Lecture notes on the Mahadev verification protocol[1]

Thomas Vidick

November 8, 2020

[1]These lecture notes are based on four lectures given in October and November 2020 at the Institut Henri Poincaré (IHP) in Paris, under the sponsorship of the Fondation Sciences Mathématiques de Paris (FSMP). I thank the FSMP, the IHP, and the students in the course for feedback and corrections. A complete set of notes for the 10-week course is available at http://users.cms.caltech.edu/ vidick/teaching/fsmp/fsmp.pdf.

Contents

Lecture 1

Testing a qubit under computational assumptions

The goal of these four lectures is to describe and analyze an interactive protocol that can be used to classically and efficiently certify that a quantum device implements an entire quantum computation of one's choice.

In the first lecture we introduce the main ideas of the protocol in an elementary setting by describing a "computational test for a qubit," whose soundness rests on a cryptographic assumption. The resulting interactive test can be executed to certify that a device has quantum capabilities. In the second lecture we formally define the problem of *delegating quantum computations* to an untrusted party and introduce a delegation protocol due to Fitzsimons and Morimae [MF16] that involves quantum communication from the prover to the verifier. In the last two lectures we combine the Fitzsimons-Morimae protocol with the computational qubit test from the first lecture and a few additional ideas to obtain the classical delegation protocol due to Mahadev [Mah18].

We assume basic familiarity with classical complexity theory (interactive proofs) and quantum computation. These four lectures are adapted from a 10-week course, and the interested reader may consult the full notes available at http://users.cms.caltech.edu/ vidick/teaching/fsmp/fsmp.pdf for background and a presentation of additional results related to the ones discussed here.

1.1 Introduction

Suppose given the possibility to interact with an arbitrary "device", or *prover*, modeled as a black-box interactive machine. Suppose that the interaction is restricted to the exchange of classical messages, and that the *verifier* executing the interaction is itself modeled as a classical probabilistic polynomial-time (PPT) machine. Can the PPT verifier implement a test that allows it to certify, with high confidence, that the behavior demonstrated by the device is quantum, i.e. it does not have a classical realization?

In general this is not possible: almost by definition any input-output behavior has a classical model. An additional assumption thus needs to be introduced. In these lectures we consider the natural assumption that the device is itself a polynomial-time machine; however, it may implement arbitrary *quantum* polynomial-time (QPT) computations. Our setting is thus that of a PPT verifier interacting with a QPT prover.

A simple interactive "test of quantumness" in this setting consists in asking the device to factor a large integer n; under the assumption that factoring is hard for classical computers this test adequately distinguishes classical from quantum devices. The main limitation of this test that is generally pointed out is that

in order for a device to successfully demonstrate its "quantumness" it needs to have the capability to implement a large, fault-tolerant quantum computation. If one's goal is solely to demonstrate quantumness then one may hope for a much simpler test, that could be implemented on a not-necessarily-universal quantum machine. Indeed, such tests have long been known under a different natural assumption of the prover, that it constitutes of two spatially isolated components.[1]

A second limitation that is relevant for us is that the factoring test does not seem to provide a means to certify anything beyond quantumness of the device. Crucially, it does not give us any handle on *how* the device successfully passed the test. To build towards more interesting tests, such as a test for certifying randomness or even a complete quantum computation, we need to develop means that allow us to exert a greater control over the device's workspace: informally, we need to be able to certify that the device "has some qubits" and operates on them in a certain way.

The goal of this lecture is to introduce a test of quantumness that has this feature: the test allows us to certify, in a precise sense, that any device that succeeds in it "has a qubit"; moreover, we will gain a solid understanding of "where" that qubit "is". A key insight from the lecture is that in order to obtain such a test we will need to assume that a certain problem is hard not only for classical computers (such as factoring) but also for quantum computers. We start with a definition of a qubit.

1.2 What is a qubit?

The "qubit" is generally introduced in quantum computing courses through its *state space*, which is the set of unit vectors in the complex vector space $\mathbb{C}^2$; we denote this space as $\mathrm{S}(\mathbb{C}^2)$. Thus the *state of a qubit* can be represented by a unit vector in $\mathbb{C}^2$; for example,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \text{and} \qquad |+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

are both valid states for a qubit. So we all know how to recognize a valid state for a qubit. But what about *the qubit itself*? Consider an analogous formalization of the notion of a "probabilistic bit", a two-level system that can be in any "superposition" of its two states, $p\,\overline{0} + (1-p)\,\overline{1}$ for any $p \in [0,1]$. What distinguishes the real, "1-dimensional" degree of freedom $p \in [0,1]$ of the probabilistic bit from the complex, "2-dimensional" degree of freedom of the qubit?

To give a precise definition we turn to the "Heisenberg representation" of quantum mechanics, that places *observable* quantities at a forefront. Informally, we will distinguish a quantum degree of freedom from a classical one by requiring that the quantum degree of freedom can be measured (observed) in two *mutually incompatible ways*. Recall that in quantum mechanics an observable is specified by a Hermitian operator O on $\mathcal{H}$. Each eigenvalue of O represents a possible outcome under a measurement of the observable, and the associated eigenvectors denote states under which the observable deterministically yields that outcome. An observable such that $O^2 = \mathrm{Id}$ has at most two eigenvalues, -1 and $+1$. Such an observable is called a *binary* observable; it is the most frequent kind of observable that we will encounter.

1.2.1 First definition of a qubit

The following definition makes precise our informal presentation of a "qubit" as "a system that can be observed in two mutually incompatible ways".

[1] See the full notes at http://users.cms.caltech.edu/ vidick/teaching/fsmp/fsmp.pdf for an in-depth discussion of this alternate model.

Definition 1.1 (Qubit, Take 1)**.** A *qubit* is a triple $(\mathcal{H}, X, Z)$ consisting of a separable Hilbert space $\mathcal{H}$ and a pair of Hermitian operators X, Z acting on a $\mathcal{H}$ such that $X^2 = Z^2 = \mathrm{Id}$ and $\{X, Z\} = XZ + ZX = 0$.

Let's see why Definition 1.1 captures the intuitive notion of "mutually incompatible" observables. Let the 'computational basis' be an eigenbasis of Z, and the 'Hadamard basis' an eigenbasis of X. Then we claim that the anticommutation relation $XZ + ZX = 0$ ensures that any vector in the former makes a 45° angle with any vector in the latter. To see this, let $|\psi\rangle$ be an eigenvector of X with associated eigenvalue $\varepsilon \in \{\pm 1\}$. Then $\langle\psi|XZ + ZX|\psi\rangle = 0$ immediately implies $2\varepsilon\langle\psi|Z|\psi\rangle = 0$. Given that Z only has -1 and $+1$ as eigenvalues, this relation implies that the projections of $|\psi\rangle$ on the two eigenspaces of Z have equal length; in other words, $|\psi\rangle$ lies exactly between the $+1$ and -1 eigenspaces of Z. (Yet another way of saying this is that all principal angles between an eigenspace of X and one of Z are $\frac{\pi}{4}$.) In this sense any X and Z satisfying the conditions of the definition are "maximally incompatible": any definite state for the one is entirely undetermined (i.e. yields uniformly random outcomes when measured) under the other.

There is a problem with this definition: by allowing the underlying Hilbert space $\mathcal{H}$ to be arbitrary we seem to have all but lost the usual requirement that a qubit is a system whose state space is "two-level" and thus identifiable with the projective space $\mathrm{S}(\mathbb{C}^2)$. Luckily, the following lemma allows us to make the connection with this requirement.

Lemma 1.2. *Let $(\mathcal{H}, X, Z)$ be a qubit. Then there is a Hilbert space $\mathcal{H}'$ and an isomorphism $\mathcal{H} \simeq \mathbb{C}^2 \otimes \mathcal{H}'$ such that under the same isomorphism, $X \simeq \sigma_X \otimes \mathrm{Id}$ and $Z \simeq \sigma_Z \otimes \mathrm{Id}$.[2] Here, σ_X and σ_Z are the usual Pauli observables on $\mathbb{C}^2$: in matrix form,*

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \textit{and} \qquad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Note that a consequence of the lemma is that qubits, as defined in Definition 1.1, only exist in spaces of even (or infinite) dimension! In particular, qubits don't exist in dimension 1; indeed, in dimension 1 all operators commute. This is satisfactory: intuitively, a situation in which all possible observables commute ought to be considered "classical" (for instance, because there is a complete set of simultaneous eigenvectors for all observables).

It will be essential for our later goals that Definition 1.1 does not *a priori* require $\mathcal{H}$ to be a two-dimensional space. Indeed, how would one test such a claim? One does not "see" the dimension of the state space; while it is possible to probe parts of it it can never be excluded that the state space is larger than what is accessible to the experimentalist's setup. In this sense Definition 1.1 has a nice "operational" flavor to it: it refers to *observables* of the system and their properties. Although much work is needed before we are able to make any of these statements formal, we see the definition as a good step towards giving us the ability to "test" that a system "is a qubit". In addition, the definition clearly has meaningful consequences; in particular it implies that qubits do not have a "classical explanation", so that a "test for a qubit" can serve as a "test for quantumness", i.e. a test that distinguishes quantum from classical behavior.

The proof of the lemma makes use of an elementary but fundamental tool in the analysis of many quantum information protocols, the CS (for "Cosine-Sine") decomposition. We recommend the proof as an exercise to the reader.[3]

[2] The reader might wonder what happened to σ_Y... Don't we need it to define our qubit? Here we are taking the "operator algebraists"' perspective, which is that if the system supports X and Z observables then it also supports $Y = iXZ$. Because Y is determined by X and Z, we do not include it in the definition.

[3] For a solution, see the full notes http://users.cms.caltech.edu/ vidick/teaching/fsmp/fsmp.pdf.

1.2.2 A second definition

Unfortunately Definition 1.1 is impossible to "test". In any actual interaction with the device the only information available to the verifier is the result of the device's measurements *when performed on its quantum state*. Requiring that the observables themselves anti-commute is too stringent: one may only hope to certify how the observables act *on the state*, not in general.

To make this point clear, consider the fist step in the analysis of any interactive protocol: how do we model the actions of an arbitrary prover? At each stage the prover receives a question $x \in \mathcal{X}$ and is expected to provide an answer $a \in \mathcal{A}$, where $\mathcal{X}$ and $\mathcal{A}$ are finite sets that are specified by the protocol. Under the assumption that the prover's actions can be modeled using quantum mechanics, which we will always make, there must exist a Hilbert space $\mathcal{H}$ associated with the prover and a state $\rho \in \mathrm{D}(\mathcal{H})$ that the prover possesses at the start of the protocol. (We use $\mathrm{D}(\mathcal{H})$ to denote the set of all density matrices, i.e. positive semidefinite matrices of trace 1, on $\mathcal{H}$.) When the prover receives its question x it measures some observable $O_x = \sum_a \lambda_a \Pi_a^x$, where λ_a are arbitrary and Π_a^x projections that sum to identity. According to the Born rule, it obtains an answer distributed as $\Pr(a|x) = \mathrm{Tr}(\Pi_a^x \rho)$. Finally the quantum state ρ of the prover gets updated as a function of the outcome obtained.[4] When the interaction is executed the only observable data that is accessible to the experimentalist is, at best, the probabilities $\Pr(a|x)$.[5] If O is an observable, $|\psi\rangle$ a state on which it acts, and U an arbitrary unitary,

$$\langle \psi | O | \psi \rangle = \langle U\psi | (UOU^\dagger) | U\psi \rangle \ .$$

Thus two models of the prover, using state $|\psi\rangle$ and observable O or using state $U|\psi\rangle$ and observable $UOU^\dagger$, lead exactly to the same observed data. Our earlier definition of a qubit, by ignoring the role played by the state and imposing constraints on the operators themselves, violates this. This leads us to update our first definition as follows.

Definition 1.3 (Qubit, Take 2)**.** A *qubit* is a triple $(|\psi\rangle, X, Z)$ such that $|\psi\rangle \in \mathrm{S}(\mathcal{H})$, where $\mathcal{H}$ is a separable Hilbert space left implicit in the notation, and X and Z are Hermitian operators on $\mathcal{H}$ such that

$$\{X, Z\}|\psi\rangle = 0 \ . \tag{1.1}$$

Note that the definition still makes the requirement that $X^2 = Z^2 = \mathrm{Id}$ as operators. This is because this requirement follows from the laws of quantum mechanics themselves; informally, it just means that each of X and Z has a spectral decomposition with two associated eigenprojections, i.e. they represent valid binary observables.

At this point there are two important questions we should be asking: (i) Is this definition meanginful? With the anti-commutator weakened as in (1.1), does the definition still capture our intuitive notion of a qubit? (ii) We weakened the definition in an arbitrary-looking way by inserting a dependence on the state vector $|\psi\rangle$. Can we justify this, i.e. are we now able to develop protocols that test the definition? The following lemma provides an answer to the first question.

[4] This formalization is fully general; in particular it can be used to model classical deterministic strategies by setting $\Pi_a^x = 1_{f(x)=a}$ where f would be the function used by the prover to determine its answers. Similarly, randomized strategies can be represented by making use of a totally mixed state $\rho = \sum_r p_r |r\rangle\langle r|$, for some arbitrary distribution $\{p_r\}$, to capture the randomness.

[5] We write "at best" because the experimentalist does not get to see probabilities. Under the i.i.d. assumption it can sometimes estimate them to within an additive error. However, in the case where $\mathcal{A}$ is a large alphabet it may be that all probabilities are exponentially small. This will be the case in some of the experiments that we describe.

Lemma 1.4. *Let $(|\psi\rangle, X, Z)$ be a qubit on $\mathcal{H}$. Then there exists a Hilbert space $\mathcal{H}'$ and an isometry $V : \mathcal{H} \to \mathbb{C}^2 \otimes \mathcal{H}'$ such that*

$$VX|\psi\rangle = (\sigma_X \otimes \mathrm{Id})V|\psi\rangle \qquad \textit{and} \qquad VZ|\psi\rangle = (\sigma_Z \otimes \mathrm{Id})V|\psi\rangle \ . \tag{1.2}$$

The following diagram illustrates the situation guaranteed by the lemma:

$$\begin{array}{ccc} \mathcal{H} & \xrightarrow{V} & \mathbb{C}^2 \otimes \mathcal{H}' \\ {\scriptstyle X,\,Z}\downarrow & & \downarrow{\scriptstyle \sigma_X\otimes\mathrm{Id},\,\sigma_Z\otimes\mathrm{Id}} \\ \mathcal{H} & \xrightarrow{V} & \mathbb{C}^2 \otimes \mathcal{H}' \end{array} \tag{1.3}$$

Note that the lemma no longer says that X is *equal* to $\sigma_X \otimes \mathrm{Id}$ (under the isomprhism π), but only that *it has the same action on the state*, up to the isometry V. In particular, it is now possible for $\mathcal{H}$ to have odd dimension. This is necessary: for example, we can set

$$|\psi\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \ , \quad X = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \ , \quad Z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and still satisfy Definition 1.3. Here, the third dimension has been added to the operators but since none of $|\psi\rangle$, $X|\psi\rangle$ or $Z|\psi\rangle$ has support on it it is "inaccessible" to any experiment that involves only this state and operators. However, it is good to verify that the definition is non-trivial, and in particular requires $\dim(\mathcal{H}) \geq 2$. Indeed, suppose that $X|\psi\rangle$ and $Z|\psi\rangle$ are colinear. Then by (1.2) it follows that $(\sigma_X \otimes \mathrm{Id})V|\psi\rangle$ and $(\sigma_Z \otimes \mathrm{Id})V|\psi\rangle$ are colinear. As we saw in the previous lecture, due to $\{\sigma_X, \sigma_Z\} = 0$ this is impossible.

The proof of the lemma again follows from Jordan's lemma. In the third lecture we will see an explicit definition for the isometry V that provides an alternate proof. For the remainder of the lecture we focus on the second question: can Definition 1.3 be tested in an interactive experiment?

Remark 1.5. Definition 1.3 requires the anti-commutator $\{X, Z\}$ to be exactly zero when evaluated on the state $|\psi\rangle$. In general with any finite test we may only hope to characterize an "approximate qubit", which is defined as a triple $(|\psi\rangle, Z, X)$ such that $\|\{X, Z\}|\psi\rangle\| \leq \varepsilon$ for some $\varepsilon \geq 0$ that measures the "quality" of the qubit. For convenience in these notes we often make a simplifying assumption of "perfect success" that allows us to achieve $\varepsilon = 0$; unless otherwise noted the statements and proofs that we give extend readily to the general case.

1.3 Simon's algorithm

Having defined the object that we aim to certify, we now remind ourselves of the tools at our disposition by reviewing the prototypical example of a task for which the manipulation of quantum information provides a computational advantage.

1.3.1 The algorithm

The input to an instance of Simon's problem is a function $f : \{0,1\}^n \to \{0,1\}^n$ that has the property that f is 2-to-1 (every value in the range has exactly two preimages) and moreover there is a string $s \in \{0,1\}^n$ such that for every $x, y \in \{0,1\}^n$, $f(x) = f(y)$ if and only if $y = x$ or $y = x + s$, where addition is performed

coordinate-wise and modulo 2. The goal is to recover the string s. It is not hard to see that in the worst case any classical algorithm requires at least $\Omega(2^{n/2})$ evaluations of f to determine s. This is because on the one hand for any deterministic algorithm that makes a smaller number of evaluations there is a function f such that all values returned by f are distinct, so no information about s is gained; similarly one can show that for any randomized algorithm if f is chosen at random then it is unlikely that the algorithm will gain any information about s in $\ll 2^{n/2}$ evaluations. On the other hand, by making roughly $\Omega(2^{n/2})$ evaluations at random points then by the birthday paradox one will likely obtain $x \neq y$ such that $f(x) = f(y)$, which immediately reveals $s = x + y$.

Simon showed that there is a quantum algorithm that can solve this problem using only $O(n)$ evaluations, provided that the function f can be evaluated "in superposition". The algorithm first evaluates f on a uniform superposition of inputs, as follows:

$$\begin{aligned} |0^n\rangle|0^n\rangle &\mapsto \frac{1}{\sqrt{2^n}} \sum_x |x\rangle|0^n\rangle \\ &\mapsto \frac{1}{\sqrt{2^n}} \sum_x |x\rangle|f(x)\rangle \ . \end{aligned}$$

It then measures the last register in the computational basis, yielding some $y = f(x_0) = f(x_1)$ where x_0 and $x_1 = x_0 + s$ are the two preimages of y under f. The re-normalized post-measurement state is

$$\frac{1}{\sqrt{2}}\big(|x_0\rangle + |x_1\rangle\big)|y\rangle \ . \tag{1.4}$$

Measuring the first register in the Hadamard basis yields a uniformly random $d \in \{0,1\}^n$ such that $d \cdot s = 0$. Repeating the entire procedure $O(n)$ times yields $(n-1)$ linearly independent such d's, which suffices to recover s with high probability.

