VERIFYING QUANTUM COMPUTATIONS AT SCALE: A CRYPTOGRAPHIC LEASH ON QUANTUM DEVICES

THOMAS VIDICK

Abstract. Rapid technological advances point to a near future where engineered devices based on the laws of quantum mechanics are able to implement computations that can no longer be emulated on a classical computer. Once that stage is reached, will it be possible to verify the results of the quantum device?

Recently Mahadev introduced a solution to the following problem: is it possible to delegate a quantum computation to a quantum device in a way that the final outcome of the computation can be verified on a classical computer, given that the device may be faulty or adversarial and given only the ability to generate classical instructions and obtain classical readout information in return?

Mahadev's solution combines the framework of interactive proof systems from complexity theory with an ingenious use of classical cryptographic techniques to tie a "cryptographic leash" around the quantum device. In these notes I give a self-contained introduction to her elegant solution, explaining the required concepts from complexity, quantum computing and cryptography and how they are brought together in Mahadev's protocol for classical verification of quantum computations.

Boaz suggest making the analogy with MAXCUT, and possibly using that as a starting point Boaz points out that the 2-to-1 structure makes the commitment perfectly hiding, and this is important for preserving the quantum superposition Amnon suggests this should called an argument, not a proof system Zvika asks if parallel repetition is trivial/doable

Quantum mechanics has been a source of endless fascination throughout the 20th century — and continues to be in the 21st. Two of the most thought-provoking aspects of the theory are the *exponential scaling* of parameter space (a pure state of n qubits requires $2^n - 1$ complex parameters to be fully specified), and the *uncertainty principle* (measurements represented by non-commuting observables cannot be performed simultaneously without perturbing the state). The conceptual difficulty of the problem of verification of quantum computations stems from both aspects. Suppose given the description of an experiment that can be modeled in quantum mechanics — say, a number n of individual photons are emitted by lasers in a certain configuration, then made to interact according to optical equipment such as mirrors and beam-splitters, and finally some observation is made, for example counting the number of photons that hit a strategically located detector within a certain time period. Quantum mechanics provides a set of formal rules that, *in principle*, allow for the computation of the distribution of possible outcomes obtained in this experiment — what is the probability that any number of photons hit the detector within the prescribed time frame. These rules yield extremely precise

2010 *Mathematics Subject Classification.* Primary 68Q12.

predictions that have been verified in countless experiments. In general however, computing the prediction requires a number of operations that scales *exponentially* with n, the total number of photons in the experiment. What this means in practice is that as soon as n exceeds, say, 80, it becomes all but infeasible, using even the most powerful supercomputers available today, to predict the outcome of any nontrivial quantum experiment.

In addition to the intrinsic exponential scaling of quantum mechanics, other quantum phenomena, such as the uncertainty principle, place fundamental limits on our ability to verify that a quantum mechanical evolution proceeds as expected. Any attempt by the experimentalist at making intermediate observations on the state of a subset of the elementary particles involved in her experiment risks altering its outcome, so that it is not clear at all if the results of low-level benchmarks can be meaningfully pieced together to certify the final result.

These obstructions should come as no surprise. Indeed it is the same difficulty, of classical simulation of quantum evolutions, that prompted Feynman to bring forward the idea of a quantum computer in the first place: a computer that by its very nature would have the ability to efficiently simulate any quantum process [15]. While such a "universal quantum simulator" remains a distant technological challenge, smaller-scale quantum devices have begun to appear that will soon have the capacity to simulate the evolution of specific quantum-mechanical systems, enabling physicists to e.g. make predictions regarding properties of new materials (see e.g. [7]). Such simulators will become interesting the day when they are able to generate predictions that could not have been obtained on a classical computer. Assuming that such a "classically unsimulatable quantum simulator" exists, can one check that the simulator accomplishes the task it was asked to perform — given that the task could not have been accomplished on a classical computer? If the simulator makes a wrong prediction, be it due to a fault in its implementation, or even due to malice on the implementer's part, is there any way that the error can be detected without having to rely on yet another quantum simulator to duplicate the first simulator's results?

Not all is lost. There obviously are *some* quantum computations whose outcome can be easily checked. A standard example is factoring: having run Shor's quantum algorithm for finding a prime factor p of an integer n, it is easy to execute Euclid's algorithm to check that $p|n$. In the language of complexity theory, the complexity class[1] associated with the set of languages that can be decided efficiently with the help of a classical (probabilistic) computer is denoted BPP (for "bounded-error probabilistic polynomial-time"), while with a quantum computer one gets BQP (for "bounded-error quantum polynomial-time"). Factoring is an example of a problem that lies in the class BQP, but is not known to lie in the class BPP.[2]

As emphasized above, the factoring problem has the additional property that solutions to it are easily verifiable. Recognizing that many algorithmically difficult problems seemed to share the feature of having easily verifiable solutions (other examples are the problem if deciding if two graphs are isomorphic and the problem

[1]A complexity class is a collection of languages, and a language is a set of strings; for example, the set of all binary representations of graphs that are 3-colorable is a language. See Section 1 for more background on complexity theory.

[2]Technically, one would have to associate a language, i.e. a set of bitstrings, to factoring; an example is $L_{factoring} = \{$binary representations of a pair of integers (n, k_1, k_2) such that n has a prime factor p such that $k_1 \leq p \leq k_2\}$.

of deciding if a graph is 3-colorable), Karp introduced in 1970 the complexity class NP (for "non-deterministic polynomial-time"). Informally, NP is the class of all languages that can be efficiently *verified* on a classical computer, given the right witness, or proof. If every problem in BQP, i.e. every problem that can be solved efficiently on a quantum computer, had this property — that the correct outcome of the computation can be efficiently verified on a classical computer — then the question of classical verifiability of quantum computation would be moot. However, there are very good reasons to think that this is not the case.

