrint Archive, Report 2019/1279. https://ia.cr/2019/1279. 2019.

[PS19] Chris Peikert and Sina Shiehian. "Noninteractive Zero Knowledge for NP from (Plain) Learning with Errors". In: Aug. 2019, pp. 89–114. ISBN: 978-3-030-26947-0. DOI: 10.1007/978-3-030-26948-7_4.

[Ala+20] Gorjan Alagic et al. "Non-interactive Classical Verification of Quantum Computation". In: *Theory of Cryptography*. Springer International Publishing, 2020, pp. 153–180. DOI: 10.1007/978-3-030-64381-2_6. URL: https://doi.org/10.1007%2F978-3-030-64381-2_6.

[BG20] Anne Broadbent and Alex B. Grilo. "QMA-hardness of Consistency of Local Density Matrices with Applications to Quantum Zero-Knowledge". In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. 2020, pp. 196–205. DOI: 10.1109/FOCS46700.2020.00027.

[CVZ20] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. "Non-Interactive Zero-Knowledge Arguments for QMA, with Preprocessing". In: *Advances in Cryptology – CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part III*. Santa Barbara, CA, USA: Springer-Verlag, 2020, pp. 799–828. ISBN: 978-3-030-56876-4. DOI: 10.1007/978-3-030-56877-1_28. URL: https://doi.org/10.1007/978-3-030-56877-1_28.

[VZ20] Thomas Vidick and Tina Zhang. "Classical zero-knowledge arguments for quantum computations". In: *Quantum* 4 (May 2020), p. 266. DOI: 10.22331/q-2020-05-14-266. URL: https://doi.org/10.22331%2Fq-2020-05-14-266.