1.3.2 Instantiating the black box

The main limitation of Simon's problem is that it only provides a *black-box* separation: the quantum advantage holds under the assumption that the classical or quantum algorithms are allowed to evaluate the function f, but they are not given an explicit description of it. Showing that the separation still holds for an explicit choice of the function f is much harder, because it is difficult to rule out some smart behavior for the classical algorithm that would take advantage of specific code for f; indeed, showing such a separation would be a major breakthrough in quantum algorithms.

This difficulty shouldn't prevent us from toying with the question: Can we identify natural candidates? For example one could take $f(x) = Ax$ for $A \in \mathbb{F}_2^{n\times n}$ a matrix of rank exactly $(n-1)$. In that case the kernel of A is spanned by a single vector $s \in \mathbb{F}_2^n$, and f is exactly 2-to-1: $f(x_0) = f(x_1)$ if and only if $A(x_0 - x_1) = 0$, i.e. $x_0 - x_1$ is either 0 or s. Unfortunately this f is not a good candidate, because there happens to be an efficient classical algorithm that directly solves Simon's problem for it: Gaussian elimination.[6] The example shows that at a minimum we need a function f that is 2-to-1 but such that finding any colliding pair of inputs (x_0, x_1) with $f(x_0) = f(x_1)$ is computationally difficult. In the next section we introduce some background from cryptography that will allow us to make this requirement precise.

[6] In the last lecture we will see that a "noisy" version of f provides a partial workaround.

1.4 Computational assumptions

In this section we briefly review the formalism for making computational assumptions precise and apply it to a specific scenario of interest for the lecture.

1.4.1 PPT and QPT procedures

The first thing that we need to make precise is our computational model. Since the protocols we consider involve interaction between a verifier and prover we focus on modeling such devices as machines that perform a computation. Loosely speaking, each device operates in a number of rounds where at each round the device performs a computation that takes it from a certain internal state as well as an input (a message received from another device) to a new internal state and an output (a message that it returns). We will model each such computation as a circuit. A circuit is a sequence of elementary operations called "gates" that operate either on a classical state (in which case the gates can be things like an AND, an OR, a NOT, etc.) or a quantum state (in which case the gates can be things like a 1-qubit Hadamard, a σ_X or σ_Z, a 2-qubit controlled NOT, etc.).[7] To recap, for us a verifier or a prover is specified by a sequence of classical or quantum circuits. We will always assume that the circuits explicitly specify which spaces they are meant to operate on (e.g. verifier's space, message from verifier to prover, etc.).

Next we discuss what it means for a verifier (or prover) to be "efficient". To make this precise we need to talk about *families* of verifiers. We will imagine that there is an underlying size parameter $n \in \mathbb{N}$ (for example, n could be the size of a 3SAT formula that the verifier aims to check, or the number of qubits that she aims to certify) and that the verifier (or prover) is specified by a classical Turing machine M that on input 1^n returns an explicit classical description of a sequence of circuits that can be used to implement the verifier (or prover) for problems of size n. We will say that the verifier (or prover) is *probabilistic polynomial time* (PPT) (resp. *quantum polynomial time* (QPT)) if this Turing machine runs in time polynomial in its input (i.e. polynomial in n; this is why we always assume that n is passed in unary to M) and returns a family of classical (resp. quantum) circuits. Note that the assumption that the Turing machine is polynomial time immediately implies that the circuits it returns act on polynomially many bits (resp. qubits) and have a polynomial number of classical (resp. quantum) gates.

In a cryptographic context we will generally allow M to take a second input 1^λ for $\lambda \in \mathbb{N}$ called the *security parameter*. While the input size n is governed by the size of the problem, the security parameter can be chosen at will; the larger it is the more "secure" the protocol is supposed to be (for example, the smaller the probability that the verifier makes an incorrect decision or the higher the quality of the certified qubits).

1.4.2 Claw-free functions

Similarly to circuits in the previous section, when we talk about computational *difficulty* of a certain problem we always need to refer to *families* of objects. This is because e.g. for any given function f there is nothing "hard" about the task of recovering specific information about f: if f is fixed everything about it is fixed as well; in particular in the case when f has the periodic structure required for Simon's problem there is a simple algorithm that identifies s, and this is the algorithm that writes s down starting from any initial state.

[7] To be fully precise we would need to fix a finite gate set for classical circuits and another for quantum circuits. What gate set is used will not matter for us; the only important point is that there exists finite universal gate sets and that all such gate sets are roughly equivalent in terms of how many gates are required to decompose any larger unitary.

For this reason we will always consider families of functions $\{f_{pk} : \{0,1\}^{m(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{pk \in \{0,1\}^{k(\lambda)}}$ where the index $\lambda \in \mathbb{N}$ is called *security parameter* and k and m are polynomially bounded functions of λ; the idea is that for each λ there is a collection of functions, indexed by strings of length $k(\lambda)$ and with the same domain and range, such that the larger the λ the more "complex" the functions are. For example, we could take $k(\lambda) = \lambda^2$, $m(\lambda) = \lambda$, and f_{pk} to be multiplication by the matrix $A \in \{0,1\}^{\lambda \times \lambda}$ obtained by "reshaping" the λ^2-bit string pk into a $\lambda \times \lambda$ square.

Let's give our first definition of a cryptographic property that applies to a family of functions.

Definition 1.6 (Claw-free function family)**.** A family $\mathcal{F} = \{f_{pk} : \{0,1\}^{m(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{pk \in \{0,1\}^{k(\lambda)}}$ is *claw-free* against classical (resp. quantum) adversaries if the following conditions hold:

- f_{pk} can be efficiently evaluated: there is a PPT procedure that given pk and x as inputs returns $f_{pk}(x)$.
- For every $\lambda \in \mathbb{N}$ and $pk \in \{0,1\}^{k(\lambda)}$, f_{pk} is 2-to-1.
- For every PPT (resp. QPT) procedure $\mathcal{A}$ the following holds: (the procedure $\mathcal{A}$ is often personified as the "adversary" trying to demonstrate that the function family is *not* claw-free) there exists a negligible[8] function $\mu : \mathbb{N} \to \mathbb{N}$ such that for every λ,

$$\Pr_{pk \leftarrow_R \{0,1\}^{k(\lambda)}} \left((x_0, x_1) \leftarrow \mathcal{A}(1^\lambda, pk) : \; x_0 \neq x_1, \; f_{pk}(x_0) = f_{pk}(x_1) \right) \leq \mu(\lambda) \, .$$

In words, the third condition states that there is no polynomial-time algorithm that given a uniformly chosen index pk for a function from the family is able to return two distinct inputs for the function that constitute a claw.[9]

Remark 1.7. In the definition we require the function family to be parametrized by arbitrary strings pk. In general this requirement can be relaxed; in fact there could even be a single function for every λ. In cryptographic constructions the function family generally comes equipped with a PPT *key generation procedure* GEN that takes 1^λ as input and returns pk.

An example of a claw-free family of functions against PPT adversaries can be constructed as follows. (This construction appears in [GMR85], where it is used to construct a digital signature scheme.) Let $N = pq$ be a product of two primes $p \equiv 3 \bmod 8$ and $q \equiv 7 \bmod 8$. This choice ensures that -1 and 2 are not squares $\bmod N$; moreover, if Q_N denotes the set of quadratic residues (i.e. squares) modulo N then $f_0(x) = x^2 \bmod N$ and $f_1(x) = 4x^2 \bmod N$ are both permutations of Q_N. (This fact requires proof but it is a simple exercise in arithmetic.) However, suppose given a claw (x_0, x_1) such that $x_0, x_1 \in Q_N$ and $f_0(x_0) = f_1(x_1)$. Then $x_0^2 = 4x_1^2$ but $x_0 \neq \pm 2x_1 \bmod N$ because $\pm 2x_1 \notin Q_N$. Thus computing the GCD of N with $x_0 \pm 2x_1$ recovers a nontrivial factor.

While this family of functions is claw-free with respect to PPT adversaries, it is clearly not claw-free against QPT adversaries, that can use Shor's algorithm to factor efficiently. We will construct such a function family in the next lecture; for the time being we assume its existence.

1.4.3 Hardcore bits

Following the initial steps of Simon's algorithm as described in Section 1.3.1 when instantiated with any 2-to-1 function enables a quantum device to generate strings $d \in \{0,1\}^m$ such that $d \cdot (x_0 + x_1) = 0$,

[8] A function $\mu : \mathbb{N} \to \mathbb{R}$ is called negligible if for every polynomial p, $p(\lambda)\mu(\lambda) \to_{\lambda \to \infty} 0$.

[9] A triple (x_0, x_1, y) such that $x_0 \neq x_1$ and $f(x_0) = f(x_1) = y$ is called a *claw*. To see why, picture the arrows $x_0 \to y$ and $x_1 \to y$ drawn with x_0, x_1 on top of each other on the left and y on the right.

where (x_0, x_1) are preimages of some $y \in \mathbb{F}_2^m$ by f. Intuitively one might expect that this represents a computational advantage, because d provides an equation in $x_0 + x_1$, which is some information about both preimages together. For example in the case where $x_0 + x_1 = s$, where s is some fixed secret independent of y, we saw that provided the equation d can be assumed to be uniformly distributed among all valid equations in s then running the procedure $O(m)$ times gives sufficiently many equations to recover s.

Unfortunately, as discussed in Section 1.3.2 for the only function that we could think of that has this property it is in fact easy to recover s, even for a classical computer. This suggests that in general the assumption that f satisfies the structure required for Simon's algorithm might be too strong to obtain an explicit candidate. Moreover, recall that at the start of the lecture we pointed out that our goal is not directly to find a task for which there is a quantum computational advantage, but instead we are trying to identify *two* tasks that the quantum device can perform *separately* but not *simultaneously*—if someone, such as a classical device, was able to execute both tasks simultaneously then it would break the computational assumption. What could those two tasks be here?

Starting from the state (1.4) it is natural to measure in the Hadamard basis, obtaining as before an equation d such that $d \cdot (x_0 + x_1) = 0$, but also in the computational basis, obtaining either x_0 or x_1. Given that "honest" measurements in the computational and hadamard basis are incompatible, these are natural candidates for our "qubit." However, we also saw that if $x_0 + x_1 = s$ for some fixed secret s then the Hadamard measurements alone allow us to recover s. So a quantum procedure could recover s "on the side" and then, knowing f explicitly, succeed in any reasonable "test" by using classical operations alone—this would make it very hard for us to identify the "qubit" that the device should have used to recover s (this is similar to the example of Shor's algorithm given at the start of the lecture). But what if the structure of f is a little more complicated, so that e.g. $x_0 + x_1 = g(x_0, s)$ for some function g? In that case a single equation in $g(x_0, s)$ might not be so useful; even many such equations for varying x_0 could be useless since without knowledge of x_0 itself one cannot determine what the equation is about. However, if one was able to obtain x_0 simultaneously with the equation then one would obtain a sequence of (possibly non-linear, depending on g) constraints on s. These considerations motivate the following computational assumption:

Assumption 1 (Adaptive hardcore bit)**.** *There is a claw-free family of functions $\mathcal{F} = \{f_{pk}\}$ such that for any QPT adversary $\mathcal{A}$ there is a negligible function μ such that*

$$\left|\frac{1}{2} - \Pr_{pk \leftarrow_R \{0,1\}^{k(\lambda)}}\left((x,d) \leftarrow \mathcal{A}(1^\lambda, pk),\ \{x_0,x_1\} \leftarrow f_{pk}^{-1}(f_{pk}(x)) :\ d \neq 0^m \wedge d \cdot (x_0 + x_1) = 0\right)\right| \leq \mu(\lambda)\ . \tag{1.5}$$

In words, the assumption is that no *quantum* polynomial-time algorithm can *simultaneously* return an element x in the domain of f and an equation d such that, letting $\{x_0, x_1\}$ be the two preimages of $f_{pk}(x)$ under f_{pk} it holds that $d \neq 0^m$ and $d \cdot (x_0 + x_1) = 0$. Note that although we required $\{f_{pk}\}$ to be claw-free, this requirement is stronger, since any algorithm that can find a claw (x_0, x_1) can be used to break (1.5).

Remark 1.8. Assumption 1 is called *adaptive hardcore bit* for the following reason. Given a function f a hardcore bit for f is a 1-bit function h such that given $f(x)$ (but not x) it is hard to predict $h(x)$. Here, the hardcore bit that underlies the assumption is the function $h(x) = d \cdot (x_0 + x_1)$ for any $d \neq 0^m$: the Goldreich-Levin theorem implies that if f is indeed claw-free then it is hard to predict $b(x)$ for a *random* d. The "adaptive" qualifier refers to the fact that in (1.5) we allow the adversary $\mathcal{A}$ itself to select the equation d without requiring that this equation is uniformly distributed (we will see why this is needed in the next section); in particular $\mathcal{A}$ may return always the same d, and this invalidates the classic Golreich-Levin argument. This makes the property harder to satisfy, because more power is given to the adversary. (Note

in particular that we had to explicitly require $d \neq 0^m$, as otherwise there is an easy adversary that always succeeds.)

1.5 A computational test for a qubit

We conclude the lecture by giving a "proof of concept" that it is possible to test a qubit based on computational assumptions. Our presentation is a simplified version of the protocol and analysis from [BCM+18]. For a related approach, see [CCKW19]. Recalling Definition 1.3, in order to do this we will have to demonstrate that any device successful in the protocol "has" (in a sense that will be made precise in the proof) two observables X and Z that are mutually incompatible. This incompatibility will be based on considerations of computational difficulty. Specifically we will show that, if the quantum device was able to measure X and Z jointly then it would break a computational problem that is assumed to be hard *even for quantum computers* — this is as a form of "computational uncertainty principle". Since by definition the device can measure X and Z separately, if they commuted then it could also measure them jointly. Therefore, the computational assumption gives rise to an *information-theoretic* consequence on the observables X and Z: they must form a qubit.

The main computational assumption that we make is the existence of a function family $\{f_{pk}\}$ that satisfies the hardcore bit assumption, Assumption 1. In fact we will need a little bit more. Let's summarize the requirements as follows:

(**F.1**) There is a 2-to-1 claw-free function family $\mathcal{F} = \{f_{pk}\}$ equipped with an efficient key generation procedure $\textsc{Gen}(1^\lambda)$ such that for each key pk the function f_{pk} can be evaluated efficiently.

(**F.2**) The function family $\mathcal{F}$ satisfies the adaptive hardcore bit assumption, Assumption 1.

(**F.3**) $\mathcal{F}$ is equipped with a trapdoor: in addition to pk, $\textsc{Gen}(1^\lambda)$ returns a trapdoor td such that given pk, td and any y in the range of f_{pk} it is possible to efficiently recover the two preimages x_0 and x_1 of y.

(**F.4**) For any pk and any y in the range of f_{pk} the two preimages of y are labelled 'x_0' and 'x_1' using some canonical efficient procedure. That is, given a key pk and an x in the domain of f_{pk} it is possible to efficiently determine if x is the 'x_0' or the 'x_1' preimage of $y = f(x)$. Let $b : \{0,1\}^m \to \{0,1\}$ be this labeling procedure; b may depend on pk.

Let us fix a function family $\mathcal{F}$ satisfying the assumptions (**F.1**) to (**F.4**). We give a protocol based on $\mathcal{F}$. The protocol describes the interaction between a classical polynomial-time verifier and a (possibly quantum) polynomial-time prover. Here, the input to both parties is the security parameter λ; when we refer to PPT or QPT we mean with respect to λ. The protocol is described in Figure 1.1. For future reference we refer to it as "protocol $\mathfrak{Q}$."

Theorem 1.9. *Let $\mathcal{F}$ satisfy the assumptions (**F.1**) to (**F.4**). Then the following hold for protocol $\mathfrak{Q}$.*

- *(Completeness:) There is a QPT prover P which succeeds with probability 1 in the protocol.*
- *(Soundness:) Suppose that a QPT prover P succeeds with probability 1 in the protocol. Then P has a (near-perfect) qubit.*

We note the slightly informal nature of the theorem and make a few comments:

- Combining Assumption 1 with the requirement that the prover P is QPT effectively means that we are assuming that P does not "have the ability" to violate (1.5). Slightly more formally, in the proof

Let $\mathcal{F}$ be a function family and $\lambda \in \mathbb{N}$ a security parameter.

1. The verifier generates $(pk, td) \leftarrow \text{GEN}(1^\lambda)$. It sends pk to the prover.
2. The prover returns $y \in \{0,1\}^m$, where $m = m(\lambda)$.
3. The verifier selects a uniformly random challenge $c \leftarrow_R \{0,1\}$ and sends c to the prover.
4. (a) *(pre-image test:)* In case $c = 0$ the prover is expected to return an $x \in \{0,1\}^m$. The verifier accepts if and only if $f(x) = 1$.
   (b) *(equation test:)* In case $c = 1$ the prover is expected to return a $d \in \{0,1\}^m$. The verifier uses td to determine the two preimages (x_0, x_1) of y by f_{pk}. She accepts if and only if $d \cdot (x_0 + x_1) = 0$.

Figure 1.1: Protocol $\mathfrak{Q}$, the computational test for a qubit. The protocol is parametrized by a function family $\mathcal{F}$ satisfying assumptions (**F.1**) to (**F.4**).

we will show that if P *does not* "have a qubit" then it can be used to construct an adversary $\mathcal{A}$ that violates (1.5). Note also that in the soundness case it should be assumed that P is in fact a family of $\{P_\lambda\}$, one for each possible choice of λ, that can be uniformly generated from λ (i.e. there is a classical Turing machine that takes 1^λ as input and returns a description of a family of circuits that can be used to implement P_λ).[10]

- Second, we ought to be a little more precise as to how P's qubit is specified. The two observables X and Z that define it will be derived from the two measurements that P makes based on the challenges $c = 0$ or $c = 1$. Since these measurements in general have outcomes in $\{0,1\}^n$ some post-processing will be required. Interestingly, the post-processing for the X observable will not be efficient, in the sense that it will require knowledge of td. So, our proof will show that there exists two anti-commuting observables on the Hilbert space of P that can be defined from P's operations *and some classical post-processing*. Since the post-processing is classical we can claim in good faith that the "qubit" is located on the prover's space, as we are not injecting any external "quantumness" in it.
- Regarding the assumption that the prover succeeds with probability 1: this assumption is, of course, unrealistic. The assumption can be lifted at the cost of some amount of work, which we discuss in more detail when we build on the present protocol to construct a more complex protocol for verifying an entire quantum computation in the next lectures.
- Finally, an explanation is in order regarding the "(near-perfect)" qualifier. This is an unavoidable consequence of the fact that the protocol relies on a computational assumption. Indeed, consider the following possible behavior for the prover. The prover first devotes a small amount of time to trying their luck at breaking the underlying computational assumption (in our case, the prover could randomly generate candidate trapdoors td' and check if they allow it to invert the function f_{pk}). If the prover succeeds then it can pass in the protocol without manipulating any quantum state, using the fake td' to find a claw that allows it to answer both types of challenges. If it does not succeed then it behaves honestly in the protocol. Such a prover succeeds with probability 1, but the measurement

[10] Non-uniform adversaries are allowed as long as we make the corresponding non-uniform cryptographic assumption.

operators associated with its answers have a part that is "classical" and from which we have no hope of extracting a qubit.