Indeed, complexity theorists strongly believe that there are problems in BQP that do not lie in NP.[3] One candidate for such a problem is the "forrelation" problem introduced in [1]. The input to this problem is a pair of Boolean functions $f, g : \mathbb{Z}_2^n \to \{-1, 1\}$, and it is promised that g and the Fourier transform $\hat{f}$ of f are either highly correlated, $|\langle \hat{f}, g \rangle| \geq 0.9$, or barely correlated, $|\langle \hat{f}, g \rangle| \leq 0.1$.[4] The goal is to determine which of the two cases hold. There is an easy quantum algorithm for this problem that requires a single evaluation of the functions f and g, whereas it was shown in [27] that, if access to f and g is restricted to evaluation queries, then the problem lies beyond even the polynomial hierarchy, which is a vast extension of the class NP. The open challenge is to find an explicit family of functions f, g for which the problem remains hard, even when given the explicit description of the functions (such as an evaluation circuit) as input.

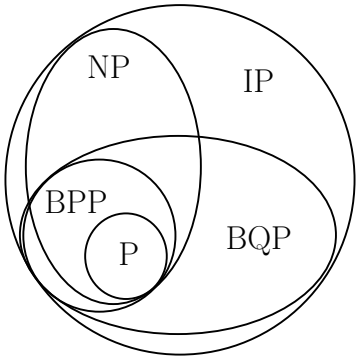

FIGURE 1. Known inclusions between complexity classes. Note that NP is not known to contain BPP because in NP, the verification procedure is assumed deterministic. With a randomized verifier, one obtains the class MA (for "Merlin-Arthur") that does contain BPP.

There exists a wide range of problems that are known to be efficiently solvable on a quantum computer, but for which there is no obvious efficiently verifiable proof (such as the forrelation problem). Is it possible to design theoretical experiments, or protocols, based on quantum mechanics, for such problems such that the proper execution of the experiment, or computation, can be certified correct using

[3]They also believe that there are problems in NP, such as the traveling salesman problem or any NP-complete problem, that are not in BQP, so that the two classes are incomparable.

[4]To be precise, the definition is $\langle \hat{f}, g \rangle = 2^{-3n/2} \sum_{x,y \in \mathbb{Z}_2^n} (-1)^{x \cdot y} f(x) g(y)$.

a verification procedure that does not itself rely on the manipulation of quantum information?

In these notes I present an elegant solution to this problem due to Mahadev [24]. Before presenting the solution, a first step is to formalize the question. This is done in two steps. First, in Section 1 I introduce the complexity-theoretic formalism of interactive proofs and arguments that forms the conceptual backdrop against which Mahadev's result is formulated. Second, in Section 2 I give standard background in quantum computing that allows me to formalize the complexity class BQP and state Mahadev's theorem. The remainder of the document is devoted to a presentation of the main ideas that constitute the proof of Mahadev's result. Very roughly, the proof proceeds through the following steps.

- An arbitrary quantum computation can be specified classically by providing the gate-by-gate description of a quantum circuit, as well as a classical description of the input on which the circuit is to be evaluated (generally, this will be a zero state representing a default initialization of all wires of the circuit). The first step is to reduce the problem of evaluating the output of such a circuit C to the problem of verifying that a local Hamiltonian H, that can be efficiently computed from C classically, has a smallest eigenvalue that is below a certain threshold. Informally, this step is the quantum analogue of encoding the tableau of a classical computation into an instance of the Ising spin problem. This step is described in Section 3.
- The previous step has reduced the verification problem to the problem of *classically* certifying that a local Hamiltonian has an eigenstate with small enough eigenvalue. Section 4 introduces the idea of a *commitment*, which is an interactive procedure by which a party can *commit* to a certain value b, without revealing information about b, yet in a way that is *biding*: the party is unable to later "change its mind" with respect to b and claim that it in fact had committed to some $b' \neq b$. The notion of a commitment protocol for quantum states is the key ingredient in Mahadev's protocol and is presented in Section 5.
- Armed with the notion of a quantum commitment scheme we describe the verification protocol itself in Section 6. Informally, in the protocol the prover first provides classical information that is meant to indicate the prover's "commitment" to being in the possession of a unique, well-defined quantum state ψ. Then, the verifier challenges the prover to "reveal" measurement outcomes performed on the state ψ that it committed to, in a way that allows the verifier to estimate the associated energy $\psi^* H \psi$ and verify that it is sufficiently small.

The security of Mahadev's quantum commitment scheme, on which her result ultimately rests, relies on a computational assumption regarding the impossibility of efficiently solving a problem known as the Learning with Errors (LWE) problem. This problem is described in Section 7.

These notes are written in a way that aims to make the most important insights of Mahadev's work accessible to any mathematician, with or without background in complexity theory and quantum information. As a result, the presentation will remain rather informal, and we alert the reader whenever the discussion makes an important shortcut.

To keep the presentation focused we do not survey prior works and other approaches to verification in any depth. The question has a long history, that prior to Mahadev's work had resulted in partial answers, obtained in a variety of models of verification. Some of the most important results include the concurrent works of Aharonov et al. [2] and Broadbent et al. [12] which showed how to achieve verification in a model where the verification procedure can make use of a small, trusted quantum computer, and the work of Reichardt et al. [29] in a model where the verification procedure is entirely classical, but has access to two spatially isolated quantum computers, sharing entanglement, whose joint implementation of a quantum computation it aims to verify. In contrast to Mahadev's result presented here, these works achieve information-theoretic (instead of computational) guarantees in their respective models. For an in-depth discussion of these and related works, I recommend the recent survey by Gheorghiu et al. [17].

1. Interactive poofs and arguments

The concept of an *interactive proof system* can be difficult to digest for the mathematician, in part because it involves some amount of personification. When they talk of an interactive protocol, computer scientists are accustomed to refer to imaginary beings known as the *verifier* and the *prover*. Even worse, these imaginary beings are endowed with *intentions*: the prover is *trying* to demonstrate something to the verifier, while the verifier *attempts* to catch any *cheating behavior* from the prover. This is not the kind of language that is frequently used in, say, the theory of differential equations or operator spaces. Please bear with me — interactive proofs are one of the most powerful ideas to have emerged out of complexity theory since the 1990s, and they are a key element of Mahadev's solution.