Proof of Theorem 1.9. The completeness part of the theorem is clear. In the first phase the prover proceeds exactly as in Simon's algorithm to obtain the state (1.4). In the second phase, it measures the preimage register in the standard basis in case $c = 0$ and in the Hadamard basis in case $c = 1$, returning the n-bit outcome obtained as its answer. This prover is always accepted with probability 1 in the protocol.

To show the soundness part of the theorem we start with the usual (and, here, crucial) modeling step.

Step 1: Modeling. Since we will not need to model the prover's actions in the first phase of the protocol in detail we directly give a name to the state of the prover at the end of step 2; let it be $|\psi\rangle \in \mathcal{H}_{\mathsf{P}}$. This state depends on pk as well as on y; for clarity we suppress this dependence from the notation. Moreover, in general $|\psi\rangle$ may be a mixed state, and we represent it as a pure state for convenience only; in general one could assume that we included a register E to denote an "environment" that holds a purification $|\psi\rangle_{\mathsf{PE}}$ of a general $\rho \in \mathrm{D}(\mathcal{H}_{\mathsf{P}})$.

At the second stage of the protocol the prover is given a challenge $c \in \{0,1\}$ and tasked with responding with an n-bit string, x or d depending on the challenge. In general, x is obtained by performing a POVM $\{\Pi_x\}$ on the prover's entire space, and similarly d is the outcome of a POVM $\{M_d\}$.[11] We make the following observations that allow us to simplify the presentation of these POVM:

- Without loss of generality both Π and M are projective measurements. This is because we can enlarge P and add sufficiently many ancilla qubits initialized to $|0\rangle$ so as to apply Naimark's theorem.
- Without loss of generality, assume that the prover has access to an m-qubit register X initialized to $|0^m\rangle$.
- Without loss of generality, assume that Π is obtained by first applying a unitary transformation U_0 on $\mathcal{H}_{\mathsf{X}} \otimes \mathcal{H}_{\mathsf{P}}$ followed by a standard basis measurement of $\mathcal{H}_{\mathsf{X}} \simeq (\mathbb{C}^2)^{\otimes m}$. Any projective measurement can be put in this form by letting U_0 be any unitary extension of the map
$$|0\rangle|\psi\rangle \in \mathcal{H}_{\mathsf{X}} \otimes \mathcal{H}_{\mathsf{P}} \;\mapsto\; \sum_x |x\rangle\sqrt{\Pi_x}|\psi\rangle \;,$$
as this map is easily verified to be an isometry on $|0\rangle_{\mathsf{X}} \otimes \mathcal{H}_{\mathsf{P}}$.
- Similarly, without loss of generality assume that M is obtained by first applying a unitary transformation U_1 on $\mathcal{H}_{\mathsf{X}} \otimes \mathcal{H}_{\mathsf{P}}$ followed by a Hadamard basis measurement of $\mathcal{H}_{\mathsf{X}}$.
- Without loss of generality assume that $U_0 = \mathrm{Id}$. This is because we can always redefine the prover's state at the end of step 2 to be $|\psi'\rangle = U_0|\psi\rangle$, in which case $U_0' = \mathrm{Id}$ and $U_1' = U_1 U_0^\dagger$. Since $U_0 = \mathrm{Id}$, we simply use U to denote U_1.

We now introduce observables Z and X on $\mathcal{H}_{\mathsf{X}}$ associated with the prover. For Z, we define it to be

$$Z = \sum_{x\in\{0,1\}^m} (-1)^{b(x)} |x\rangle\langle x| \otimes \mathrm{Id}_{\mathsf{P}} \;, \tag{1.6}$$

where $b : \{0,1\}^n \to \{0,1\}$ is the function from assumption (**F.4**). Z is efficiently computable since b is. For X, we define it to be

$$X = \sum_{d\in\{0,1\}^m} (-1)^{d\cdot(x_0+x_1)}\, U^\dagger \big(H_{\mathsf{X}}^{\otimes m} \otimes \mathrm{Id}_{\mathsf{P}}\big)^\dagger \big(|d\rangle\langle d|_{\mathsf{X}} \otimes \mathrm{Id}_{\mathsf{P}}\big)\big(H_{\mathsf{X}}^{\otimes m} \otimes \mathrm{Id}_{\mathsf{P}}\big) U \;, \tag{1.7}$$

[11] We do not need to explicitly mark any dependence of Π or M on pk and y, because without loss of generality the prover has kept a classical copy of these strings in its quantum state $|\psi\rangle$, which can be used as a classical control by both Π and M.

where x_0 and x_1 are the two preimages under f_{pk} of the string y returned by P at step 2. (There is an observable X for each possible string y, but we suppress this dependence from the notation for clarity.) Note that X is *not* efficient, because we are not assuming that determining $x_0 + x_1$ from y is efficient in general. However, X can be computed in a straightforward manner by applying the prover's efficient measurement $\{M_d\}$ followed by (non-efficient) classical post-processing. (We insist on this point to clarify that our definition is not injecting "quantumness" artificially.) Informally, X can be thought of as the observable that determines if the equation d returned by the prover on challenge $c = 1$ is correct or not. In particular, later we will use that for a prover that always succeeds to a challenge $c = 1$ we have $X|\psi\rangle = |\psi\rangle$, i.e. $|\psi\rangle$ is a $+1$ eigenstate of X.

Step 2: Establishing a qubit. The goal for the remainder of the proof is to show that $(|\psi\rangle, Z, X)$ form a qubit, i.e. that the two observables X and Z anticommute on $|\psi\rangle$. Informally, this is because if X and Z were jointly measurable then they could be used to simultaneously obtain a preimage of y and a valid equation d in $x_0 + x_1$, thereby violating (**F.2**). We proceed with the details. The heart of the proof is the following claim.

Claim 1.10. *For any $b \in \{0,1\}$,*

$$\big|\langle\psi|Z_b X Z_b|\psi\rangle\big| \;=\; \mathrm{negl}(\lambda)\;,$$

where $Z_b = (\mathrm{Id} + (-1)^b Z)/2$, $\mathrm{negl}(\lambda)$ denotes some negligible function of λ, and the expression on the left should be understood on average over $pk \leftarrow \textsc{Gen}(1^\lambda)$ and the distribution of y as returned by P in the protocol.

Proof. We do the proof for the case $b = 0$, the other case being similar. Suppose for contradiction that there is a polynomial $q : \mathbb{N} \to \mathbb{R}_+$ such that

$$\big|\langle\psi|Z_0 X Z_0|\psi\rangle\big| \;\geq\; \frac{1}{q(\lambda)} \tag{1.8}$$

for infinitely many values of λ. We use this assumption to construct an adversary in (1.5). The adversary proceeds as follows. Given as input 1^λ and pk the adversary first executes the first phase of the prover, obtaining an outcome y and a state $|\psi\rangle$. Then, the adversary measures the m qubits in register X in the computational basis to obtain a value $x \in \{0,1\}^m$. If $b(x) = 0$ then the adversary applies the unitary V and measures register X (again) in the Hadamard basis to obtain $d \in \{0,1\}^n$. The adversary returns the pair (x, d). If $b(x) = 1$ then the adversary chooses $d \in \{0,1\}^m$ uniformly at random and returns d. Since the prover P and b are both efficient, $\mathcal{A}$ is efficient.

Note that this adversary does something "unusual" in the sense that it sequentially applies two operators that the prover would never have applied simultaneously in the protocol. It is to make sense of this sequential application that we made the structural simplifications at the start of the proof. Let's analyze the success probability of $\mathcal{A}$ by using (1.8). There are two cases. Suppose first that the adversary obtains an x such that $b(x) = 0$. Then since P is assumed to succeed with probability 1 in case $c = 0$, we know that necessarily $x = x_0$, and moreover prior to the measurement the support of $|\psi\rangle$ on X contained only the two values $|x_0\rangle$ and $|x_1\rangle$ (as otherwise there would be a chance that the prover returns an invalid preimage to the challenge $c = 0$). Thus by definition of Z in (1.6) the post-measurement state is $Z_0|\psi\rangle$ (suitably re-normalized). The probability that $\mathcal{A}$ obtains $b(x) = 0$ and then a correct equation is then, by definition of X in (1.6) and $X_0 = \frac{1}{2}(\mathrm{Id} + X)$, precisely

$$\langle\psi|Z_0 X_0 Z_0|\psi\rangle \;=\; \frac{1}{2}\big(\langle\psi|Z_0|\psi\rangle + \langle\psi|Z_0 X Z_0|\psi\rangle\big)\;.$$

For the second case assume that $\mathcal{A}$ obtains an x such that $b(x) = 1$. In this case it returns a uniformly random equation; since $x_0 + x_1 \neq 0$ this has probability exactly $\frac{1}{2}$ of being correct. Overall, the adversary's success probability is

$$\frac{1}{2}\langle\psi|Z_1|\psi\rangle + \frac{1}{2}\big(\langle\psi|Z_0|\psi\rangle + \langle\psi|Z_0XZ_0|\psi\rangle\big) \;=\; \frac{1}{2} + \frac{1}{2}\langle\psi|Z_0XZ_0|\psi\rangle \;,$$

where the equality uses $Z_0 + Z_1 = \mathrm{Id}$ to combine the first two terms. Using (1.8), this violates (1.5). □

To conclude the proof of the theorem we need the following simple calculation.

Claim 1.11. *Let X, Z be any two binary observables on $\mathcal{H}$. Then*

$$\frac{1}{4}\{X, Z\}^2 \;=\; XZ_0XZ_0 + Z_1XZ_1X \;.$$

Proof. This can be verified by direct calculation. Using that X and Z are Hermitian and square to identity we get by expanding the square

$$\{X, Z\}^2 = 2 + XZXZ + ZXZX \;. \tag{1.9}$$

Expanding $Z = Z_0 - Z_1$,

$$ZXZ = Z_0XZ_0 + Z_1XZ_1 - Z_0XZ_1 - Z_1XZ_0 \;.$$

Moreover, using $Z_0 + Z_1 = \mathrm{Id}$ it follows that

$$Z_0XZ_0 + Z_1XZ_1 + Z_0XZ_1 + Z_1XZ_0 = X$$

Putting the two equations together, $ZXZ = 2\big(Z_0XZ_0 + Z_1XZ_1\big) - X$. Plugging back into (1.9) and using $X^2 = \mathrm{Id}$ proves the claim. □

Combining Claim 1.11 and Claim 1.10 we make the following calculation:

$$\begin{aligned}\frac{1}{4}\big\|\{X, Z\}|\psi\rangle\big\|^2 &= \langle\psi|XZ_0XZ_0|\psi\rangle + \langle\psi|Z_1XZ_1X|\psi\rangle \\ &= \langle\psi|Z_0XZ_0|\psi\rangle + \langle\psi|Z_1XZ_1|\psi\rangle \\ &= \mathrm{negl}(\lambda) \;,\end{aligned}$$

where to obtain the second line we used that $X|\psi\rangle = |\psi\rangle$ since the prover is assumed to succeed with probability 1 in the protocol (and hence always return a correct equation). This shows that $(|\psi\rangle, Z, X)$ is a near-perfect qubit, completing the proof. □

Lecture 2

Delegating Quantum Computations

In this lecture we give a general introduction to the problem of delegating quantum computation, and then present a specific protocol due to Fitzsimons and Morimae that we will build on in the following lectures. Most of the material in the lecture is standard and may be skipped by some readers, who may nevertheless wish to briefly read over the description of the Fitzsimons-Morimae protocol in Section 2.2.2 to make sure they are comfortable with the notation introduced for it.[1]

2.1 Problem statement

2.1.1 Quantum circuits and the class BQP

For us, a quantum circuit $\mathcal{C}$ is specified by an integer n and an ordered sequence of elements of the form (G, i, j) where $G \in \{H, CNOT, T\}$ and $i, j \in \{1, \ldots, n\}$. Letting

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, \tag{2.1}$$

the circuit $\mathcal{C} = ((G_1, i_1, j_1), \ldots, (G_T, i_k, j_k))$ represents the unitary U on $(\mathbb{C}^2)^{\otimes n}$ obtained as $U = U_T \cdots U_1$ where for all $t \in \{1, \ldots, T\}$, U_t acts as the unitary associated with G_t in (2.1) on qubits i_t and j_t and as identity on the other qubits. (In case $G_t \in \{H, T\}$ it is required that $i_t = j_t$.) The Solovay-Kitaev theorem shows that any n-qubit unitary can be arbitrary well-approximated, in operator norm, by the unitary derived from a circuit; however, the size of the circuit may (in fact, must) in general grow exponentially fast with n. Those unitaries that can be represented by small circuits are called "efficient".

Given a quantum circuit $\mathcal{C}$ acting on n qubits and $x \in \{0,1\}^m$ for some $m \leq n$ we say that "$\mathcal{C}$ accepts input x with probability p" if the probability of obtaining the outcome 1 after a measurement in the computational basis of the first qubit of the n-qubit state obtained by applying the unitary $\mathcal{C}$ to the input state $|x\rangle|0^{n-m}\rangle$ is p.

[1] Some of the material for this lecture is taken from an overview of Mahadev's result written for a mathematical audience and published in the Bulletin of the AMS [Vid20]. Some of it is reproduced from lecture notes prepared for a winter school at UCSD: `http://cseweb.ucsd.edu/~slovett/workshops/quantum-computation-2018/`.

Definition 2.1. We say that a promise language $L = (L_{yes}, L_{no})$ is in BQP if there exists a family of polynomial-time generated quantum circuits $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ such that for all integer n and $x \in \{0,1\}^n$,[2]

- (Completeness:) If $x \in L_{yes}$ then $\mathcal{C}_n$ accepts x with probability at least $\frac{2}{3}$;
- (Soundness:) If $x \in L_{no}$ then $\mathcal{C}_n$ accepts x with probability at most $\frac{1}{3}$.

Note the requirement that the family $\{\mathcal{C}_n\}$ is polynomial-time generated. This means that there exists a classical Turing Machine that on input 1^n runs in time $\mathrm{poly}(n)$ and returns a description of $\mathcal{C}_n$ as a sequence of gates taken from a fixed universal set—here we use (2.1), but the specific choice will not matter for us.

The definition of BQP sets arbitrary values $2/3$ and $1/3$ for the completeness and soundness parameters. Error amplification works just as for the case of BPP, by repeating the circuit sequentially. This requires intermediate measurements, but it is not hard to show that these can be postponed till the end of the computation by the use of ancilla qubits and CNOT gates. As a result, any choice of a, b such that $a - b > \mathrm{poly}^{-1}(n)$ gives the same definition: for any such a, b, and for any fixed polynomial q, $\mathrm{BQP}(a,b) = \mathrm{BQP} = \mathrm{BQP}(1-2^{-q}, 2^{-q})$.

Exercise 2.1. Show that BQP is included in PP, the class of languages for which there exists a probabilistic Turing machine that accepts YES inputs with probability $> 1/2$, and rejects NO inputs with probability $> 1/2$. (Hint: first show inclusion in PSPACE by giving space-efficient implementations of basic linear algebra operations. Inclusion in PP follows from similar arguments, but is a bit more delicate.)

The class PP lies outside of the polynomial hierarchy. The most commonly-held belief is that the intersection of BQP and PH is non-trivial: it is neither BPP, nor PH itself. Recently Raz and Tal [RT19] showed that an oracle problem introduced by Aaronson [Aar10] is in BQP but not in PH.

We end this section by defining a family of complexity classes associated with interactive proof systems.

Definition 2.2 (Adapted from [AG17])**.** Given complexity classes $\mathcal{P}$ and $\mathcal{Q}$, $\mathrm{IP}[\mathcal{P}, \mathcal{Q}]$ is the class of (promise) languages L such that there is a polynomial-time Turing machine M that on input 1^n returns the description of classical circuits for the verifier V_n in an interactive protocol with a prover P such that

- (Completeness:) There is a family of provers $\{P_n\}_{n\in\mathbb{N}}$ that lie in the class $\mathcal{P}$ such that for all $x \in L$ the interaction of $V_{|x|}$ and $P_{|x|}$ on common input x accepts with probability at least $\frac{2}{3}$.
- (Soundness:) For any family of provers $\{P_n\}_{n\in\mathbb{N}}$ that lie in the class $\mathcal{Q}$, for all $x \in L_{no}$ the interaction of $V_{|x|}$ and $P_{|x|}$ on common input x accepts with probability at most $\frac{1}{3}$.

When the classes $\mathcal{P}$ and $\mathcal{Q}$ coincide we simply write $\mathrm{IP}[\mathcal{P}]$ for $\mathrm{IP}[\mathcal{P}, \mathcal{P}]$. We use the standard notation $\mathrm{IP} = \mathrm{IP}[\mathrm{BPP}, \mathrm{ALL}]$ with ALL the class of all languages (i.e. soundness is proves without any restriction on the prover).

The definition is slightly informal, because for some classes $\mathcal{P}$ it may not be clear what it means for the prover to lie in $\mathcal{P}$. For us the meaning will always be clear from context, as $\mathcal{P}$ and $\mathcal{Q}$ will always be either BPP, BQP or ALL.

2.1.2 Delegating quantum computations

The fact that BQP is not (believed to be) in NP implies that in general we do not expect there to exist classically verifiable proofs for the correctness of an arbitrary quantum computation. This poses a challenge:

[2]Note that in general, $\mathcal{C}_n$ may act on $\mathrm{poly}(n)$ qubits, the first n of which are by convention destined to receive the input x and the first of which also serves as output qubit.

as we see quantum computers emerging, how will we test their predictions? This is a practical problem — will anyone trust the "quantum cloud" — but also a philosophical one — is quantum mechanics a testable theory? (For more on this, see [AV13].)

Not all is lost. What we *do* know is that BQP is included in PSPACE, the class of languages that can be decided using polynomial space (and arbitrary time); in fact Exercise 2.1 asked you to show a stronger statement. And even though it is not a trivial result, it is known that PSPACE $=$ IP. So all languages in BQP have *classical* interactive proofs, with an efficient classical verifier! Unfortunately there is a major caveat to this observation. The proof that PSPACE is in IP is based on the classical SUM-CHECK protocol, which in general requires the server to execute PSPACE-complete computations (essentially, the server has to compute exponentially large sums in order to determine answers that will satisfy the client). (For an exposition of the proof we refer to the book [AB09].)

So, even though a protocol exists, it is unknown if there is such a protocol in which a honest server is only required to have the power of BQP. Today this is a major open question:

Open Question 2.3. Is $\mathrm{BQP} \subseteq \mathrm{IP}[\mathrm{BQP}, \mathrm{ALL}]$? In words, do all languages in BQP have single-server interactive proofs in which the client has the power of BPP and for which completeness holds with a BQP server and soundness holds against any server?