It all starts with the complexity class NP, whose study originates in the works of Cook, Karp, and Levin in the 1970s. A complexity class is a collection of *languages*. A (promise) language L is a pair of subsets $L_{yes}, L_{no} \subseteq \{0,1\}^*$, where $\{0,1\}^*$ is the set of all "bit strings": sequences of arbitrary but finite length over the alphabet $\{0,1\}$. For example, L_{yes} could be the set of (suitably encoded) 3-SAT formulas that admit a satisfying assignment,[5] and L_{no} the set of formulas that are not satisfiable. The language $L = (L_{yes}, L_{no})$ is called 3-SAT. Informally, a language L is in the class NP if valid statements have an efficiently verifiable proof, and invalid statements have no valid proof. To formalize the notion of "valid proof", we introduce the notion of "verifier", represented by a polynomial-time Turing machine V (for our purposes, the reader may replace the intimidating notion of "Turing machine" by any intuitive notion of efficient computation, such as an "algorithm") whose goal is to "verify" claimed proofs. Thus a language L is in the class NP if there exists a real polynomial p and a Turing machine V such that for all $x \in L_{yes} \cup L_{no}$, (i) if $x \in L_{yes}$ then there exists a w, the *witness*, or *proof*, such that $V(x,w)$ halts in at most $p(|x|)$ steps, where $|x|$ denotes the length of x, and returns 1 (for "accept"), and (ii) if $x \in L_{no}$ then for any w, $V(x,w)$ halts in at most $p(|x|)$ steps and returns 0 (for "reject"). Property (i) is usually referred to as the *completeness* condition, and (ii) as the *soundness* condition. The fact that 3-SAT is in NP follows since given as input a 3-SAT formula φ, if the formula is

[5] A 3-SAT formula is an AND of 3-variable ORs, i.e. a Boolean formula of the form $\varphi = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_5 \vee \overline{x_6}) \wedge \cdots$, where the variables $x_i \in \{0,1\}$ and $\overline{x_i}$ denotes variable negation.

satisfiable then there exists a witness w (a satisfying assignment for φ) that proves this, whereas if the formula is not satisfiable, it is easy for V to check that any given purported assignment w is indeed invalid.

Informally, NP captures the collection of problems that have efficiently verifiable solutions, such as factoring, 3-SAT, the traveling salesman problem, or mathematical theorems (that have reasonably short proofs within a prespecified axiomatic system).

1.1. **Interactive proof systems.** A key insight from research in complexity theory in the 1990s is that the collection of languages that admit "efficient verification" can be substantially extended beyond the class NP by allowing *interactive protocols* for verification [5, 18]. Consider a verification procedure V, referred to as the *verifier* (personification, here it comes...), that is allowed to "ask questions" to another entity, the *prover*, about a claimed proof, as illustrated in Figure 2. Can this allow verification of languages other than the languages in NP, which admit "static" proofs?

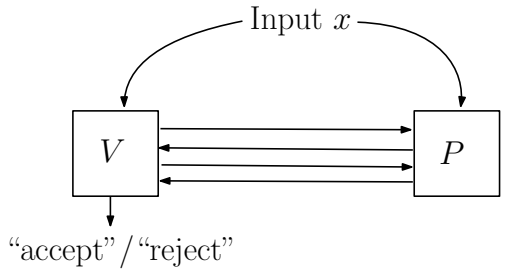

FIGURE 2. An example of a 4-message interactive proof between a verifier V, assumed to be computationally bounded, and an all-powerful prover P.

The verifier in an interactive proof system is modeled as a *randomized polynomial-time procedure*. Informally, this refers to any algorithm that makes a total number of steps that is bounded by a polynomial function (that may depend on the algorithm) of the bitlength of its input, and has access to a source of uniformly random bits. Formally, the verifier is modeled by a randomized interactive Turing machine, but we will not need to descend to that level of detail. It is important that the verifier be randomized: indeed, interactive proof systems with deterministic verifiers reduce to NP, because in the case of deterministic messages from the verifier there is no advantage to interaction (the prover can send the entire transcript as "proof"). A randomized verifier may sometimes make an erroneous decision, as long as for every input the probability of making an error in deciding that input is small: this ensures that repeating the verification procedure sufficiently many times and taking the majority decision yields an outcome that is erroneous with arbitrarily small probability.

Definition 1.1. A language $L = (L_{yes}, L_{no})$ has an interactive proof system if there exists a randomized polynomial-time verifier V such that the following hold.

- *Completeness:* For any $x \in L_{yes}$, there exists a prover P, referred to as the "honest prover", such that V accepts P on input x with probability at least $\frac{2}{3}$.

- *Soundness:* For any $x \in L_{no}$ and any prover P^*, V accepts P^* on input x with probability at most $\frac{1}{3}$.

The collection of languages for which there exists an interactive proof system is denoted IP.

Note the asymmetry between the "completeness" and "soundness" conditions in the definition, which reflects a similar asymmetry in the definition of NP: it is always assumed that the prover aims to convince the verifier that the input is *in* the language (e.g. for the case of NP, that a 3SAT formula is satisfiable). The verifier, in turn, aims to make the right decision, by taking information from the prover but "verifying" it before making a decision, so as not to wrongly accept a negative instance.

To gain intuition as to how interaction may help, consider the following toy problem. The input x is interpreted as a bivariate polynomial $P(y, z)$ defined over a prime field $\mathbb{F}_p$, such that P has degree at most d in each of its variables. Think of p as much larger than d: the input size, measured in number of bits, is roughly $(d+1)^2 \log p$ (a list of coefficients), and p could be exponentially larger. Let L_{yes} be the set of all such polynomials such that $S = \sum_{y,z} P(y, z) = 0$, and L_{no} the set of polynomials such that $S \neq 0$. Computing S naïvely takes time $O(p^2)$, where the $O(\cdot)$ hides a multiplicative factor of order the time it takes to evaluate P at any given point, i.e. polynomial in d and $\log p$. Consider the following protocol, using which the verifier can be led to making the right decision while having to invest a computational effort of $O(p)$ only. The first step in the protocol is a message from the prover to the verifier, which consists in the claimed value of S, together with a "supporting statement" in the form of a degree-d polynomial $Q(z)$ such that $S = \sum_y Q(z)$. (The honest prover can find such a Q without difficulty by setting $Q(z) = \sum_y P(y, z)$.) Upon receipt of an arbitrary (S, Q), the verifier first checks the equality $S = \sum_z Q(z)$, which takes $O(p)$ time. Next, she selects a uniformly random $z^* \in \mathbb{F}_p$ and checks that $Q(z^*) = \sum_y P(y, z^*)$, which again requires time $O(p)$. Note that, if it was the case that $Q(\cdot) \neq \sum_y P(y, \cdot)$, by the Schwartz-Zippel lemma the probability that $Q(z^*) = \sum_y P(y, z^*)$ would be at most d/p, which is small provided p is much larger than d, as we assumed. Thus the verifier makes the right decision with high probability, on any input P.