There are some partial impossibility results [ACGK17] on this question, as well as possibility results where completeness holds for provers that require more power than BQP but not necessarily the entire power of PSPACE; see e.g. [AG17]. If, however, one allows slightly more power to the verifier then there are scenario in which the question is known to have a positive answer:

1. The client has access to a limited quantum computer, such as the ability to prepare single qubits in arbitrary states and send them to the server, or receive single qubits from the server and make simple measurements on them;

2. The client is allowed to interact with multiple quantum servers sharing entanglement.

The question as formulated above asks for *verifiable* delegation: given a quantum circuit (deciding some BQP language L), is there a protocol that allows a classical client to extract the outcome of the circuit from a BQP server, in a way that any cheating server, attempting to convince the client of the wrong outcome, will be detected? A second desirable property of a delegation protocol is *blindness*: while the client would like to learn the valid outcome of her circuit, she might not want to disclose the particular circuit or input she is interested in to the server. This is a distinct property from verifiability; in particular, one may ask for blindness in the "honest-but-curious" model, where verifiability is trivial. The following definition introduces these properties slightly more formally.

Definition 2.4 (Delegated computation)**.** In the task of delegated computation, a client (sometimes called the *verifier*) has an input $(x, \mathcal{C})$, where x is a classical string and $\mathcal{C}$ the classical description of a quantum circuit. The client has a multiple-round interaction with a quantum server (sometimes also called *server*). At the end of the interaction, Alice either returns a classical output y, or she aborts. A protocol for delegated computation is called:

- *Correct* if whenever both the client and the server follow the protocol, with high probability Alice accepts (she does not abort) and $y = \mathcal{C}(x)$. (This property is sometimes called *completeness*.)
- *Verifiable* if for any server deviating from the protocol, the client either aborts or returns $y = \mathcal{C}(x)$. (This property is analogous to what we have been calling *soudness*.)

- *Blind* if for any server deviating from the protocol, at the end of the protocol the server has no information at all about the client's input $(x, \mathcal{C})$.

The definition remains rather informal. For example, how should we formalize the "information" that the server has at the end of the computation? This can be rather delicate, especially once one starts taking into account a small chance ε of deviation from the perfect properties. A precise definition satisfying all the desired properties (universal composability in particular) would take us too far. Such a definition was given using the framework of *abstract cryptography* in [DFPR14].

The informal definition will be sufficient for our purposes. Note that in spite of being rather similar neither of the properties of verifiability or blindness is known to directly implies the other. In practice verifiability often follows from blindness by arguing, using "traps", that if a protocol is already blind then the server's trustworthiness can be tested by making it run "dummy" computations for which Alice already knows the output, without the server being able to distinguish whether it is asked to do a real or dummy computation. We will see an example of this technique later on.

Open Question 2.5. Is there a general transformation from any protocol satisfying blindness, to a protocol satisfying both verifiability and blindness? See [KMW17] for a possible approach.

Remark 2.6. The problem of delegating computation is interesting even for classical computation. In this case the client herself could directly execute the classical circuit $\mathcal{C}$. But it makes sense to be even more demanding, and seek protocols where the client is super-efficient: the best we could hope for is a client that runs in time *linear* in the input length, and independent of the size of the circuit. In addition, we would like the overhead for the server to be as small as possible, so that the honest behavior requires a server effort of the same order as the size of the circuit, $|\mathcal{C}|$. This kind of interactive proofs are called *doubly efficient* interactive proofs [GKR08]. The paper [RRR16] shows how to achieve such proofs with client runtime that is linear in the input length, polynomial in the space required by $\mathcal{C}$, and polylogarithmic in $|\mathcal{C}|$. If one is willing to make computational assumptions (essentially, subexponential LWE) then even more efficient delegation is possible [KRR14], with client runtime that is linear in the input size and poly-logarithmic in $|\mathcal{C}|$.

These results usually do not put emphasis on the requirement of blindness: they focus on verifiability alone. One reason for this is that blindness is "trivially solved" by employing homomorphic encryption [Gen09]. This, however, requires computational assumptions, and induces significant computational overhead.

2.2 The Fitzsimons-Morimae protocol

We describe the receive-and-measure protocol from [MF16], as it will form the basis for the Mahadev protocol.

2.2.1 The circuit-to-Hamiltonian reduction

The Cook-Levin theorem showing NP-completeness of the 3SAT problem is based on what could be called a "circuit-to-formula" reduction: given a classical circuit, the computation performed by the circuit on some input is represented as a "tableau" such that the property of being a valid tableau can be encoded in a formula whose variables represent the state of any given wire in the circuit and whose constraints enforce correct propagation of the gates of the circuit.

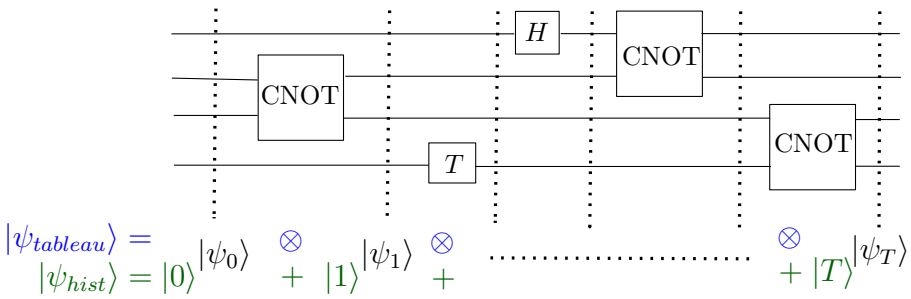

Figure 2.1: Two different ways to create a tableau from a quantum circuit. The state $|\psi_{tableau}\rangle$ is the tensor product of the state of the circuit at each time step. The state $|\psi_{hist}\rangle$ is their superposition, indexed by a clock register that goes from $|0\rangle$ to $|T\rangle$.

For quantum circuits the idea of a tableau of the computation is less straightforward. The most direct analogue is to consider the juxtaposition of the quantum state of a T-gate circuit at each step of the computation, i.e. the tensor product $|\psi_0\rangle \otimes \cdots \otimes |\psi_T\rangle$ of the states $|\psi_i\rangle$ obtained by executing the circuit from scratch and stopping after i gates have been applied. While this is a well-defined $n(T+1)$-qubit quantum state (see Figure 2.1) the property of being a valid "quantum tableau" cannot be enforced using *local* constraints! The reason is subtle, and has to do with the possible presence of entanglement at intermediate steps of the computation. Indeed, there are quantum states that are very different, in the sense that they are perfectly distinguishable by some *global* observable, yet cannot be distinguished at all by any *local* observable, that would act on at most, say, half the qubits. An example is given by the two n-qubit "cat" (named after the homonymous animal) states

$$|\psi_\pm\rangle = \frac{1}{\sqrt{2}}\big(|0\cdots 0\rangle \pm |1\cdots 1\rangle\big) .$$

The two states $|\psi_+\rangle$ and $|\psi_-\rangle$ are easily seen to be orthogonal, so that they can be perfectly distinguished by a measurement. But it is an exercise to verify that for any observable that acts on at most $(n-1)$ of the n qubits, both states give exactly the same expectation value. (Informally, this is because any measurement on a strict subset of the qubits of the state necessarily destroys the coherence; the only relevant information, the $\pm$ sign, is encoded "globally" and cannot be accessed locally.) Note that this is a uniquely quantum phenomenon: if two classical strings of bits have each of their bits equal, one pair at a time, then the strings are "globally" identical. Not so for quantum states.

So naïve tableaus will not do. In the late 1990s Alexei Kitaev introduced a very powerful idea that provides a solution. Kitaev's idea is to replace the juxtaposition of snapshot states by their *superposition* (see Figure 2.1). A special ancilla system, called the "clock", is introduced to index different elements of the superposition. Thus, instead of defining a tableau as $|\psi_0\rangle \cdots |\psi_T\rangle$, Kitaev considers the state

$$|\psi_{hist}\rangle \;=\; \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle|\psi_t\rangle \; . \tag{2.2}$$

Note that this takes less qubits to store, but this is not the important point. Kitaev showed that, assuming the clock register is encoded in unary, it is possible to check the correct propagation of every step of the circuit directly on this superposition by only applying local observables: there is a set of observables H_{in} that checks that $|\psi_0\rangle$ has the right format; a set of observables H_{prop} that checks propagation of the circuit,

and an observable H_{out} that checks that the output qubit of the circuit is in the right state. (In addition, there is a term H_{clock} that checks that the clock register is well-formed, i.e. contains the representation of an integer in unary. This can be done locally by penalizing configurations of the form "$\cdots 10 \cdots$".) The key point that makes this possible is that, while equality of quantum states cannot be decided locally when the states are juxtaposed, it becomes possible when they are given in superposition. As an exercise, we can verify that a measurement of the first qubit of the state

$$|\psi_{SWAP}\rangle \;=\; \frac{1}{\sqrt{2}}\big(|0\rangle|\psi_0\rangle + |1\rangle|\psi_1\rangle\big)$$

in the Hadamard basis $\{|+\rangle, |-\rangle\}$ returns the first outcome with probability exactly $\frac{1}{2}(1 + |\langle\psi_0|\psi_1\rangle|^2)$. With more work, replacing the use of gadgets in the classical Cook-Levin reduction by techniques from perturbation theory, it is possible to write the resulting observables as a linear combination of local terms that all take a particularly simple form. The result is the following theorem from [CM16].

Theorem 2.7. *For any integer $n \geq 1$ there are $n' = \mathrm{poly}(n)$, $a = a(n)$ and $\delta \geq 1/\mathrm{poly}(n)$ such that the following holds. Given a T-gate quantum circuit $\mathcal{C} = ((G_1, i_1, j_1), \ldots, (G_T, i_T, j_T))$ acting on n qubits, such that $T = \mathrm{poly}(n)$, and an input x for the circuit, there exist efficiently computable real weights $\{J_{ij},\ i, j \in \{1, \ldots, n'\}\}$ such that $|J_{ij}| \leq 1$ for all i, j and moreover if*

$$H_{\mathcal{C}} \;=\; -\sum_{i,j} \frac{J_{ij}}{2}\left(\sigma_{X,i}\sigma_{X,j} + \sigma_{Z,i}\sigma_{Z,j}\right), \tag{2.3}$$

where $\sigma_{X,i}$ and $\sigma_{Z,j}$ denote single-qubit Pauli X and Z operators acting on the i-th and j-th qubit respectively, then:

- *(Completeness) If the circuit $\mathcal{C}$ accepts its input x with probability at least $2/3$, then the smallest eigenvalue of $H_{\mathcal{C}}$ is at most a;*
- *(Soundness) If the circuit $\mathcal{C}$ accepts its input x with probability at most $1/3$, then the smallest eigenvalue of $H_{\mathcal{C}}$ is at least $a + \delta$.*

Remark 2.8. It is possible to modify Theorem 2.7 so that the completeness and soundness statements specify that "if there exists a state $|\phi\rangle$ such that $\mathcal{C}$ accepts on input $(x, |\phi\rangle)$ with probability at least $2/3$..." and "if there does not exist a state $|\phi\rangle$ such that $\mathcal{C}$ accepts on input $(x, |\phi\rangle)$ with probability greater than $1/3$..." respectively. Thus, Theorem 2.7 can be adapted to show that the problem of estimating the minimal energy of a Hamiltonian of the form (2.3) is a QMA-complete problem.

Theorem 2.7 provides us with a roadmap for the verification of quantum circuits: it is sufficient to verify the *existence* of a quantum state that yields certain statistics, when some of its qubits are measured in the computational (σ_Z observable) or Hadamard (σ_X observable) basis. The reason this can be considered progress is that we no longer need to check the time evolution of a quantum state under a quantum cicuit; it is sufficient to collect measurement statistics and estimate the "energy" $\langle\psi|H|\psi\rangle$. In particular, the theorem readily leads to a verification protocol in a model where the prover has a full quantum computer, and the verifier only has a limited quantum device — namely, a one-qubit memory, together with the ability to measure the qubit using either the σ_X or σ_Z observables.

2.2.2 The protocol

Such a verification protocol was introduced by Fitzsimons and Morimae and refined in a paper with Hadjušek. The protocol is summarized in Figure 2.2. In the protocol, the prover is required to prepare a smallest eigenstate of the Hamiltonian $H_{\mathcal{C}}$ given in (2.3). While it may not be immediately obvious at the level of our

description, it is possible to prepare such a "history state" (2.2) by executing a quantum circuit that is only mildly more complex than the original circuit $\mathcal{C}$.

Let $\mathcal{C}$ be a quantum circuit provided as input, and $H_{\mathcal{C}}$ the n-qubit Hamiltonian obtained from $\mathcal{C}$ as in (2.3).

1. The verifier initializes a counter γ to 0. She executes the following interaction with the prover independently $N = \frac{C}{\delta^2}\binom{n'}{2}^2 \ln(1/\varepsilon)$ times, where C is a large enough universal constant:
   (a) The prover creates an eigenstate $|\psi\rangle$ of H with smallest eigenvalue.
   (b) The prover sends the qubits of $|\psi\rangle$ one by one to the verifier.
   (c) The verifier selects a measurement $W \in \{X, Z\}$ uniformly at random, and measures each qubit in the associated basis upon reception. Let $b_{W,i} \in \{-1, 1\}$ be the outcome for the i-th qubit.
   (d) The verifier selects $i \neq j \in \{1, \ldots, n'\}$ uniformly at random. She updates her counter $\gamma \leftarrow \gamma - J_{ij} b_{W,i} b_{W,j}$.
2. If $\frac{\gamma}{N}\binom{n'}{2} \leq a + \delta/2$ the verifier accepts the interaction. Otherwise, she rejects.

Figure 2.2: The Fitzsimons-Hadjuček-Morimae verification protocol, parametrized by a quantum circuit $\mathcal{C}$ and an accuracy parameter $\varepsilon > 0$.

We note that in the protocol, the verifier measures the qubits in a randomly chosen basis, and then selects a single pair (i, j) such that $J_{ij} \neq 0$ uniformly at random to update her counter. One could imagine small optimizations where e.g. a maximum matching of such pairs is measured at each step. Such optimizations only bring marginal improvements in efficiency of the protocol; moreover they complicate the extension to a classical verifier that we will see later. For this reason, we prefer to keep the simplest expression possible for the protocol.

Theorem 2.9. *Let $\mathcal{C}$ be a quantum circuit and $H_{\mathcal{C}}$ the Hamiltonian associated to it as in* (2.3)*. Let x be an input to the circuit $\mathcal{C}$ and $\varepsilon > 0$ a parameter for the protocol. Then the following hold:*

- *(Completeness:) If $\mathcal{C}$ accepts x with probability at least $2/3$, then there is a QPT prover that is accepted with probability at least $1 - \varepsilon$*
- *(Soundness:) If $\mathcal{C}$ accepts x with probability at most $1/3$, then any prover is accepted with probability at most ε.*

Note that in the theorem, the soundness statement does not place any computational assumption on the prover.

Proof. The key calculation that underlies the proof is the following.

Claim 2.10. *Let ρ be the density matrix that represents the mixture over the N n'-qubit states sent by the prover in the protocol (in general these states may be entangled). Then the expectation of γ/N is exactly*

$$\mathrm{E}\Big[\frac{\gamma}{N}\Big] \;=\; -\frac{1}{\binom{n'}{2}} \sum_{i \neq j} \frac{J_{ij}}{2} \operatorname{Tr}\big((\sigma_X^i \sigma_X^j + \sigma_Z^i \sigma_Z^j)\rho\big) \;=\; \frac{1}{\binom{n'}{2}} \operatorname{Tr}\big(H\rho\big)\,. \tag{2.4}$$

Moreover, for N chosen as in the protocol for a large enough choice of the constant C it holds that

$$\Pr\left(\left|\frac{\gamma}{N}\binom{n'}{2} - \mathrm{Tr}(H\rho)\right| > \frac{\delta}{2}\right) \leq \varepsilon\,. \tag{2.5}$$

Proof. For $t \in \{1,\ldots,n\}$ let G_t denote the product of the two outcomes $b_{W,i}$ and $b_{W,j}$ obtained by the verifier at step (c) of the protocol, where W, i and j are as sampled at step (d). Then the random variables $G_t \in \{-1,1\}$ are i.i.d. such that for each t, $\mathrm{E}[G_t] = \mathrm{Tr}(\sigma_W^i \sigma_W^j \rho)$, with W, i and j are the values sampled in step t. Since $\gamma = -\sum_t J_{ij} G_t$, averaging over those choices gives (2.4). Using $|J_{ij}| \leq 1$, by Hoeffding's inequality for any $s > 0$

$$\Pr\left(\left|\gamma - \mathrm{E}[\gamma]\right| > s\right) \leq e^{-\frac{2s^2}{4N}}\,.$$

By choosing N sufficiently large with respect to $\binom{n'}{2}^2 \delta^{-2} \ln(1/\varepsilon)$ we get (2.5). □

Based on Claim 2.10 the proof of Theorem 2.9 follows rather directly. For the completeness, we take $\rho = |\psi\rangle\langle\psi|$ such that $\langle\psi|H|\psi\rangle \leq a$, whose existence is guaranteed by the completeness case of Theorem 2.7. As noted above, this ρ can be prepared efficiently by a QPT prover. Using (2.5) it follows that this prover is accepted with probability at least $1-\varepsilon$. For the soundness, ρ is arbitrary. Using the soundness case of Theorem 2.7 it must be that $\mathrm{Tr}(H\rho) \geq a + \delta$, so that the conclusion follows again from (2.5). □

Even though the verifier's "quantumness" in this protocol is limited — she only needs to hold one qubit at a time — this capability is crucial for the analysis, as it is used to guarantee the "existence" of the state that is being measured: it allows us to meaningfully talk about "the state ρ whose first qubit is the first qubit received by the verifier; whose second qubit is the second qubit received by the verifier; etc.". These qubits are distinct, because the verifier has seen and then discarded them (it would be a different matter if they were returned to the prover). In particular, the fact that a one-qubit computer can be trivially simulated on a classical piece of paper is immaterial to the argument.

With a classical verifier things become substantially more delicate. How can we verify the existence of an n-qubit state with certain properties, while having only access to classical data about the state, data that, for all we know a priori, could have been generated by a simple — classical — laptop? To achieve this we need to find a way for the verifier to establish that the prover holds an n-qubit state, without ever having the ability to directly probe even a single qubit of that state. In the previous lecture we saw a means to achieve this for a single qubit based on the computational hardness of certain functions called "claw-free". In the next lecture we extend that method to introduce a protocol by which the prover can certify the existence of any single-qubit state that is a low-energy eigenstate of a single-qubit Hamiltonian. In the last lecture we combine this extension with the Fizsimons-Morimae protocol to obtain a protocol for delegating quantum computations with a classical client.

Lecture 3

Verifying a single qubit-Hamiltonian

In the previous lecture we introduced the circuit-to-Hamiltonian construction, that given a quantum circuit $\mathcal{C}$ and an input x to it returns a Hamiltonian $H_{\mathcal{C}}$ of the form (2.3) such that the completeness and soundness properties stated in Theorem 2.7 hold. This construction allowed us to reduce the problem of delegating a quantum computation to the problem of deciding if a certain publicly known, explicitly specified exponential-size Hermitian matrix $H_{\mathcal{C}}$ has an eigenvalue below a certain threshold a, or all its eigenvalues are above $b+\delta$ for a δ that is at least inverse polynomial in the number of qubits n on which $H_{\mathcal{C}}$ acts.[1] We then introduced the Fitzsimons-Morimae protocol (Figure 2.2) that is a protocol with one-way communication for verifying this fact.