This "protocol" reduces the verifier's effort from order p^2 to order p, since the verifier only needs to perform summations over a single variable at a time, instead of both variables for the naïve computation. The attentive reader will have realized that the protocol is in fact non-interactive — there is a single message from the prover to the verifier. Applying the same idea to polynomials with more variables yields an interactive protocol, in which variables are randomly fixed by the verifier one at a time, with exponential savings in the amount of time required for verification, from order p^m to order mp, with m representing the number of variables. This idea, of efficiently verifying claims about the sum of the values taken by a multivariate polynomial, is central to the proof of a celebrated result in complexity theory, namely the inclusion of PSPACE (the class of languages that can be decided using polynomial space, but arbitrary time) in IP [23, 30]. While the state of the art in complexity theory does not allow one to prove that PSPACE is a strictly larger class than NP, this is generally believed to be the case, so that interaction seems to significantly broaden the class of languages that have efficient verifiers.

In fact, the possibility of interaction sufficiently strengthens the verifier so as to allow it to verify a class of languages that includes the class BQP of languages that can be efficiently decided on a *quantum* computer! Using the idea of Feynman path integrals the probability that a quantum circuit returns the outcome "1" can be expressed as the sum of exponentially many complex numbers, each of which can be directly computed by taking a product of entries of the matrices that specify the quantum gates of the circuit; this exponential sum can be exactly computed (modulo finite-precision issues) in polynomial space. Given that PSPACE is in IP, it follows that there exists an interactive protocol, of the form described above, that allows a classical polynomial-time verifier to verify the outcome of a quantum computation by asking questions to an untrusted prover. But there is a hitch. The model of interactive proofs does not place limitations on the computational effort required of the prover. In the protocol for computing sums of polynomials described earlier, the prover has to compute multiple exponential-sized intermediate sums, that in general may take time p^m tocompute. Unfortunately, following the proofs that BQP is in PSPACE is in IP leads to a very similar protocol, in which the prover has to compute exponentially large sums that do not seem to be amenable to efficient computation even by a quantum procedure. There has been very little progress in tweaking the protocols obtained in this way to decrease the computational effort required for the honest prover (see e.g. [3] for a discussion).

1.2. **Interactive arguments.** If we are to make the requirement that the actions of the honest prover, i.e. the prover in charge of convincing the verifier in the case of an input in L_{yes}, can be implemented efficiently (on a quantum computer), it is reasonable also to ask, not that there does not exist a P^* that would improperly convince the verifier to accept an input in L_{no}, but merely that no such P^* exists that can be implemented in (quantum) polynomial time. Interactive proof systems such that the soundness condition holds under such a computational assumption are called *arguments*, a notion introduced in [11].

Since it is not known that the classes P and NP, or even P and IP, are distinct, in order for the assumption to be effective we further need to posit that a particular (class of) problems *cannot* be solved in (quantum) polynomial time. For example, we could make the assumption that the problem of factoring an integer cannot be solved in probabilistic polynomial time. Another example is that a certain explicit family of functions $f = \{f_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$, where $\lambda \in \mathbb{N}$ plays the role of the input size parameter, cannot be inverted with non-negligible (in λ) success probability in (quantum) polynomial time.[6] We refer to any such assumption as a *computational assumption*. The following definition parallels Definition 1.1.

Definition 1.2. A language $L = (L_{yes}, L_{no})$ has an interactive argument *under computational assumption (A)* if there exists a randomized polynomial-time verifier V such that the following hold.

- *Completeness:* For any $x \in L_{yes}$, there exists a prover P such that V accepts P on input x with probability at least $\frac{2}{3}$.[7]

[6] A negligible function $\varepsilon(\lambda)$ is one such that $\varepsilon(\lambda)p(\lambda) \to_{\lambda\to\infty} 0$ for any polynomial $p(\lambda)$. A non-negligible function grows faster than any negligible function.

[7] The definition of the completeness property does not explicity require P to be efficient. In practice, the properties of the honest prover are discussed on a case-by-case basis, depending on the argument system considered.

- *Soundness:* For any $x \in L_{no}$ and any (quantum) polynomial-time prover P^*, V accepts P^* on input x with probability at most $\frac{1}{3}$.

The relaxed soundness condition of an interactive argument has proved fruitful to design more efficient proof systems than are known otherwise. For example, there are arguments for languages in NP such that the total communication is only poly-logarithmic in the size of the input [22]; no such protocol is known in the interactive proof model. The construction of such arguments rests upon two notions: *probabilistically checkable proofs* and *commitments*. Intuitively, the computational assumption is used by the verifier to "delegate" parts of its own verification procedure to the prover. The notion of "commitment" is used to tie the prover to performing the right actions. We discuss commitments in Section 4, as they play a central role in Mahadev's interactive argument for quantum computation. To formulate her result precisely, it only remains to introduce the formalism of quantum computation, and the associated complexity class BQP. This is done in the next section.

2. Quantum computation

In this section we give a light introduction to the formalism of quantum computing. Our goal in doing so is to provide the minimal background required to formally state Mahadev's theorem, which is given at the end of the section, as well as describe the main ideas of her proof, introduced in the following sections. (The reader already familiar with quantum computing may directly skip ahead to the statement of Theorem 2.3 at the end of the section.)

2.1. **Quantum states and observables.**

2.1.1. *States.* An n-qubit quantum state is specified by a *density matrix*, a positive semidefinite matrix ρ on the 2^n-dimensional Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ such that ρ has trace 1. Density matrices generalize classical probability distributions over n bits, as the latter can be represented by a probability vector $p : \{0,1\}^n \to [0,1]$ that we can embed on the diagonal of a density matrix.

Even though in general the formalism of quantum mechanics extends to arbitrary separable Hilbert spaces, for convenience in these notes Hilbert spaces always come endowed with a canonical decomposition as a tensor product of n copies of $\mathbb{C}^2$, for some finite integer n, such that each copy of $\mathbb{C}^2$ has a canonical basis (e_0, e_1). We use the "ket" notation to write the canonical basis as $|0\rangle = e_0$, $|1\rangle = e_1$, and we refer to this basis as the "computational basis". A quantum state is called *pure* if its density matrix has rank 1; in this case we can also represent the state as a unit vector expressed in the canonical basis $\{e_{x_1} \otimes \cdots \otimes e_{x_n}, x \in \{0,1\}^n\}$, or $\{|e_{x_1} \cdots e_{x_n}\rangle\}$ in the more compact ket notation. An arbitrary pure state thus has an expansion $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$, where the $\{\alpha_x\}$ are complex coefficients such that $\sum_x |\alpha_x|^2 = 1$. The associated density matrix is the rank-1 projection $\rho = |\psi\rangle(|\psi\rangle)^\dagger = |\psi\rangle\langle\psi|$, where the "bra" notation $\langle\psi| = (|\psi\rangle)^\dagger$ is used for the conjugate-transpose. For a density matrix ρ that lies in the tensor product of multiple spaces we write e.g. $\rho_{\mathsf{AB}} \in \mathcal{H}_{\mathsf{A}} \otimes \mathcal{H}_{\mathsf{B}}$ using boldface A, B to identify the different subsystems, often referred to as *registers*.

2.1.2. *Observables.* A measurement of a set of qubits is specified by an orthonormal basis of the Hilbert space associated with the qubits. The outcome of the measurement is the label of one of the basis vectors, and the probability with which each basis vector is obtained equals the squared norm of the component of the state that is in the direction of that basis vector. Formally, suppose $|\psi\rangle$ is a state in $(\mathbb{C}^2)^{\otimes n} \otimes (\mathbb{C}^2)^{\otimes m}$, and that the first n qubits of $|\psi\rangle$ are measured in the orthonormal basis $\{|\phi_i\rangle\}_{i\in 1,\ldots,2^n}$ of $(\mathbb{C}^2)^{\otimes n}$. To compute the probability of the i-th outcome being obtained, we expand $|\psi\rangle$ in the basis $\{|\phi_i\rangle\}$ as

$$|\psi\rangle \;=\; \sum_{i=1}^{2^n} |\phi_i\rangle \otimes |\phi'_i\rangle \;,$$

where the $|\phi'_i\rangle$ are arbitrary vectors in $(\mathbb{C}^2)^{\otimes m}$ (not necessarily normalized or orthogonal). The probability of the i-th outcome is given by $\||\phi'_i\rangle\|^2$. It will later be important to remember that a measurement *collapses* the state: once the outcome i has been obtained and recorded, the state undergoes a non-unitary evolution $|\psi\rangle \mapsto |\psi_i\rangle = \frac{|\phi_i\rangle|\phi'_i\rangle}{\||\phi'_i\rangle\|}$.[8]

A measurement in the basis $\{|\phi_i\rangle\}$, together with a choice of a real number λ_i associated with each outcome i, can be succinctly represented as an "observable" $O = \sum_i \lambda_i |\phi_i\rangle\langle\phi_i|$. For a quantum state ρ, the real number $\mathrm{Tr}(O\rho)$ is precisely the expectation of λ_i, under the distribution on i obtained by measuring the state ρ in the basis $\{|\phi_i\rangle\}$. An example is the observable associated with a measurement of a qubit in the computational basis $\{|0\rangle, |1\rangle\}$, labeling the first outcome "1" and the second "-1". The associated observable is the Pauli σ_Z matrix,

$$\sigma_Z \;=\; |0\rangle\langle 0| - |1\rangle\langle 1| \;=\; \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \;.$$

Similarly, a measurement in the Hadamard basis $\{H|0\rangle, H|1\rangle\}$ is represented by the Pauli σ_X observable,

$$\sigma_X \;=\; H\sigma_Z H^\dagger \;=\; \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \;.$$

2.2. **Circuits.** Evolution in quantum mechanics is unitary. Given a unitary U, a pure state $|\psi\rangle$ evolves as $|\psi\rangle \mapsto U|\psi\rangle$, and a density matrix ρ evolves as $\rho \mapsto U\rho U^\dagger$.

Not every unitary requires the same amount of resources to be implemented. Multiple models have been introduced to evaluate the complexity of implementing a unitary transformation. These include quantum Turing machines, the quantum circuit model, measurement-based computation, adiabatic computation, topological computation, and others. All these models have been shown equivalent up to a polynomial re-scaling of the resources required. For our purposes, the simplest and most convenient model is the circuit model, that we describe next.

In the circuit model, in order to implement an n-qubit unitary U the unitary must first be decomposed as a product $U = U_T \cdots U_1$, where each U_i is a unitary that acts non-trivially on at most two qubits, i.e. it can be written as a tensor product of a unitary on $\mathbb{C}^2 \otimes \mathbb{C}^2$ with the identity on the remaining space. Moreover, each U_i should be taken from a finite "gate set" of allowed operations on the computer. A

[8] If the conflict between the statements that "quantum mechanics requires all evolutions to be unitary" and "a measurement is an irreversible process" puts you ill at ease, you are not alone.

widely used gate set is $\{H, T, CNOT\}$, where $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is the Hadamard gate, $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ the T (sometimes also called $\pi/8$) gate, and CNOT the two-qubit unitary that sends $|a\rangle|b\rangle \mapsto |a\rangle|a \oplus b\rangle$ for any $a, b \in \{0,1\}$.

A fundamental theorem in quantum computing, the Solovay-Kitaev theorem, states that for the purpose of efficient circuit representation any finite set of 1 and 2-qubit gates is as good as any other, as long as it generates a dense subgroup in $SU(2)$ (which is the case for the above-defined set). More formally,

Theorem 2.1 (Solovay-Kitaev '97)**.** *There is a constant c such that for any finite gate set $G \subseteq SU(2)$ such that the group $\langle G \rangle$ generated by G is dense in $SU(2)$ and G is closed under inverse, for any $\varepsilon > 0$ there is an $\ell = O(\log^c(1/\varepsilon))$ such that G^ℓ is an ε-net in $SU(2)$.*[9]

We can now define the class BQP of problems that we are concerned about.