Our goal in the next two lectures is to combine the Fitzsimons-Morimae protocol with the computational test for a qubit from lecture 1, Section 1.5 to obtain a classical protocol with similar guarantees to the Fitzsimons-Morimae protocol. For this we will develop a test that allows one to verify that a prover "has" a quantum state $|\psi\rangle$ with certain properties (e.g. it satisfies $\langle\psi|H|\psi\rangle \leq a+\delta/2$, i.e. certifies that the outcome of the computation is '1'. Note that even though in principle it is sufficient for the verifier to be convinced that such a $|\psi\rangle$ exists to make the right decision, we will see from the proofs that we can go a little further and give a precise meaning to the notion that the prover 'has' $|\psi\rangle$. This, however, will not be as strong as the claim that the prover 'has n qubits in state $|\psi\rangle$' in the sense that we gave to the phrase 'has n qubits', i.e. we will not quite exhibit $2n$ Pauli operators X_i, Z_i that satisfy all the required relations.

Remark 3.1. In passing to the Hamiltonian model of computation we relaxed our main goal, from obtaining a value $b \in \{0,1\}$ that is distributed as a measurement of the output qubit of the quantum circuit $\mathcal{C}$ in the standard basis to obtaining a value that is 1 whenever this measurement returns 1 with probability larger than $\frac{2}{3}$, and 0 whenever it is less than $\frac{1}{3}$. In particular, we make no requirement for circuits that are "undecided", e.g. return a random bit as output. This is typical to applications in complexity where it is assumed that circuits of interest make a clear-cut decision, 0 or 1; this is the setting discussed in Section 2.1. By tweaking the definition of $H_{\mathcal{C}}$ it is in fact possible to guarantee that any state $|\psi\rangle$ such that $\langle\psi|H_{\mathcal{C}}|\psi\rangle \leq a+\delta/2$ is such that a measurement of the first qubit of $|\psi\rangle$ in the standard basis yields an outcome whose distribution is within total variation distance, say, $\frac{1}{100}$ from a measurement of the output qubit of $\mathcal{C}$. Using this observation the protocol given at the end of this lecture can be adapted to return outcomes that are distributed close to the circuit output distribution, even in cases where the output is not assumed to be biased one way or the other. For simplicity we leave this extension as an exercise to the reader.

[1] In the previous lecture this number of qubits was called n', with n the number of qubits of the circuit $\mathcal{C}$. For the next two lectures, $\mathcal{C}$ disappears and so we re-use n to measure the size of $H_{\mathcal{C}}$.

3.1 A test for a specific single-qubit Hamiltonian

We start with an "easy" case: we show how the computational test for a qubit from lecture 1, protocol $\mathfrak{Q}$, can be cast as a verification protocol for the claim that the Hamiltonian $H = -\sigma_Z$ has a "low" eigenvalue, equal to -1. We go a little further by showing how such an eigenstate can be "extracted" from any successful prover in the protocol.

3.1.1 An explicit isometry

Our main result on the computational qubit test, Theorem 1.9, states that any successful prover in the protocol must "have a qubit". The proof achieves slightly more than that, as it explicitly states what the observables Z in (1.6) and X in (1.7) that define the qubit $(|\psi\rangle, Z, X)$ are. As we saw in Lemma 1.4 the qubit implies the existence of an isometry $V : \mathcal{H} \to \mathbb{C}^2 \otimes \mathcal{H}'$ under which $Z \simeq \sigma_Z$, $X \simeq \sigma_Z$, and $|\psi\rangle \simeq |\psi'\rangle \in \mathbb{C}^2 \otimes \mathcal{H}'$, giving us an identification of the "abstract" qubit $(|\psi\rangle, Z, X)$ with a "concrete" qubit, i.e. the space $\mathbb{C}^2$ and its algebra of operators, of which σ_Z and σ_X form a linear basis.

With our present goal of "extracting" a specific quantum state (a low-energy eigenstate for the Hamiltonian H_C) it is worthwhile making V a little more explicit. Indeed, an important point that we did not emphasize so far is that this identification is not "canonical". The standard proof of Lemma 1.4 involves an application of Jordan's lemma to identify a block structure such that in each block, Z and X act like σ_Z and σ_X respectively. These blocks are obtained by diagonalizing the operator $(X + Z)$. In the case where X and Z anti-commute this operator has only two eigenvalues, $\pm\sqrt{2}$, and the associated eigenspaces are highly degenerate. Any choice of a basis for one of the eigenspaces can be used to specify an isometry V (a basis for the other eigenspace is determined by the first). (That there would be such a degeneracy is easily seen by observing that composing V with any unitary on $\mathcal{H}'$ still gives a valid isometry with the same properties.)

It is, in fact, possible to define a canonical choice for the isometry. This choice has the advantage that it is explicit and from a computational viewpoint leads to a circuit for V that can be constructed from circuits for X and Z. The idea behind the definition is to use the operators X and Z to "teleport" the abstract qubit $(|\psi\rangle, Z, X)$ into a "concrete" qubit $(|\varphi\rangle, \sigma_Z, \sigma_X)$ by means of an EPR pair. This is done in the following proposition.

Proposition 3.2. *Let $(|\psi\rangle, Z, X)$ be a qubit on $\mathcal{H}$. Let $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle) \in \mathbb{C}^2 \otimes \mathbb{C}^2$ be the state of an EPR pair. Let $\mathcal{H}' = \mathbb{C}^2_A \otimes \mathcal{H}$ and $V : \mathcal{H} \to \mathbb{C}^2_Q \otimes \mathcal{H}'$ defined by*

$$\forall |\varphi\rangle \in \mathcal{H}\ , \quad V|\varphi\rangle = \frac{1}{2}\big(\mathrm{Id} \otimes \mathrm{Id}_A \otimes \mathrm{Id}_Q + X \otimes \sigma_X \otimes \mathrm{Id}_Q + Z \otimes \sigma_Z \otimes \mathrm{Id}_Q + XZ \otimes \sigma_X\sigma_Z \otimes \mathrm{Id}_Q\big)|\varphi\rangle|\phi^+\rangle_{AQ}\ , \tag{3.1}$$

where the systems in the range of V are re-ordered so that the first factor $\mathbb{C}^2$ is associated with the second qubit of $|\phi^+\rangle_{AQ}$ in (3.1)*, and $\mathcal{H}'$ consists of the state of the first qubit of $|\phi^+\rangle$, i.e. register A, as well as the part of the state in $\mathcal{H}$. Then V is an isometry and for all $W \in \{X, Z\}$,*

$$VW|\psi\rangle = (\sigma_W \otimes \mathrm{Id})V|\psi\rangle\ . \tag{3.2}$$

Proof. The proof is by direct calculation. First we verify that V is indeed an isometry. This is simply because the four states $\{(\sigma_X(a)\sigma_Z(b) \otimes \mathrm{Id})|\phi^+\rangle,\ a, b \in \{0, 1\}\}$ are orthonormal[2] and X and Z are observables, so that for normalized $|\varphi\rangle$ each of the four terms on the right-hand side of (3.1) has norm exactly 1. Note that

[2]For $a, b \in \{0, 1\}$ we use the notation $\sigma_X(a)$ for σ_X^a and similarly $\sigma_Z(b)$ for σ_Z^b. The motivation for this notation will be seen later when we consider n-qubit Pauli operators.

this does not require any other condition on X, Z than that they are observables (in fact, unitarity suffices). In particular, they do not need to anti-commute. Next we verify (3.2). Take $W = X$. Then

$$\begin{aligned} VX|\psi\rangle &= \frac{1}{2}\big(X\otimes \mathrm{Id}\otimes \mathrm{Id} + \mathrm{Id}\otimes\sigma_X\otimes \mathrm{Id} - XZ\otimes\sigma_Z\otimes \mathrm{Id} - Z\otimes\sigma_X\sigma_Z\otimes \mathrm{Id}\big)|\psi\rangle|\phi^+\rangle \\ &= \frac{1}{2}\big(X\otimes \mathrm{Id}\otimes \mathrm{Id} + \mathrm{Id}\otimes\sigma_X\otimes \mathrm{Id} - XZ\otimes\sigma_Z\otimes \mathrm{Id} - Z\otimes\sigma_X\sigma_Z\otimes \mathrm{Id}\big)|\psi\rangle(\sigma_X\otimes\sigma_X)|\phi^+\rangle \\ &= \frac{1}{2}\big(X\otimes\sigma_X\otimes\sigma_X + \mathrm{Id}\otimes \mathrm{Id}\otimes\sigma_X + XZ\otimes\sigma_X\sigma_Z\otimes\sigma_X + Z\otimes\sigma_Z\otimes\sigma_X\big)|\psi\rangle|\phi^+\rangle \\ &= (\sigma_X\otimes \mathrm{Id})V|\psi\rangle\ , \end{aligned}$$

where for the first line we used that X and Z anti-commute on $|\psi\rangle$, for the second that $\sigma_X\otimes\sigma_X|\phi^+\rangle = |\phi^+\rangle$, for the third that σ_X and σ_Z anti-commute, and for the last we re-ordered terms. □

3.1.2 Extraction of prover's qubit

In the proof of Theorem 1.9 we defined a specific X and Z from the prover's actions and showed that they anti-commute. Moreover, we showed that for a prover that always succeeds in the equation test (case $c = 1$) it must be the case that the state $|\psi\rangle$ of the prover is a $+1$ eigenstate of X, $X|\psi\rangle = |\psi\rangle$. For the definition of $|\psi\rangle \in \mathcal{H}_{\mathsf{X}}\otimes\mathcal{H}_{\mathsf{P}}$, recall that we had assumed that the prover directly measures the qubits in register X on challenge $c = 0$, and applies an arbitrary unitary U before measuring in the Hadamard basis on challenge $c = 1$. This means that after the isometry V, the prover's state $|\psi\rangle$ is mapped to a $+1$ eigenstate of σ_X, i.e. the state $|+\rangle$. The following corollary summarizes this discussion.

Corollary 3.3. *Suppose that a prover succeeds with probability* 1 *in the protocol. Then the isometry V defined from the observables Z in* (1.6) *and X in* (1.7) *sends $|\psi\rangle$ to $|+\rangle_{\mathsf{Q}}|aux\rangle$, for some state $|aux\rangle$ on $\mathcal{H}'$.*

In the context of this lecture we interpret Corollary 3.3 as our first test for a quantum computation in the Hamiltonian-based model: in this test, the verifier is effectively checking that the prover has prepared a $+1$ eigenstate of the Hamiltonian σ_X, or in other words a ground state of $H = -\sigma_X$. While we know that this eigenstate always exists, the analysis of the protocol shows that in some sense the prover has prepared the state. This additional observation allows us to make stronger conclusions from the protocol. For example, just as a measurement of $|+\rangle$ in the computational basis yields an unbiased random bit, we are able to deduce that the value of $b(x)$ with x the prover's answer on challenge $c = 0$ is an unbiased random bit. This makes the protocol potentially useful for cryptographic applications where the generation of certified unbiased randomness serves as a resource.

In the development of our test we were greatly aided by the fact that we *know* what is the ground state of $H = -\sigma_X$, and in particular we know that a Hadamard basis measurement of it yields the outcome 0 (for '+') with probability 1. This "knowledge" was indirectly encapsulated in the test performed for the case $c = 1$, the analysis of which led us to conclude that $X|\psi\rangle = +|\psi\rangle$. But what if we didn't? What if H is a general Hamiltonian of the form (2.3), for which we can't a priori predict measurement outcomes?

3.2 Extracting a qubit: general case

In Section 3.1.1 we made the important observation that the map V in (3.1) is a well-defined isometry for any choice of the two observables X and Z. In particular, this map allows us to make a meaningful definition of a *space* for a qubit, and a *state* for that qubit, associated with *any* prover in protocol $\mathfrak{Q}$, the computational

test for a qubit described in Figure 1.1. This definition does not guarantee that the prover "has a qubit," because it does not say anything about how the prover's observables operate on it. However, it still allows us to define a *candidate* for a single-qubit state on which σ_X and σ_Z measurements can *in principle* be made. The next claim evaluates how outcomes of these measurements when performed on the extracted qubit are distributed as a function of the observables X and Z on the prover's state $|\psi\rangle$.

Claim 3.4. *Let $|\psi\rangle \in \mathcal{H}$ and X, Z observables on $\mathcal{H}$ be arbitrary. Let V be defined as in* (3.1)*. Then the following hold:*

$$\langle\psi|V^\dagger\big(\sigma_Z \otimes \mathrm{Id}\,\big)V|\psi\rangle = \langle\psi|Z|\psi\rangle\ , \tag{3.3}$$

$$\langle\psi|V^\dagger\big(\sigma_X \otimes \mathrm{Id}\,\big)V|\psi\rangle = \frac{1}{2}\big(\langle\psi|X|\psi\rangle - \langle\psi|ZXZ|\psi\rangle\big)\ , \tag{3.4}$$

where the σ_Z and σ_X operators act on the first tensor factor $\mathbb{C}^2$ in the range of V and the identities act on $\mathcal{H}'$.

Recalling the diagram 1.3 introduced to illustrate Lemma 1.4, Claim 3.4 can similarly be illustrated as follows, where E_b denotes the average over b:

$$\begin{array}{ccc} \mathcal{H} & \xrightarrow{\ V\ } & \mathbb{C}^2 \otimes \mathcal{H}' \\ {\scriptstyle Z} \Big\downarrow {\scriptstyle \mathrm{E}_{b\in\{0,1\}}(-1)^b Z^b X Z^b} & & \Big\downarrow {\scriptstyle \sigma_Z\otimes\mathrm{Id}} \\ & & \Big\downarrow {\scriptstyle \sigma_X\otimes\mathrm{Id}} \\ \mathcal{H} & \xrightarrow{\ V\ } & \mathbb{C}^2 \otimes \mathcal{H}' \end{array} \tag{3.5}$$

We emphasize that this diagram is purely illustrative and should be understood exactly in the sense of (3.3) and (3.4); i.e. it does not imply a relation on the operators but only on the expectation values on the state $|\psi\rangle$. Informally, when considering expectation values only the isometry has the effect of applying a Z-twirl to the Hadamard basis observable X.

Proof. Let $W \in \{X, Z\}$. Expanding from the definition of V,

$$\begin{aligned} \langle\psi|V^\dagger\big(\sigma_W \otimes \mathrm{Id}\,\big)V|\psi\rangle &= \frac{1}{4}\sum_{P,Q\in\{I,X,Z,XZ\}} \langle\psi|P^\dagger Q|\psi\rangle \cdot \langle\phi^+|\sigma_P^\dagger\sigma_Q \otimes \sigma_W|\psi\rangle \\ &= \frac{1}{4}\sum_{P,Q:\,\sigma_P^\dagger\sigma_Q=\sigma_W} \langle\psi|P^\dagger Q|\psi\rangle\ , \end{aligned}$$

where for the second line we used that $\langle\phi^+|\sigma_W \otimes \sigma_{W'}|\phi^+\rangle = \delta_{W,W'}$ with δ the Kronecker symbol. In case $W = Z$ the pairs P, Q that appear in the last summation above are $(X, I), (I, X), (XZ, X)$ and (X, XZ). Using $X^2 = \mathrm{Id}$ we obtain (3.3). In case $W = X$ then the summation is over $(Z, I), (I, Z), (XZ, Z)$ and (Z, XZ) and has a minus sign for the last two terms due to $\sigma_X\sigma_Z = -\sigma_Z\sigma_X$. Thus we get (3.4) as well. □

Observe that if X and Z anti-commute then Claim 3.4 gives us the result that we expect: in this case $(|\psi\rangle, Z, X)$ is a qubit so Proposition 3.2 applies and the isometry "intertwines" measurements X and Z on $|\psi\rangle$ with σ_X and σ_Z respectively on the first factor of $V|\psi\rangle$. At the other extreme, if X and Z commute then (3.4) indicates that a measurement in the Hadamard basis of the extracted qubit returns an unbiased random bit. This is expected of a "classical" state, which always leads to uniformly random results in the

Hadamard basis. The lemma in some sense interpolates between these results. Importantly, it allows us to associate a qubit with the state of an arbitrary prover in the protocol, that is such that the distribution of measurements on the extracted qubit can be related to quantities that involve the prover's state and observables in the protocol. For convenience we make this into a definition.

Definition 3.5 (Extracted qubit)**.** Let P be a prover in protocol $\mathfrak{Q}$. Let $|\psi\rangle$ be the state of P after having sent y in the first round of interaction. Let V be defined as in (3.1). Then we call the reduced density of $V|\psi\rangle$ on the first factor $\mathbb{C}^2$, associated with register Q, the *extracted qubit* and denote it by ρ_{Q}.

Lemma 3.6. *Let P be a prover that succeeds with probability 1 in the pre-image test of protocol $\mathfrak{Q}$ and such that the string d returned in the equation test is $d = 0^m$ with probability that is negligibly small in λ. (No other assumption is made on the equation test.) Let ρ be the extracted qubit, as defined in Definition 3.5. Then the following hold:*

- *(Z-measurement:) The outcome of measuring ρ in the computational basis is identically distributed to the bit $(-1)^{b(x)}$ computed from the prover's answer x in case $c = 0$.*
- *(X-measurement:) Under assumption (**F.2**), the outcome of measuring ρ in the Hadamard basis is* computationally indistinguishable *from the bit $(-1)^{d\cdot(x_0+x_1)}$ where d is obtained from the prover in case $c = 1$.*

Remark 3.7 (Computational distinguishability). The statement of the lemma refers to two distributions being computationally indistinguishable. Informally, this means that no computationally efficient procedure can distinguish a sample taken from one distribution from a sample taken from the other. Formally, families of distributions $D = \{D_\lambda\}$ and $D' = \{D'_\lambda\}$ on universes $\{\mathcal{X}_\lambda\}$ are said to be computationally indistinguishable if for any PPT (or QPT for computational indistinguishability against quantum adversaries) procedure $\mathcal{A}$ there is a negligible function μ such that for every λ,

$$\Big|\Pr_{x\leftarrow D_\lambda}\big(\mathcal{A}(1^\lambda,x)=1\big)-\Pr_{x'\leftarrow D'_\lambda}\big(\mathcal{A}(1^\lambda,x')=1\big)\Big| \leq \mu(\lambda)\,.$$

Here, when we refer to computational indistinguishability we will always mean against QPT adversaries. Note that for distributions on a family of universes $\{\mathcal{X}_\lambda\}$ such that $|\mathcal{X}_\lambda|$ grows at most polynomially with λ the notion of computational indistinguishability is equivalent to statistical indistinguishability, i.e. the total variation distance between D_λ and D'_λ goes to 0 as fast as some negligible function. (Showing this formally is a good exercise to practice with the definitions.)