Definition 2.2. A promise language $L = (L_{yes}, L_{no})$ is in BQP if there exists a classical deterministic polynomial-time Turing machine such that, on input $x \in L_{yes} \cup L_{no}$ the Turing machine returns the description of a quantum circuit C over the gate set $\{H, T, CNOT\}$, together with a specially designated output qubit for the circuit, such that

- If $x \in L_{yes}$ then the probability that a measurement of the output qubit of C, when all input qubits are initialized to the $|0\rangle$ state, is 1, is at least $\frac{2}{3}$;
- If $x \in L_{no}$ then the same probability is at most $\frac{1}{3}$.

Due to the fact that quantum gates are by necessity reversible, it may not be immediately obvious that quantum computations generalize classical computation. In fact, given a function $f : \{0,1\}^n \to \{0,1\}^m$ specified by a classical circuit, it is always possible to devise a quantum circuit of comparable size for the unitary U_f that maps $|x\rangle|b\rangle$ to $|x\rangle|f(x) \oplus b\rangle$ for $x \in \{0,1\}^n$, $b \in \{0,1\}^m$. This can be achieved by showing that any classical circuit over, say, $\{AND, OR, NOT\}$ can be efficiently simulated by a reversible circuit; we omit the details. We often consider the application of the unitary U_f "in superposition": by linearity,

$$U_f : \quad \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle|0^m\rangle \mapsto \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle|f(x)\rangle \ .$$

We close this section by stating the main result that these notes aim to present. The computational assumption that underlies soundness of the interactive argument, the learning with errors assumption, is a standard assumption in post-quantum cryptography (classical cryptography designed to be secure against quantum adversaries) that we review in Section 7.

Theorem 2.3 (Mahadev 2018)**.** *Any language in BQP has an interactive argument that is computationally sound against quantum polynomial-time provers under the learning with errors assumption. Moreover, the verifier runs in probabilistic polynomial time, the honest prover can be executed in quantum polynomial-time, and the interaction between verifier and prover consists of 2 rounds (4 messages) only.*

[9] An ε-net is a set of points $S \subseteq SU(2)$ such that for all $U \in SU(2)$, there is $V \in S$ such that $\|U - V\| \leq \varepsilon$. Here the norm is the operator norm, but any norm would give essentially the same result, since the space has small dimension.

It is an important open question if the same theorem can be stated in the formalism of interactive proofs, instead of arguments (i.e. without making a computational assumption on the cheating prover), while still keeping the condition that the honest prover can be executed in quantum polynomial time. Intuitively, in Mahadev's proof system the post-quantum cryptographic assumption is used to restore symmetry between the quantum prover and the less powerful classical verifier. A specific cryptographic construction, a *quantum commitment scheme*, is used to "tie the prover's hands" in a way such that it has no choice but to implement the required computation — unless it has the power to break the cryptographic assumption. Nevertheless, while it is clear how the assumption is leveraged in the design of the protocol, it is not known whether one could do without it.

3. Certificates for computation

The first step in the proof of Theorem 2.3 is to identify a *quantum* certificate for the validity of a given quantum computation (i.e. a certificate that the probability that a quantum circuit provided as input returns the outcome "1", when all input qubits are initialized to the $|0\rangle$ state, is at least $\frac{2}{3}$; cf. Definition 2.2). This is achieved by the main theorem of this section, Theorem 3.1, that is due to Kitaev and predates Mahadev's work. In the following sections we will show how the existence of the kind of certificate provided by Theorem 3.1 can be verified using an interactive argument with a classical verifier.

We first consider the case of a classical computation. Given as input the description of a classical circuit, what is a good "certificate" for the claim that the circuit returns the value 1 when all input bits are initialized to the 0 state? While such a certificate is not really needed, as the verifier can simply execute the circuit by itself, this solution does not generalize well to the case of a quantum circuit and a classical verifier. In the following section we leverage the theory of NP-completeness to identify a particular kind of certificate for the validity of a classical circuit that has the advantage that it will generalize to the case of quantum circuits.

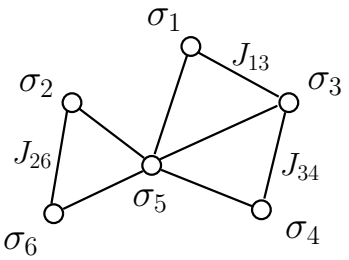

Figure 3. Schematic representation of an instance of the Ising spin problem. Here each σ_i is a variable in $\{0,1\}$, and each J_{ij} a fixed coupling constant in $[-1,1]$. The goal is to find an assignment to the variables that minimizes the expression (3.1).

3.1. Warmup: the Ising spin problem.

We consider a problem from classical statistical physics and show how to reduce the verification of a *classical* computation to it. Consider a graph with n vertices, such that each vertex $i \in \{1,\ldots,n\}$ is associated a value (or "state") $\sigma_i \in \{0,1\}$, and each edge (i,j) is associated a real

weight (or "coupling constant") J_{ij} such that $|J_{ij}| \leq 1$. (See Figure 3.) Vertices represent particles that can be in one of two states, $\sigma_i = 0$ or $\sigma_i = 1$, and edges represent interactions between particles, where the interaction can be attractive ($J_{ij} \geq 0$) or repulsive ($J_{ij} < 0$). The *energy* of a configuration $\sigma \in \{0,1\}^n$ is defined as[10]

$$H_{ising}(\sigma) = -\sum_{(i,j)} J_{ij} (-1)^{\sigma_i + \sigma_j} . \tag{3.1}$$

Informally, the energy functional (a.k.a. *Hamiltonian*) H_{ising} measures the number of interaction constraints that are not satisfied by an assignment σ, with each violation giving a penalty of $|J_{ij}|$. It is well-known that the problem of deciding, given as input n, the coefficients J_{ij}, and two thresholds a, b such that $b-a$ is at least a constant independent of n,[11] whether the minimum of H_{ising} over all $\sigma \in \{0,1\}^n$ is less than a, or larger than b, is an NP-complete problem (i.e. any problem in NP reduces to it); moreover, this holds even for the case where $J_{ij} \in \{-1, 0, 1\}$ for all (i, j).

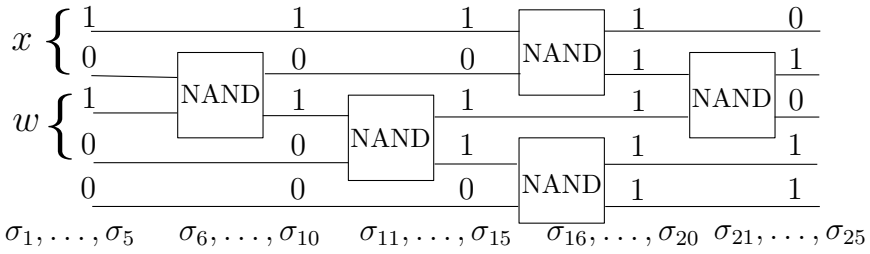

FIGURE 4. The tableau of a classical circuit. Here $x = 10$, $w = 10$, and there is one ancilla qubit, initialized to 0. The circuit has 5 NAND gates $(a, b) \mapsto (a, 1 - ab)$.The tableau is given by $\sigma \in \{0,1\}^{25}$, that represents the state of each of the 5 circuit wires at successive stages of an execution of the circuit.

To understand why the Ising problem is as hard as any problem in NP, let's see how we can reduce to it the problem of deciding whether, given a classical circuit $\mathcal{C}$ acting on $n + m + r$ bits and an input string $x \in \{0,1\}^n$, there exists a string $w \in \{0,1\}^m$ such that the circuit accepts $(x, w, 0^r)$. By definition any problem in NP can be expressed in this form, with x as the input, w as the witness, and $\mathcal{C}$ as the verifier's circuit. In general $\mathcal{C}$ is specified by a sequence of gates $(C_1, \ldots, C_\ell)$ taken from some fixed gate set, that we may without loss of generality restrict to the sole NAND gate.[12] Next consider the *tableau* of the computation performed by the circuit $\mathcal{C}$. (See also Figure 4.) This is simply a list of values associated with each wire in the circuit: the input wires (initialized to $(x, w, 0^r)$), the output wire of

[10]Technically the expression in (3.1) can take negative values, which may not seem appropriate for an "energy". Correcting this is a matter of introducing an additive shift.

[11]It is typical to normalize H_{ising} by dividing by its "norm" $\sum_{i,j} |J_{ij}|$, in which case the promise on the gap $b - a$ becomes that it is at least an inverse polynomial in n.

[12]The NAND gate maps (a, b) to $(a, 1 - ab)$. (The first output is included for convenience, to keep the number of wires constant, but is not necessary.) It is a universal gate, meaning that any circuit can be simulated by one made of NAND games only with a polynomial overhead in the total number of gates.

any gate in the circuit, and the final output wire (that should equal 1, which stands for "accept"). Given a tableau, it is possible to verify that the tableau is correct by checking the propagation of each gate, one at a time: if the inputs to a NAND gate are $\sigma_{i_1}, \sigma_{i_2} \in \{0,1\}$, the output should be $(\sigma_{i_3} = \sigma_{i_1}, \sigma_{i_4} = 1 - \sigma_{i_1}\sigma_{i_2})$. Wires corresponding to the input string x should be initialized to the corresponding input bit, whereas wires associated with the witness string w can take arbitrary values. We can express this set of constraints as a Hamiltonian $H_{\mathcal{C}} : \{0,1\}^T \to \mathbb{R}$, where T is the total number of wires in the circuit:

$$H_{\mathcal{C}} \;=\; H_{in} + H_{prop} + H_{out}\;, \tag{3.2}$$

where H_{in} is a summation of energy penalties[13] $1_{\sigma_{i_k} \neq x_k} = \frac{1}{2}(1 - (-1)^{x_k + \sigma_{i_k}})$ with i_k the input wire associated with the k-th bit of x and $1_{\sigma_{i_j} \neq 0}$ with i_j the input wire associated with the j-th ancilla bit, H_{prop} a summation of penalties of the form $1_{\sigma_{i_k} \neq \sigma_{i_j}} + 1_{\sigma_{i_{k+1}} \neq 1 - \sigma_{i_j}\sigma_{i_\ell}}$ for each NAND gate mapping (i_j, i_ℓ) to (i_k, i_{k+1}), and H_{out} consists of a single penalty term $1_{\sigma_{i_T} \neq 1}$, for i_T the output wire. Then, $H_{\mathcal{C}}$ is such that there exists σ such that $H_{\mathcal{C}}(\sigma) = 0$ if and only if there is a witness w such that $\mathcal{C}$ accepts $(x, w, 0^r)$; otherwise, $H_{\mathcal{C}}(\sigma) \geq 1$ for all σ.

Note that $H_{\mathcal{C}}$ doesn't quite have the form (3.1) of an Ising spin Hamiltonian yet: some of its terms involve three variables at a time, and moreover not all terms are directly expressed as a function of the parity of the sum of two variables. With a little more work, using so-called "gadgets" it is possible to complete the reduction and find an $H'_{\mathcal{C}}$ that is equivalent to $H_{\mathcal{C}}$ (in terms of capturing the same decision problem) but is of the form (3.1) [6].

Using the trivial observation that P $\subseteq$ NP (any language that can be efficiently decided can be efficiently verified by ignoring the witness and performing the efficient decision procedure), it follows from the above discussion that the problem of deciding whether a classical circuit returns 1 on the all 0 input, which is complete for P, can be efficiently reduced to the problem of deciding whether an Ising Hamiltonian has a configuration with energy at most some threshold value a (that can be efficiently determined by following the steps of the reduction), or if all configurations have energy at least b, for some b such that $b - a = \Omega(1)$. (Explicitly, the Hamiltonian is obtained by adding penalty terms $1_{\sigma_{i_k} \neq 0}$ for all input wires associated to w to the Hamiltonian $H_{\mathcal{C}}$ obtained from the reduction described in the preceding paragraphs.)

Our next step is to devise a quantum analogue of this reduction.