Proof. The first item follows immediately from (3.3) in Claim 3.4 and the definition of Z in (1.6), which guarantees that the bit $(-1)^{b(x)}$ obtained from the prover in case $c = 0$ has expectation precisely $\langle\psi|Z|\psi\rangle$.

To show the second item we assume for contradiction that the two distributions are computationally distinguishable. Since the distributions are over a single bit, as recalled in Remark 3.7 this is equivalent to statistical distinguishability: there must exist a polynomial $q(\lambda)$ such that for infinitely many values of λ,

$$\big|\langle\psi|X|\psi\rangle+\langle\psi|ZXZ|\psi\rangle\big| > \frac{1}{q(\lambda)}\,, \tag{3.6}$$

where recall that the expression on the left should be understood on average over the generation of pk by the verifier and the message y sent by the prover in the first round of interaction. We derive a contradiction with (**F.2**) by constructing an adversary in (1.5). Given λ and pk as input, $\mathcal{A}$ prepares the state $|\psi\rangle$. $\mathcal{A}$ then measures register X in the standard basis to obtain an outcome x. Using the assumption that the prover

succeeds with probability 1 in the pre-image test, $f_{pk}(x) = y$ and the (unnormalized) post-measurement state is exactly $Z_{b(x)}|\psi\rangle$, where as usual $Z_b = (\mathrm{Id} + (-1)^b Z)/2$. Finally, the adversary applies the prover's unitary U and measures in the Hadamard basis to obtain a string d. It returns the pair (x, d). The expected value of $(-1)^{d\cdot(x_0+x_1)}$ under this procedure is

$$\langle\psi|Z_0XZ_0|\psi\rangle + \langle\psi|Z_1XZ_1|\psi\rangle \;=\; \frac{1}{2}\big(\langle\psi|X|\psi\rangle + \langle\psi|ZXZ|\psi\rangle\big)\;,$$

which can be seen by expanding $Z_b = (\mathrm{Id} + (-1)^b Z)/2$ for $b \in \{0,1\}$ and canceling cross-terms. Using (3.6) and the fact that, $\mathcal{A}$ violates (1.5).[3] □

3.3 A single-qubit verification protocol

In the previous section we showed how to identify a "qubit" such that for any prover in the protocol, as long as the prover succeeds in the preimage test then it is possible for the verifier to infer from the prover's answers a bit whose distribution is statistically indistinguishable from outcomes of σ_Z or σ_X measurements on a well-defined quantum state. In order to turn this into a verification protocol for a single-qubit Hamiltonian, we are missing the completeness statement: while in Section 1.5 we saw how a prover could behave in such a way that the extracted qubit is a $|+\rangle$ state, we do not yet know if it is possible to use the protocol for the verification of other single-qubit states. In order for this to work out, we make the following assumption that replaces assumption (**F.4**):

(**F.4'**) For any pk and any y in the range of f_{pk} the two preimages of y take the form (b, x_b) where $b \in \{0,1\}$ and $x_b \in \{0,1\}^{m(\lambda)-1}$. In particular, the function $b : \{0,1\}^m \to \{0,1\}$ returns the first bit of its input.

This assumption is mainly for convenience and holds for most constructions of claw-free functions, including the one that we sketch in the next lecture. Given a 2-to-1 function family that satisfies (**F.4'**) the following lemma shows how a prover can behave in the protocol so that the extracted qubit defined in the previous section is a state $|\varphi\rangle$ of its choice.

Lemma 3.8. *Let $|\varphi\rangle \in \mathbb{C}^2$ be any state. Then there is a way for the prover to behave in protocol $\mathfrak{Q}$ such that the prover is accepted with probability 1 in the preimage test and moreover the extracted qubit satisfies $\rho_Q = |\varphi\rangle\langle\varphi|$.*

Proof. Let $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ for $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$. The prover performs the following steps:

1. Prepare the initial state

$$|\psi^{(0)}\rangle_{\mathsf{BXY}} \;=\; |\varphi\rangle_{\mathsf{B}} \otimes \Big(\frac{1}{2^{m-1}} \sum_{x\in\{0,1\}^{m-1}} |x\rangle_{\mathsf{X}}\Big)|0\rangle_{\mathsf{Y}}\;,$$

   where B is a one-qubit register, X an $(m-1)$ and Y an m-qubit register, for $m = m(\lambda)$.

[3] The end of the proof glosses over a detail: one needs to guarantee that the equation d returned by $\mathcal{A}$ is not 0. While the lemma assumes that this is the case when the equation is measured directly on $|\psi\rangle$, here $\mathcal{A}$ measures after $|\psi\rangle$ has already been measured using the observable Z. To show that the assumption that $d \neq 0^m$ with probability negligibly close to 1 still holds one needs to use the "collapsing" property of f_{pk}, that we will introduce in the next lecture.

2. Upon receipt of the function index pk, coherently evaluate f_{pk} on the input in registers BX, writing the output in register Y to obtain the state

$$|\psi^{(1)}\rangle_{\mathsf{BXY}} = \frac{\alpha}{2^{m-1}} \sum_{x\in\{0,1\}^{m-1}} |0\rangle_{\mathsf{B}}|x\rangle_{\mathsf{X}}|f(0x)\rangle_{\mathsf{Y}} + \frac{\beta}{2^{m-1}} \sum_{x\in\{0,1\}^{m-1}} |1\rangle_{\mathsf{B}}|x\rangle_{\mathsf{X}}|f(1x)\rangle_{\mathsf{Y}} .$$

3. Measure the last register to obtain a y. Let $(0,x_0)$ and $(1,x_1)$ be the two preimages of y under f_{pk}. Then the re-normalized post-measurement state is

$$|\psi^{(2)}\rangle_{\mathsf{BXY}} = \big(\alpha|0\rangle_{\mathsf{B}}|x_0\rangle_{\mathsf{X}} + \beta|1\rangle_{\mathsf{B}}|x_1\rangle_{\mathsf{X}}\big)|y\rangle_{\mathsf{Y}} .$$

4. Upon receipt of challenge c, perform as the honest prover in protocol $\mathfrak{Q}$: if $c = 0$ measure registers BX in the standard basis and return the outcome $x = (b, x_b)$; if $c = 1$ measure in the Hadamard basis and return the outcome d.

This prover always returns a valid preimage in the case of a challenge $c = 0$, so it is accepted with probability 1. Observe that the operator Z associated to this prover is equal to a σ_Z on register B. Regarding the operator X, a simple calculation reveals that the action of X restricted to the span of $|0,x_0\rangle_{\mathsf{BX}}$ and $|1,x_1\rangle_{\mathsf{BX}}$ consists in exchanging these two basis states. Using the explicit form of the isometry V given in (3.1) one can verify that

$$V|\psi^{(2)}\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle_{\mathsf{B}}|x_0\rangle_{\mathsf{X}}|0\rangle_{\mathsf{A}} + |1\rangle_{\mathsf{B}}|x_1\rangle_{\mathsf{X}}|1\rangle_{\mathsf{A}}\big) \otimes \big(\alpha|0\rangle_{\mathsf{Q}} + \beta|1\rangle_{\mathsf{Q}}\big) \otimes |y\rangle_{\mathsf{Y}} ,$$

where AQ are the two registers introduced to hold the EPR pair $|\phi^+\rangle_{\mathsf{AQ}}$ used in the definition of V. Register Q contains the extracted qubit. □

Let $\mathcal{F}$ be a 2-to-1 trapdoor claw-free function family and $\lambda \in \mathbb{N}$ a security parameter. Let $\varepsilon, \delta > 0$ be accuracy parameters. Let $\gamma = 0$ and $N = \frac{C}{\delta^2}\ln(1/\varepsilon)$ for some large constant C. The verifier and prover repeat the following interaction N times.

1. The verifier generates $(pk, td) \leftarrow \textsc{Gen}(1^\lambda)$. It sends pk to the prover.
2. The prover returns $y \in \{0,1\}^m$, where $m = m(\lambda)$.
3. The verifier selects a uniformly random challenge $c \leftarrow_R \{0,1\}$ and sends c to the prover.
4. (a) *(Computational basis, $c = 0$:)* In case $c = 0$ the prover is expected to return an $x \in \{0,1\}^m$. If $f(x) \neq 0$ then the verifier immediately aborts. The verifier sets $a \leftarrow (-1)^{b(x)}$ and $\gamma \leftarrow \gamma - J_Z a$.
   (b) *(Hadamard basis, $c = 1$:)* In case $c = 1$ the prover is expected to return a $d \in \{0,1\}^m$. The verifier uses td to determine the two preimages (x_0, x_1) of y by f_{pk}. She sets $b \leftarrow (-1)^{d\cdot(x_0+x_1)}$ and $\gamma \leftarrow \gamma - J_X b$.

If the verifier has not aborted at any of the steps $c = 0$, she returns the real number $o = \frac{1}{N}\gamma$.

Figure 3.1: Verification protocol for a single-qubit Hamiltonian $H = -\frac{J_X}{2}\sigma_X - \frac{J_Z}{2}\sigma_Z$.

The following proposition summarizes what we have achieved so far, a verification protocol for single-qubit Hamiltonians and a completely classical verifier. (Of course a simpler protocol would be to have the verifier classically do the computation themselves! The point is that this protocol is not too hard to extend to n qubits, as we will see in the next lecture.) The protocol, which combines protocol $\mathfrak{Q}$ with the Fitzsimons-Morimae verification protocol, is summarized in Figure 3.1.

Proposition 3.9. *Let $H = -\frac{J_X}{2}\sigma_X - \frac{J_Z}{2}\sigma_Z$ be a single-qubit Hamiltonian and $\delta, \varepsilon > 0$ accuracy parameters. Then the verification protocol from Figure 3.1 has the following properties:*

1. *(Completeness:) For any single-qubit state $|\varphi\rangle$, there is a QPT prover that is accepted with probability 1 in the protocol and such that the value o returned by the verifier at the end of the protocol satisfies $\mathrm{E}[o] = \langle\varphi|H|\varphi\rangle$.*
2. *(Soundness:) For any QPT prover that is accepted with probability 1 in the protocol, there is a single-qubit state ρ such that the value o returned by the verifier at the end of the protocol satisfies $\mathrm{E}[o] = \mathrm{Tr}(H\rho)$.*

Moreover, with the value of N specified in the protocol in both cases it holds that $\Pr(|o - \mathrm{Tr}(H\rho)| > \delta) \leq \varepsilon$.

Remark 3.10. The assumption that the prover succeeds with probability negligibly close to 1 in the protocol can be relaxed to a constant sufficiently close to 1, where the distance to 1 will affect the distance $|\mathrm{E}[o] - \mathrm{Tr}(H\rho)|$. First we observe that a success probability negligibly close to 1 is sufficient; this can be verified by going through the argument again, and nothing needs to be changed. Second, it is possible to show that any prover with success probability $1 - \kappa$ for some $\kappa \geq 0$ can be transformed to a prover with success probability negligibly close to 1, affecting the distribution of o proportionately to κ. Intuitively, the new prover will test if the value y that the old prover would have returned will lead to success on challenge $c = 0$, in case that the test is actually executed by the verifier. This can be done efficiently by the prover by evaluating the pre-image condition on register X. If this test fails then the new prover simply re-executes the old prover from scratch, until it is certain to achieve success. With probability negligibly close to 1 this iterative procedure will stop in a polynomial number of steps, and using the "pretty-good lemma" it is possible to show that the prover's distribution of outcomes is affected by some $O(\sqrt{\kappa})$ in statistical distance. We omit the details.

Proof. The completeness statement follows from Lemma 3.8. For soundness we use Lemma 3.6. This shows that the expectation of the bit a in step 4.(a) satisfies $\mathrm{E}[a] = \mathrm{Tr}(\sigma_Z\rho)$ where ρ is the extracted qubit. Similarly, the bit b in step 4.(b) satisfies $\mathrm{E}[b] = \mathrm{Tr}(\sigma_Z\rho)$. Since by definition

$$\mathrm{E}[o] \;=\; -\frac{J_Z}{2}\,\mathrm{E}[a] - \frac{J_X}{2}\,\mathrm{E}[b]$$

the proposition follows. □

Lecture 4

Verification for n-qubit Hamiltonians in $XX - ZZ$ form

Moving beyond the case of a single qubit, our goal in this lecture is to generalize the results from Section 3.2 to a procedure for extracting n qubits from a prover, together with a statement that allows us to relate measurements in the standard or Hadamard basis of the extracted qubits to measurements performed by the prover in the protocol. Once this has been put in place we will not be far from a delegation protocol along the lines of the Fitzsimons-Morimae protocol from Section 2.2, but, crucially, with classical verifier and communication.

4.1 Setup

To set the stage we first give a straightforward generalization of the single-qubit verification protocol from Figure 3.1 to the case of n qubits. Recall from Section 2.2.2 that for our purposes it suffices to consider Hamiltonians that take the form (2.3). This form allows us to restrict our attention to a collection of measurement outcomes on an n-qubit state such that all qubits are measured in the same basis, computational or Hadamard. In particular we do not need to consider "mixed" measurements, with some qubits measured in the standard basis and other qubits in the Hadamard basis, because (2.3) does not have mixed terms such as $\sigma_{X,i}\sigma_{Z,j}$. (See Remark 4.1 regarding extensions to the mixed case.) It is then natural to request that the honest prover behaves exactly as in the single-qubit verification protocol, except that each action should be repeated independently for each of the n qubits: in the first phase the verifier sends the information for n functions $f_{pk_1}, \ldots, f_{pk_n}$, the prover executes the encoding procedure from the proof of Lemma 3.8 independently for each of the n qubits of its claimed low-energy eigenstate $|\varphi\rangle$ of $H_{\mathcal{C}}$ and reports the n images $y_1, \ldots, y_n$ obtained; in the second phase the verifier sends a single-bit challenge $c \in \{0,1\}$ and the prover measures all its qubits in the computational or Hadamard basis and returns the outcomes $x_1, \ldots, x_n$ or $d_1, \ldots, d_n$ respectively. Note that the only part that is not repeated is the challenge, which is identical for each of the n concurrent repetitions. The reason that we can restrict ourselves to such challenges is due to the form of $H_{\mathcal{C}}$ from (2.3) and, as we will see, greatly simplifies the analysis. The complete protocol is given in Figure 4.1.

Before proceeding to the analysis of the protocol we examine the question, "Where are the qubits?" For the single-qubit verification protocol our initial intuition came from the qubit computational test from lecture 1, for which we were able to argue that the prover indeed has a qubit $(|\psi\rangle, Z, X)$. For the verification protocol seen in the last lecture we saw that in order to allow verification of other states than the $|+\rangle$ state

Let $\mathcal{F}$ be a 2-to-1 trapdoor claw-free function family and $\lambda \in \mathbb{N}$ a security parameter. Let $\varepsilon, \delta > 0$ be accuracy parameters. Let $\gamma = 0$ and $N = \frac{C}{\delta^2}\binom{n}{2}^2 \ln(1/\varepsilon)$. The verifier and prover repeat the following interaction N times.

1. The verifier selects a pair $i \neq j \in \{1, \ldots, n\}$ and $W \in \{X, Z\}$ uniformly at random.
2. For $\ell = 1, \ldots, n$ the verifier generates $(pk_\ell, td_\ell) \leftarrow \textsc{Gen}(1^\lambda)$. It sends $(pk_1, \ldots, pk_n)$ to the prover.
3. The prover returns $y_1, \ldots, y_n \in \{0,1\}^m$, where $m = m(\lambda)$.
4. The verifier selects a uniformly random challenge $c \leftarrow_R \{0,1\}$ and sends c to the prover.
5. (a) *(Computational basis, $c = 0$:)* In case $c = 0$ the prover is expected to return $x_1, \ldots, x_n \in \{0,1\}^m$. If $f_{pk_\ell}(x_\ell) \neq 0$ for any ℓ then the verifier immediately aborts. The verifier sets

$$\gamma \leftarrow \gamma - J_{ij}\,(-1)^{b(x_i)}(-1)^{b(x_j)}\,.$$

   (b) *(Hadamard basis, $c = 1$:)* In case $c = 1$ the prover is expected to return $d_1, \ldots, d_n \in \{0,1\}^m$. The verifier uses td_i and td_j to determine the preimages $(x_{i,0}, x_{i,1})$ of y_i by f_{pk_i} and $(x_{j,0}, x_{j,1})$ of y_j by f_{pk_j} respectively. She sets

$$\gamma \leftarrow \gamma - J_{ij}\,(-1)^{d_i \cdot (x_{i,0} + x_{i,1})}(-1)^{d_j \cdot (x_{j,0} + x_{j,1})}\,.$$

If the verifier has not aborted at any of the steps $c = 0$, she returns the real number $o = \frac{1}{N}\binom{n}{2}\gamma$.

Figure 4.1: Verification protocol $\mathfrak{V}_n$ for an n-qubit Hamiltonian $H_{\mathcal{C}} = -\sum_{i,j} \frac{J_{ij}}{2}(\sigma_{X,i}\sigma_{X,j} + \sigma_{Z,i}\sigma_{Z,j})$.

we had to remove some of the tests done by the verifier (specifically, the equation check) and that due to this we were no longer able to guarantee a qubit in the sense of Definition 1.3. Nevertheless we were able to get around this by defining an abstract *extracted qubit* that did not directly correspond to the prover's observables but was still such that measurement outcomes on the extracted qubit could be shown to have a distribution that is negligibly close to outcomes obtained from the prover in the actual protocol (Lemma 3.6).

For the case of demonstrating n qubits a priori one would have to show that the prover has a state $|\psi\rangle$ and two families of observables $\{X(a) : a \in \{0,1\}^n\}$ and $\{Z(b) : b \in \{0,1\}^n\}$ that satisfy the Pauli commutation and anti-commutation relations when they act on $|\psi\rangle$. Indeed, a straightforward generalization of Lemma 1.4 then guarantees the existence of a suitable isometry with the space of n actual qubits. Showing this is challenging; luckily, for our purposes it is also not necessary. Indeed, just as in the single-qubit case it is worth emphasizing that in the context of verification we do not need to guarantee that the prover has a certain quantum state, nor that it is able to perform certain measurements on it. The only real requirement is that a state $|\varphi\rangle$ *exists* such that $\langle\varphi|H_{\mathcal{C}}|\varphi\rangle \leq a$. Thus, as we did in the analysis of the single-qubit verification protocol we will first introduce a abstract *extracted n qubit* defined from the prover's state and actions in the protocol but that also include additional ingredients that make it at first unclear how they relate to the prover itself. The definition of the extracted qubits is given in Section 4.2. Once this has been defined we will perform the second, crucial step, which is to relate the distribution of measurement outcomes on the extracted qubits to quantities that are directly observable in the protocol. This is done in Section 4.3. Finally in Section 4.4 we put everything together and show the completeness and soundness properties of the verification protocol given in Figure 4.1. In addition in Section 4.5 we will sketch a construction of a function family based on the Learning With Errors (LWE) problem that (approximately) satisfies all required assumptions and can thus be used to instantiate the protocol.

4.2 The n extracted qubits

4.2.1 Modeling the prover

We start by introducing notation that allows us to model an arbitrary prover in the protocol. Similarly to how we modeled the prover for the analysis of the computational qubit test in Section 1.5, a prover in the n-qubit verification protocol from Figure 4.1 can be represented using the following objects:

1. A state $|\psi\rangle$, that may depend on $pk_1, \ldots, pk_n$ and $y_1, \ldots, y_n$, such that $|\psi\rangle \in \mathcal{H}_{\mathsf{X}_1} \otimes \cdots \mathcal{H}_{\mathsf{X}_n} \otimes \mathcal{H}_{\mathsf{P}}$ with each space $\mathcal{H}_{\mathsf{X}_i}$ isomorphic to $(\mathbb{C}^2)^{\otimes m}$. The state $|\psi\rangle$ represents the state of the prover and the message registers at the end of step 3 in the protocol.

2. For the case $c = 0$, the prover directly measures all the X registers in the standard basis to obtain $x_1, \ldots, x_n$ that it returns to the verifier. For a string $a \in \{0,1\}^n$ we let

$$Z(a) = \sum_{x_1,\ldots,x_n} (-1)^{a_1 \cdot b_1(x_1)} \cdots (-1)^{a_n \cdot b_n(x_n)} |x_1\rangle\langle x_1| \otimes \cdots |x_n\rangle\langle x_n| \,, \tag{4.1}$$

   where the functions b_i are not necessarily all equal since they may depend on pk_i. This is analogous to (1.6).

3. For the case $c = 1$, the prover applies an arbitrary unitary U followed by a measurement of the qubits

in X in the Hadamard basis to obtain $d_1, \ldots, d_n$. For a string $b \in \{0,1\}^n$ we let

$$X(b) = \sum_{d_1,\ldots,d_n} (-1)^{b_1(d_1\cdot(x_{1,0}+x_{1,1}))} \cdots (-1)^{b_n(d_n\cdot(x_{n,0}+x_{n,1}))} \cdot U^\dagger \big(H_{\mathsf{X}}^{\otimes nm} \otimes \mathrm{Id}_{\mathsf{P}}\big)^\dagger \big(|d_1,\ldots,d_n\rangle\langle d_1,\ldots,d_n|_{\mathsf{X}} \otimes \mathrm{Id}_{\mathsf{P}}\big)\big(H_{\mathsf{X}}^{\otimes nm} \otimes \mathrm{Id}_{\mathsf{P}}\big) U \,. \quad (4.2)$$

Remark 4.1. Note that the fact that the protocol only has two different challenges, $c = 0$ and $c = 1$, allows us to have a simple description for all $Z(a)$ and all $X(b)$ observables that involves only one "adversarial" unitary U. If we had to design a protocol that allows more general Hamiltonians with mixed terms of the form $\sigma_{X,i}\sigma_{Z,j}$ we would need to consider more challenges, and this would require a more complex analysis. This is done in [Mah18].

4.2.2 The isometry V

Next we define the n-qubit isometry V, and the extracted qubits.

Claim 4.2. *Let $|\psi\rangle \in \mathcal{H}$ and for every $a, b \in \{0,1\}^n$, $X(a)$ and $Z(b)$ observables on $\mathcal{H}$ such that all $X(a)$ (resp. all $Z(b)$) mutually commute and moreover $X(a)X(a') = X(a+a')$ for any $a, a' \in \{0,1\}^n$. Let $V : \mathcal{H} \to \mathcal{H}_{\mathsf{Q}} \otimes \mathcal{H}_{\mathsf{A}} \otimes \mathcal{H}'$ where each of $\mathcal{H}_{\mathsf{Q}}$ and $\mathcal{H}_{\mathsf{A}}$ is $(\mathbb{C}^2)^{\otimes n}$ and $\mathcal{H}' \simeq \mathcal{H}$ be defined for all $|\varphi\rangle \in \mathcal{H}$ as*

$$V|\varphi\rangle \;=\; \Big(\frac{1}{2^n}\sum_{a,b} \mathrm{Id} \otimes \sigma_X(a)\sigma_Z(b) \otimes X(a)Z(b)\Big)|\phi^+\rangle^{\otimes n}|\varphi\rangle \;, \quad (4.3)$$

where each EPR pair $|\phi^+\rangle$ has one qubit in register Q and the other in register A and the σ_X and σ_Z operators act on register A. Then V is an isometry.

The proof of the claim is immediate and only uses that the family of states

$$\big\{(\sigma_X(a)\sigma_Z(b) \otimes \mathrm{Id})|\phi^+\rangle^{\otimes n} : \, a, b \in \{0,1\}^n\big\}$$

is orthonormal. Similarly to Definition 3.5 we can now define the n extracted qubits.

Definition 4.3 (Extracted qubits)**.** Let P be a prover in the verification protocol $\mathfrak{V}_n$ described in Figure 4.1. Let $|\psi\rangle$ be the state of P after having sent $y_1, \ldots, y_n$ at step 3 of the t-th iteration, for some $t \in \{1, \ldots, N\}$. Let V be defined in (4.3). Then we call the reduced density of $V|\psi\rangle$ on register Q the *extracted qubits* (implicitly, at iteration t) and denote them by $\rho_{\mathsf{Q}_1\cdots\mathsf{Q}_n}$.

4.3 Measurements on the extracted qubits

We start with the following analogue to Claim 3.4, which gives an explicit formula for the distribution of measurements in the standard or Hadamard basis on the n extracted qubits as a function of the prover's state and observables.

Claim 4.4. *The following hold for any prover, with ρ the n extracted qubits at any iteration (Definition 4.3):*

$$\forall b \in \{0,1\}^n \,, \qquad \mathrm{Tr}\big(\sigma_Z(b)\rho\big) = \langle\psi|Z(b)|\psi\rangle \,, \quad (4.4)$$

$$\forall a \in \{0,1\}^n \,, \qquad Tr\big(\sigma_X(a)\rho\big) = \frac{1}{2^n}\sum_b (-1)^{a\cdot b}\langle\psi|Z(b)X(a)Z(b)|\psi\rangle \,. \quad (4.5)$$

The claim can be illustrated using the following generalization of (3.5)

$$\begin{array}{ccc} \mathcal{H} & \xrightarrow{V} & \mathbb{C}^2 \otimes \mathcal{H}' \\ {\scriptstyle Z(b)} \big\downarrow & & \big\downarrow {\scriptstyle \sigma_Z(b)\otimes \mathrm{Id}} \\ {\scriptstyle \mathrm{E}_{b\in\{0,1\}}(-1)^{b\cdot a} Z(b)X(a)Z(b)} \big\downarrow & & \big\downarrow {\scriptstyle \sigma_X(a)\otimes \mathrm{Id}} \\ \mathcal{H} & \xrightarrow{V} & \mathbb{C}^2 \otimes \mathcal{H}' \end{array} \tag{4.6}$$

Proof. Eq. (4.4) is immediate using that $X(a)$ are observables and $\langle \phi^+|^{\otimes n} \sigma_X(a')\sigma_Z(b') \otimes \sigma_Z(b) |\phi^+\rangle^{\otimes n}$ is zero unless $a' = 0$ and $b = b'$. Eq. (4.5) is shown similarly by direct calculation, using $X(a')X(a'') = X(a' + a'')$ and $\sigma_Z(b)\sigma_X(a)\sigma_Z(b) = (-1)^{a\cdot b}\sigma_X(a)$. □

The next lemma is the key lemma. It argues that for computationally bounded provers, the quantity on the right-hand side of (4.5) is close to the simpler quantity $\langle \psi | X(a) | \psi \rangle$, that in particular can be inferred in the protocol from the prover's outcomes y_i and d_i (for those i such that $a_i = 1$). Before we can state the lemma we need to introduce one last assumption on the function family $\mathcal{F}$. Intuitively, this assumption is a natural quantum analogue of the classical property of collision resistance, but is stronger than it.

(**F.5**) Consider the following abstract game between an arbitrary "adversary" (think prover) and a trusted (quantum) "challenger" (think verifier). First, the adversary is provided a label pk (generated at random by the challenger) and required to prepare an arbitrary state of the form $|\phi\rangle = \sum_x \alpha_x |x\rangle$, where x ranges over the domain of f_{pk}. (In general the adversary may keep an additional register entangled with this state. For ease of notation we do not consider such entanglement in this description.) The adversary hands the state $|\phi\rangle$ over to the challenger, who evaluates f_{pk} in superposition on $|\phi\rangle$ and measures the image register, obtaining a y in the range of f_{pk} and the (suitably re-normalized) post-measurement state $|\phi'\rangle = \sum_{x:f_{pk}(x)=y} \alpha_x |x\rangle$. The challenger then returns to the adversary the string c together with *either* the state $|\phi'\rangle$ *or* the probabilistic mixture $\sum_{x:f(x)=c} |\alpha_x|^2 |x\rangle\langle x|$ obtained by measuring the same state $|\phi'\rangle$ in the computational basis (and throwing away the outcome). The adversary wins if it correctly guesses which is the case. Assumption (**F.5**) on the function family $\mathcal{F}$ states that for any QPT adversary $\mathcal{A}$ there is a negligible function μ such that for any λ, $\mathcal{A}$ succeeds in this game with probability that deviates from $\frac{1}{2}$ by at most $\mu(\lambda)$.

Remark 4.5. Assumption (**F.5**) is referred to as the "collapsing" property for the function family $\mathcal{F}$. This property was introduced by Unruh as a strengthening of the classical property of collision resistance required for his work on the security of commitment protocols that are computationally binding against quantum adversaries [Unr16]. The reason that this assumption implies collision resistance is that, if the function were not collision resistant, the adversary could identify a colliding pair (x_0, x_1) and submit $|\phi\rangle = \frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$ to the challenger. It could then measure the challenger's response in a basis containing the two states $\frac{1}{\sqrt{2}}(|x_0\rangle \pm |x_1\rangle)$ and guess that, in case the "$-$" outcome is obtained, the challenger must have measured; in the other case, the adversary guesses at random.

Note that assumption (**F.2**) *also* trivially implies collision resistance, since the ability to identify a claw allows one to generate arbitrary equations in it. It is possible to show that both (**F.2**) and (**F.5**) are strictly stronger than collision resistance. It is likely that the two assumptions are incomparable, but I have not tried to show this explicitly.

We can now state and prove the key lemma.

Lemma 4.6. *Let P be a prover that succeeds with probability 1 in protocol $\mathfrak{V}_n$. Let ρ be the n extracted qubits, as defined in Definition 4.3 (for any iteration). Then the following hold for any $i \neq j \in \{1,\ldots,n\}$:*

- *(Z-measurement:) The outcome of measuring qubits i and j of ρ in the computational basis is identically distributed to the bits $b(x_i)$ and $b(x_j)$ obtained from the prover in case $c = 0$.*
- *(X-measurement:) Under assumptions (**F.2**) and (**F.5**) the outcome of measuring qubits i and j of ρ in the Hadamard basis is* computationally indistinguishable *from the pair of bits $d_i \cdot (x_{i,0} + x_{i,1})$ and $d_j \cdot (x_{j,0} + x_{j,1})$ where d_i and d_j are obtained from the prover in case $c = 1$.*

As already noted in the previous lecture, for distributions on two bits the notions of computational and statistical indistinguishability are essentially equivalent. The lemma generalizes to the joint distribution of any number of bits, and in this case it is only the weaker computational indistiguishability that is obtained. For simplicity we restrict ourselves to proving the lemma for the setting of two bis only.

Proof. For the case of a measurement in the computational basis the lemma follows directly from (4.4) in Claim 4.4 and the definition of the extracted qubits. For the case of a measurement in the Hadamard basis we proceed in two steps.

In the first step we show that for any $b \in \{0,1\}^n$ the states $|\psi\rangle$ and $Z(b)|\psi\rangle$ are computationally indistinguishable. We show this by performing a reduction to an adversary that breaks assumption (**F.5**). Fix any $i \in \{1,\ldots n\}$ and suppose for contradiction that there exists an efficient observable R such that

$$\left|\langle\psi|R|\psi\rangle - \langle\psi|Z(e_i)RZ(e_i)|\psi\rangle\right| \;>\; \frac{1}{q(\lambda)}\,,$$

for some polynomial q and where the left-hand side should be understood on expectation over the creation of a state $|\psi\rangle$ according to the first three steps of protocol $\mathfrak{V}_n$. Let i be a position in which $b_i \neq 0$. Our goal is to reach a contradiction with (**F.5**). Towards this we construct an adversary $\mathcal{A}$ to the collapsing game that underlies assumption (**F.5**). Upon input pk, $\mathcal{A}$ creates the state $|\psi\rangle$ and returns to the challenger only the m-qubit register X_i. Note that the first part of the challenger's actions in the game does not change $|\psi\rangle$, since the prover has already collapsed it to a pair of preimages. The two cases correspond to the challenger returning either the mixed state $\sum_{b\in\{0,1\}} Z_{b,i}|\psi\rangle\langle\psi|Z_{b,i}$ or $|\psi\rangle\langle\psi|$, where $Z_{b,i} = (\mathrm{Id} + (-1)^b Z(e_i))/2$. The adversary $\mathcal{A}$ measures R and returns the outcome. The advantage of $\mathcal{A}$ in distinguishing the two cases is

$$\left|\langle\psi|R|\psi\rangle - \sum_b \langle\psi|Z_{b,i}RZ_{b,i}|\psi\rangle\right| \;=\; \frac{1}{2}\left|\langle\psi|R|\psi\rangle - \langle\psi|Z(e_i)RZ(e_i)|\psi\rangle\right|\,,$$

where the equality follows by definition of $Z_{b,i}$. Since by (**F.5**) this advantage should be negligible, we deduce that for every $i \in \{1,\ldots,n\}$ and every efficient observable R it must be that

$$\langle\psi|R|\psi\rangle \approx \langle\psi|Z(e_i)RZ(e_i)|\psi\rangle\,. \tag{4.7}$$

Since for any b the observable $Z(b)RZ(b)$ itself is efficient, applying (4.7) n times (with different choices of R) we deduce that for any b and efficient R,

$$\begin{aligned}\langle\psi|R|\psi\rangle &\approx \langle\psi|Z(b_1e_1)RZ(b_1e_1)|\psi\rangle\\ &\approx \langle\psi|Z(b_1e_1+b_2e_2)RZ(b_1e_1+b_2e_2)|\psi\rangle\\ &\approx \cdots\\ &\approx \langle\psi|Z(b)RZ(b)|\psi\rangle\,.\end{aligned}$$

We now extend the preceding reasoning to show that for any a of the form $a = a_i e_i + a_j e_j$ with e_i, e_j the canonical basis vectors and $a_i, a_j \in \{0,1\}$,

$$\Big| \frac{1}{2^n} \sum_b (-1)^{a\cdot b} \langle \psi | Z(b) X(a) Z(b) | \psi \rangle \\ - \frac{1}{4} \sum_{b_i, b_j \in \{0,1\}} (-1)^{a_i b_i + a_j b_j} \langle \psi | Z(b_i e_i + b_j e_j) X(a) Z(b_i e_i + b_j e_j) | \psi \rangle \Big| \leq \mu(\lambda) \,, \tag{4.8}$$

for some negligible function ν. Supposing this were not the case, by the triangle inequality and an averaging argument there must exist a b such that

$$\big| \langle \psi | Z(b) X(a) Z(b) | \psi \rangle - \langle \psi | Z(b_i e_i + b_j e_j) X(a) Z(b_i e_i + b_j e_j) | \psi \rangle \big| > \frac{1}{q(\lambda)} \,,$$

for some polynomial q. This leads to a contradiction with (**F.5**) using the same reasoning as before, because from the point of view of the statement of (**F.5**) for qubits not in positions i and j, the observable $X(a)$ is efficient, as its computation only requires trapdoors td_i and td_j.

To obtain the second part of the claim it remains to handle the $Z(e_i)$ and $Z(e_j)$ operators. For the positions i and j the associated trapdoor information *is* used in the computation of $X(a)$, so the preceding reasoning cannot be applied. Instead, we proceed similarly to the proof of Lemma 3.6, by reduction to the adaptive hardcore bit property, assumption (**F.2**). Note that if $a_i = 0$ or $a_j = 0$ then our task is exactly the task handled in Lemma 3.6. So assume $a_i = a_j = 1$. We perform a reduction to Lemma 3.6 via a simple hybrid argument. Suppose for the sake of contradiction that

$$\Big| \langle \psi | X(a) | \psi \rangle - \frac{1}{4} \sum_{b_i, b_j \in \{0,1\}} (-1)^{a_i b_i} \langle \psi | Z(b_i e_i + b_j e_j) X(a) Z(b_i e_i + b_j e_j) | \psi \rangle \Big| > \frac{1}{q(\lambda)} \,.$$

Then by the triangle inequality and averaging it must be that either

$$\Big| \langle \psi | X(a) | \psi \rangle - \frac{1}{2} \sum_{b_i \in \{0,1\}} (-1)^{a_i b_i + a_j b_j} \langle \psi | Z(b_i e_i) X(a) Z(b_i e_i) | \psi \rangle \Big| > \frac{1}{q(\lambda)} \,,$$

or

$$\Big| \langle \psi | Z(b_j) X(a) Z(b_j) | \psi \rangle - \frac{1}{2} \sum_{b_i \in \{0,1\}} (-1)^{a_i b_i} \langle \psi | Z(b_i e_i + e_j) X(a) Z(b_i e_i + e_j) | \psi \rangle \Big| > \frac{1}{q(\lambda)} \,.$$

The first case is ruled out directly by Lemma 3.6. The second case is ruled out by the same lemma, simply considering a prover that creates the state $Z(b_j)|\psi\rangle$ instead of $|\psi\rangle$ at the 3rd step (i.e. the step where $|\psi\rangle$ is defined). □

4.4 An n-qubit verification protocol

The following theorem is the main result of the past four lectures. It generalizes Proposition 3.9 to the case of an n-qubit Hamiltonian.

Theorem 4.7. *Let $\mathcal{F}$ be a function family satisfying (**F.1**), (**F.2**), (**F.3**), (**F.4'**) and (**F.5**). Let $H_{\mathcal{C}}$ be an n-qubit Hamiltonian of the form* (2.3) *and $\delta, \varepsilon > 0$ accuracy parameters. Then the verification protocol from Figure 4.1 has the following properties:*

1. *(Completeness:) For any n-qubit state $|\varphi\rangle$, there is a QPT prover that is accepted with probability* 1 *in the protocol and such that the value o returned by the verifier at the end of the protocol satisfies* $\mathrm{E}[o] = \langle\varphi|H|\varphi\rangle$.

2. *(Soundness:) For any QPT prover that is accepted with probability* 1 *in the protocol, there is an n-qubit state ρ such that the value o returned by the verifier at the end of the protocol satisfies* $\Pr(|o - \mathrm{Tr}(H\rho)| > \delta) \leq \varepsilon$.

Remark 4.8. The protocol in Figure 4.1, as the one in Figure 3.1, involves N repetitions of an elementary 4-message procedure. It is possible to parallelize the protocol to a single repetition in which the prover is asked to perform measurements on all N qubits of a ground state of $H_{\mathcal{C}}$. This however requires more work, because in the parallelized protocol the verifier needs to request "mixed" measurements from the prover; see Remark 4.1.

Remark 4.9. We pause to insist on how amazing Theorem 4.7 is. Due to Kitaev's circuit-to-Hamiltonian construction (Section 2.2.1) it is known that, under the widely believed assumption that QMA $\neq$ QCMA (where QCMA is the class of languages that admit classical proofs verifiable by QPT verifiers), there exist families of Hamilitonians of the form $H_{\mathcal{C}}$ such that any sufficiently low-energy eigenstate of $H_{\mathcal{C}}$ cannot have a simple classical description; in particular, there is no small quantum circuit to prepare such eigenstates, they must have high entanglement, etc. Yet Theorem 4.7 states that through an efficient classical interaction with a device that has the ability to prepare such states it is possible to *efficiently* verify their *existence*. There are two ways in which one might aim to strengthen that statement. First, in the spirit of "proofs of knowledge" we might aim to show that the prover *has* such a state, and not only that it *exists*. Showing this requires a formalization of the notion of the prover "having" a certain quantum state, but it can be done without any modification to the protocol itself; see [VZ20]. Second, in the spirit of our "test for a qubit" we might aim to show that the prover *has n qubits*. This we do not know how to show in the computational setting: it is an open question. (See the full notes at http://users.cms.caltech.edu/ vidick/teaching/fsmp/fsmp.pdf for how to achieve this broader goal under a different assumption, that of spatial assumption between two provers.)

Remark 4.10. The assumption that the prover succeeds with probability 1 that is made in the soundness statement is not difficult to relax; see Remark 3.10.

Proof. The completeness statement is entirely analogous to the same statement for Proposition 3.9. In slightly more detail, at each of the N iterations the honest prover prepares a fresh copy of the state $|\varphi\rangle$ and then applies the procedure described in the proof of Lemma 3.8 independently to each of the n qubits of $|\varphi\rangle$, using the key pk_i for the i-th qubit and obtaining an outcome y_i. For each qubit the post-measurement state is in an m-qubit register $\mathcal{X}_i$ that the prover measures in the standard basis in case of challenge $c = 0$, and Hadamard basis in case $c = 1$. It can then be verified by direct calculation that in case $c = 0$ for any pair $i \neq j$ the parity $(-1)^{b(x_i)+b(x_j)}$ is distributed as a measurement of $\sigma_Z(e_i + e_j)$ on $|\varphi\rangle$, and similarly in case $c = 1$ for any pair $i \neq j$ the parity $(-1)^{d_i\cdot(x_{i,0}+x_{i,1})+d_j\cdot(x_{j,0}+x_{j,1})}$ is distributed as a measurement of $\sigma_X(e_i + e_j)$ on $|\varphi\rangle$.

For soundness we use Lemma 4.6. The lemma shows that for any iteration $t = 1, \ldots, N$ in the protocol we can define a state ρ_t such that averaging over the verifier's choice of qubits i and j it holds that, whenever $c = 0$ then

$$\mathrm{E}\left[J_{ij}\,(-1)^{b(x_i)}(-1)^{b(x_j)}\right] \;=\; J_{ij}\,\mathrm{Tr}\big(\sigma_{Z,i}\sigma_{Z,j}\rho_t\big)\;.$$

and whenever $c = 1$ then

$$\mathrm{E}\left[J_{ij}\,(-1)^{d_i\cdot(x_{i,0}+x_{i,1})}(-1)^{d_j\cdot(x_{j,0}+x_{j,1})}\right] \;\approx\; J_{ij}\,\mathrm{Tr}\big(\sigma_{X,i}\sigma_{X,j}\rho_t\big)\;,$$

where the approximation is up to some negligible quantity in λ. Averaging these two quantities we see that on average over all the rounds,

$$\begin{aligned}\mathrm{E}[o] &\approx \frac{1}{N}\sum_{t=1}^{N}\sum_{i\neq j}\Big(-\frac{1}{2}J_{ij}\,\mathrm{Tr}\big(\sigma_{Z,i}\sigma_{Z,j}\rho_t\big)-\frac{1}{2}J_{ij}\,\mathrm{Tr}\big(\sigma_{X,i}\sigma_{X,j}\rho_t\big)\Big)\\ &= \frac{1}{N}\sum_{t=1}^{N}\mathrm{Tr}\big(H_{\mathcal{C}}\rho_t\big)\\ &= \mathrm{Tr}\big(H_{\mathcal{C}}\rho\big)\ ,\end{aligned}$$

where we defined $\rho = \frac{1}{N}\sum_t \rho_t$. The more quantitative statement given in the soundness part of the theorem follows directly by using a martingale concentration argument, provided the constant C in the definition of N is chosen large enough. □

4.5 Construction of a claw-free function family $\mathcal{F}$

The presentation of this section is adapted from [Vid20].

In Section 1.5 we have identified four assumptions (we added a fifth one in Section 4.3) on a family of functions $\{f_{pk(\lambda)} : \{0,1\}^{m(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$, such that the five assumptions together are sufficient for the resulting delegated computation protocol to be sound. Can the five assumptions be simultaneously satisfied? Strictly speaking, we do not know the answer. In this section we sketch a construction that *nearly* satisfies the assumptions. The construction appears in [BCM$^+$18], and a mild modification of it is used in Mahadev's protocol. Even though the desired assumptions will not all be strictly satisfied by the construction,[1] it is possible to verify that the protocol itself remains sound.

4.5.1 The LWE problem

Our starting point is the *Learning with Errors* problem, introduced by Regev [Reg09]. The hardness of this problem has become a widely used computational assumption in cryptography, for at least three reasons. The first is that it is very versatile, allowing the implementation of advanced primitives such as fully homomorphic encryption [Gen09, BV14], attribute-based encryption [GVW15], program obfuscation [WZ17, GKW17], traitor tracing [GKW18], and many others. The second is that the assumption can be reduced to the hardness of *worst-case* computational problems on lattices: an efficient procedure that breaks the LWE assumption *on average* can be used to solve the closest vector problem in (almost) any lattice. The third reason, that is most relevant to the use of the LWE assumption made here, is that in contrast to the RSA assumption on the hardness of factoring or the discrete logarithm problem so far it is believed that the LWE problem may be hard for quantum computers, so that cryptographic schemes based on it remain (to the best of published knowledge) secure against quantum attacks.

The LWE assumption comes in multiple flavors, all roughly equivalent. Here we formulate the *decisional LWE* assumption on the difficulty of distinguishing samples from two distributions. To state the problem, fix a size parameter $n \geq 1$, an integer modulus $q \geq 2$, a number of equations $m \geq n\log q$, and an

[1] In particular, we construct functions from $\mathbb{Z}_q^m$ to Z_q^m for some q that is required to be large and may not necessarily be chosen even. The definition of assumption (**F.2**) considers equations modulo 2, and this is naturally tailored to the capabilities of a quantum prover, for whom it is possible to generate such equations by measuring in the Hadamard basis. The family of functions constructed in this section can be shown to possess the hardcore bit property over $\mathbb{Z}_q$, but proving it over $\mathbb{Z}_2$ requires more work.

error distribution χ over $\mathbb{Z}_q$.[2] Given χ, write χ^m for the distribution over $\mathbb{Z}_q^m$ that is obtained by sampling each entry of a vector independently according to χ. The decisional LWE assumption is the following.

> *(Decisional LWE, informal)* Let A be a uniformly random matrix in $\mathbb{Z}_q^{m\times n}$, s a uniformly random vector in $\{0,1\}^n$, e a random vector in $\mathbb{Z}_q^m$ drawn from χ^m, and r a uniformly random vector in $\mathbb{Z}_q^m$. Then no classical or quantum probabilistic polynomial-time procedure can distinguish $(A, As+e)$ from (A, r).

Note that the distribution of $(A, As+e)$ and the distribution of (A, r) are in general very far from each other: provided m is sufficiently larger than n a random vector r will not lie in the column span of A, nor even be close to it. What the (decisional) LWE assumption asserts is that, even though in principle these distributions are far from each other, it is computationally difficult, given a sample from the one or the other, to tell which is the case. Note that without the error vector e the task would be easy: given (A, y), solve for $As = y$ and check whether the solution has coefficients in $\{0,1\}$. The LWE assumption is that the inclusion of e makes the task substantially more arduous. In particular, it is well-known that Gaussian elimination is very sensitive to errors, which rules out the most natural approach.

The definition we gave is informal because we have not specified how the parameters n, m and q should be chosen as a function of the security parameter λ, and we have not specified the distribution χ. In general one can make the decisional LWE assumption for any choice of these parameters—but for some choices the assumption will be invalidated by existing algorithms. We comment on come choices of parameters that are made in cryptography. The integer n should generally be thought of as commensurate with the security parameter λ, i.e. $n = \Theta(\lambda)$. The modulus q should be at least polynomial in n, but can be as large as exponential; this will be the case in our construction. The error distribution χ can be chosen in multiple ways. A common choice is to set χ a discretized centered Gaussian distribution with variance αq, for some small parameter α (typically chosen as an inverse polynomial function of n); this is generally denoted $D_{\mathbb{Z}_q,\alpha q}$. For more details on LWE and its applications, we refer to the survey [P+16].

4.5.2 Construction

To specify the function family $\mathcal{F}$ we first describe how public and private parameters for the function are chosen. Let λ be the security parameter (i.e. the number 2^λ is thought of as an estimate of the time required to break assumptions such as (**F.2**)).

First, integers n, m and a modulus q are chosen such that $n = \Omega(\lambda)$, $q \geq 2$ is a prime, and $m = \Omega(n \log q)$. Then, a matrix $A \in \mathbb{Z}_q^{m\times n}$ is sampled at random, together with a "trapdoor" in the form of a matrix $R \in \mathbb{Z}_q^{\ell\times m}$, where $n \leq \ell \leq m$ is a parameter. The sampling procedure has the property that the distribution of A is statistically close to uniform, and R is such that $G = RA \in \mathbb{Z}_q^{\ell\times n}$ is a "nice" matrix, in the sense that given $b = Gs + e$, for any $s \in \mathbb{Z}_q^n$ and e small enough, it is computationally easy to recover s.[3] That such a sampling procedure would exist and be efficiently implementable is non-trivial, and relies on the underlying lattice structure given by the columns of A; see [MP12]. Finally, a uniformly random $s \in \{0,1\}^n$, and a random $e \in \mathbb{Z}_q^m$ distributed according to $D_{\mathbb{Z}_q,\alpha q}$ with α of order $1/(\sqrt{mn\log q})$,[4]

[2] The use of the parameters n, m and q is local to this section. In particular, the m that specifies the domain and range of the function f_{pk} is not identical to the m here; see below.

[3] One can think of G as a matrix whose rows are almost orthonormal, so that Gaussian elimination on G induces only small propagation of the errors.

[4] The precise choice of α is delicate, and the parameters given here should only be treated as indicative; we refer to [BCM+18, Section 8] for the right setting of parameters.

are sampled. The public information is $pk = (A, z = As + e)$. The trapdoor information is the pair $td = (R, s)$. Note that pk is not uniformly distributed, but pairs (pk, td) can be sampled in randomized polynomial time in λ.

Next we discuss how the function $f = f_{pk}$ can be evaluated, given the public parameters $pk = (A, z)$. We define two functions f_0, f_1 that should be understood as $f(0\|\cdot)$ and $f(1\|\cdot)$ respectively. Each function goes from $\mathbb{Z}_2^{wn}$ to $\mathbb{Z}_2^{wm}$ for $w = \lceil \log q \rceil$. For $b \in \{0,1\}$ the function f_b takes as input an $x \in \mathbb{Z}_q^n$ (that can be seen as an element of $\mathbb{Z}_2^{wn}$ through its binary representation) and returns $Ax + e' + bz$, which is an element of $\mathbb{Z}_q^m \subseteq \mathbb{Z}_2^{wm}$. Here, e' is a vector sampled at random from a distribution $D_{\mathbb{Z}_q,\alpha' q}$ such that α' is "much larger" than α. The inclusion of e' makes f a "randomized" function, which is the main way in which the construction differs from the requirements expressed in Section 1.5. A formal way around this is to think of f_b as the function that returns not $Ax + e' + bz$, but the *distribution* of $Ax + e' + bz$, when $e' \sim D_{\mathbb{Z}_q,\alpha' q}$ and all other variables are fixed. In practice, the evaluation of f on a quantum computer (as required of the honest prover in the verification protocol) involves preparing a weighted superposition over all error vectors, and computing the function in superposition.

We would, of course, rather do away with this complication. Why is the error vector necessary? It is there to satisfy the important requirement that the functions f_0 and f_1 are injective with overlapping ranges, so that f itself is 2-to-1. Injectivity follows from the existence of the trapdoor for A and an appropriate setting of the standard deviation of the error distribution, which guarantee that (given the trapdoor) x can be recovered from $Ax + e' + bz$ (with high probability over the choice of e'). To make the function ranges overlap, we need the distribution of $Ax + e'$ to be statistically close to the distribution of $Ax' + e' + z = A(x' + s) + (e' + e)$. The first distribution considers an arbitrary vector in the column span of A, shifted by e; the second considers the same, except that the shift is by $(e' + e)$. For the two distributions to (almost) match, we need the distribution of e' to (almost) match the distribution of $e + e'$. This is possible as long as the standard deviation $\sigma' = \alpha' q$ is substantially larger than the standard deviation $\sigma = \alpha q$; provided this holds it is an exercise to compute the statistical distance between the two Gaussian and verify that it can be made very close to 1.

With this important caveat in place, we have specified the function f and verified property (**F.1**). Property (**F.3**) follows from the existence of the secret information $td = (R, s)$. Given a $b \in \{0,1\}$ and an element $y = Ax + e' + bz = A(x + bs) + (e' + be)$ in the range of f_b it is possible to use the trapdoor matrix R to recover $x + bs$ and subtract bs to deduce the preimage x of z under f_b. Property (**F.4**) holds trivially from the construction. Note that the function f has domain and range that are different. In particular, here the domain is larger than the range, and in case q is not a power of 2 f is only defined on a subset of its natural domain $\mathbb{Z}_2^{wn}$. These points are not very important and can be ignored at the level of our discussion.

Showing the hardcore bit property (**F.2**) and the collapsing condition (**F.5**) require more work, and we refer to [BCM+18] for a detailed exposition.[5] Similar "hardcore bit" properties to (**F.2**) have been shown for many LWE-based cryptographic schemes (see e.g. [AGV09]). Usually the property states that "for any vector $d \in \mathbb{Z}_q^n \backslash \{0\}$, the value $d \cdot s \in \mathbb{Z}_q$ is indistinguishable from uniform, even given a sample $(A, As + e)$". Our property (**F.2**) is subtly stronger, in that the adversary may choose the vector d itself, possibly as a function of the sample $(A, As + e)$. An additional difficulty stems from the specific equation that the adversary is asked to return. In the definition of Assumption (**F.2**) this is a d such that $d \cdot (x_0 + x_1) = 0$, where x_0, x_1 are the *binary representation* of the two preimages in $\mathbb{Z}_q^n$ of the prover's first message string $y \in \mathbb{Z}_q^m$. (The use of the binary representation comes from the requirements on the honest prover, that is asked to perform a measurement in the Hadamard basis, yielding a binary string of outcomes.) So here

[5] The collapsing condition is not shown in [BCM+18]. It is implicitly shown in [Mah18], where it can be seen to follow from property 2 in Definition 4.4 of an extended trapdoor claw-free family. (The connection is made explicit in [GV19].)

$x_0 = (0, r_0)$ and $x_1 = (1, r_1)$ such that r_0, r_1 are binary representations for two elements $x'_0, x'_1 \in \mathbb{Z}_q^n$ such that $x'_1 = x'_0 - s$ over $\mathbb{Z}_q$. Since the binary representation is not linear the equation obtained is not directly a linear equation in the secret s. Completing the argument showing that a procedure that returns the information asked for in Assumption (**F.2**), i.e. the pair $(x = (b, r_b), d)$, can be turned into a procedure that breaks the decisional LWE assumption, requires a little more work; this is where we need to assume that the secret vector s is a binary vector.

Bibliography

[Aar10] Scott Aaronson. Bqp and the polynomial hierarchy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 141–150. ACM, 2010.

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[ACGK17] Scott Aaronson, Alexandru Cojocaru, Alexandru Gheorghiu, and Elham Kashefi. On the implausibility of classical client blind quantum computing. *arXiv preprint arXiv:1704.08482*, 2017.

[AG17] Dorit Aharonov and Ayal Green. A quantum inspired proof of $\mathrm{P}^{\setminus\#\mathrm{p}} \subseteq$ IP. *arXiv preprint arXiv:1710.09078*, 2017.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *Theory of Cryptography Conference*, pages 474–495. Springer, 2009.

[AV13] Dorit Aharonov and Umesh Vazirani. *Is quantum mechanics falsifiable? A computational perspective on the foundations of quantum mechanics*. Computability: Turing, Gödel, Church, and Beyond. MIT Press, 2013.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 320–331. IEEE, 2018.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

[CCKW19] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. Qfactory: classically-instructed remote secret qubits preparation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 615–645. Springer, 2019.

[CM16] Toby Cubitt and Ashley Montanaro. Complexity classification of local hamiltonian problems. *SIAM Journal on Computing*, 45(2):268–316, 2016.

[DFPR14] Vedran Dunjko, Joseph F Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 406–425. Springer, 2014.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N Rothblum. Delegating computation: interactive proofs for muggles. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 113–122. ACM, 2008.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 612–621. IEEE, 2017.

[GKW18] Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 660–670. ACM, 2018.

[GMR85] Shafi Goldwasser, Silvio Micali, and Ronald Rivest. A "paradoxical" solution to the signature problem. In *Advances in Cryptology*, pages 467–467. Springer, 1985.

[GV19] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1024–1033. IEEE, 2019.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. *Journal of the ACM (JACM)*, 62(6):45, 2015.

[KMW17] Elham Kashefi, Luka Music, and Petros Wallden. The quantum cut-and-choose technique and quantum two-party computation. *arXiv preprint arXiv:1703.03754*, 2017.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D Rothblum. How to delegate computations: the power of no-signaling proofs. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 485–494. ACM, 2014.

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

[MF16] Tomoyuki Morimae and Joseph F Fitzsimons. Post hoc verification with a single prover. *arXiv preprint arXiv:1603.06046*, 2016.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012.

[P+16] Chris Peikert et al. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[RRR16] Omer Reingold, Guy N Rothblum, and Ron D Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 49–62. ACM, 2016.

[RT19] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 13–23, 2019.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 497–527. Springer, 2016.

[Vid20] Thomas Vidick. Verifying quantum computations at scale: A cryptographic leash on quantum devices. *Bulletin of the American Mathematical Society*, 57(1):39–76, 2020.

[VZ20] Thomas Vidick and Tina Zhang. Classical proofs of quantum knowledge. *arXiv preprint arXiv:2005.01691*, 2020.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611. IEEE, 2017.