3.2. **Quantum spin problems.** We can interpret the Hamiltonian H_{ising} introduced in (3.1) as an "energy functional", that associates an energy to any configuration σ. In quantum mechanics, a Hamiltonian is any linear operator on Hilbert space, with the restriction that the operator should be Hermitian (and bounded; for convenience here we only consider finite-dimensional spaces, so that the latter condition is automatic). The interpretation is that the Hamiltonian associates a definite energy $\lambda_i \in \mathbb{R}$ to any quantum state that happens to be in one of the Hamiltonian's eigenstates $|\phi_i\rangle$. The energy of an arbitrary state ρ is then computed as $\mathrm{Tr}(H\rho)$.[14] Often it is also required that the Hamiltonian be *local*, meaning that

[13] We use the notation 1_E for the indicator that event E occurs.

[14] The reader may have noticed that the syntactic requirements for "Hamiltonians" and "observables" are identical. Physically, a Hamiltonian is meant to represent a specific observable, that corresponds to the energy of a system; mathematically, the two notions are interchangeable.

H can be expressed as a sum of a polynomial number of terms h_i, each of which acts nontrivially on at most k qubits (i.e. each term can be expressed as the tensor product of the identity on $(n-k)$ qubits, and an arbitrary Hamiltonian on the remaining k qubits), for some constant k. This constraint reflects the fact that each term in the Hamiltonian is meant to represent a physical interaction between a small number of (typically spatially close) elementary particles.

Using the notation introduced in the previous section, the Ising spin Hamiltonian can be recast as a quantum Hamiltonian, $H'_{ising} = -\sum_{(i,j)} J_{ij}\, \sigma_Z^i \sigma_Z^j$, where σ_Z^i is shorthand for the observable that is σ_Z on the i-th qubit and the identity on the others, $\sigma_Z^i = \mathrm{Id}^{\otimes(i-1)} \otimes \sigma_Z \otimes \mathrm{Id}^{\otimes(n-i)}$. Since this Hamiltonian is diagonal in the computational basis, its eigenstate with smallest eigenvalue, also called its "ground state" or minimal energy state, is always attained at a pure computational basis state $|\sigma\rangle$, for some $\sigma \in \{0,1\}^n$.

Things get more interesting when we consider Hamiltonians made of a combination of non-commuting observables. Consider for example the 2-qubit Hamiltonian

$$H_{EPR} = -\frac{1}{2}(\sigma_X^1 \sigma_X^2 + \sigma_Z^1 \sigma_Z^2)\ . \tag{3.3}$$

As a matrix, this can be written as

$$H_{EPR} = \begin{pmatrix} -1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & -1 \end{pmatrix}\ .$$

What is remarkable about this Hamiltonian is that its smallest-eigenvalue eigenstate is a state

$$|\phi^+\rangle \;=\; \frac{1}{\sqrt{2}}\big(|0\rangle|0\rangle + |1\rangle|1\rangle\big) \;=\; \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}\ ,$$

also known as a "Bell state", or "EPR pair". This state has the property of being *entangled*: it cannot be expressed in the form $|\phi_1\rangle \otimes |\phi_2\rangle$, for any two single-qubit states $|\phi_1\rangle$ and $|\phi_2\rangle$.

The possibility for quantum Hamiltonians to force entanglement in their ground state distinguishes them from classical Hamiltonians, whose eigenstates are computational basis states, and in particular can be expressed as a tensor product of single-qubit states. As a result, while a classical Hamiltonian always has a minimal energy configuration that can be described using n bits (hence, as already observed, the problem of deciding its minimum energy is in NP, with the witness being a classical n-bit description of the minimal energy configuration), for quantum Hamiltonians this need not be the case. The complexity class QMA (for "Quantum Merlin-Arthur") is the quantum analogue of NP: QMA is the collection of all (promise) languages $L = L_{yes} \cup L_{no}$ such that it is possible for a quantum polynomial-time verifier to correctly decide whether an input $x \in L_{yes}$ or $x \in L_{no}$, with error at most $\frac{1}{3}$, with the help of a "quantum proof" $|\phi\rangle$ provided by an all-powerful, but untrusted, quantum prover. The problem of deciding the minimal energy of a local Hamiltonian to within some inverse polynomial precision is an example of a problem that is in QMA: informally, given a claimed minimum-eigenvalue eigenstate presented as a quantum state, it is possible to estimate the

associated eigenvalue by making the appropriate energy measurement. Kitaev established a quantum analogue of NP-completeness of 3-SAT by showing that the local Hamiltonian problem is QMA-complete, i.e. the constraints expressed by any polynomial-time quantum verification procedure can be reduced to constraints of the form checked by a local Hamiltonian. We give an overview of Kitaev's reduction next.

3.3. **Certificates for quantum computations.** In Section 3.1 we have seen that the computation carried out by a classical circuit can be represented as a "tableau", such that the property of being a valid tableau can be encoded in a classical Hamiltonian, thereby reducing the task of deciding whether a classical circuit accepts its input to the task of deciding whether the associated Hamiltonian (that can be efficiently computed from the circuit) has a small enough eigenvalue.

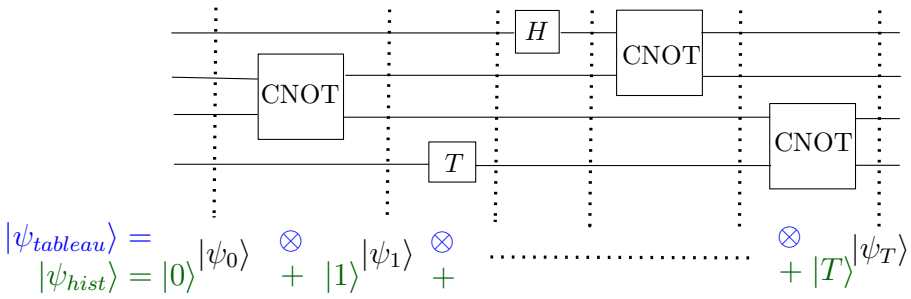

FIGURE 5. Two different ways to create a tableau from a quantum circuit. The state $|\psi_{tableau}\rangle$ is the tensor product of the state of the circuit at each time step. The state $|\psi_{hist}\rangle$ is their superposition, indexed by a clock register that goes from $|0\rangle$ to $|T\rangle$.

What is the correct notion of a tableau for quantum circuits? The first idea is to consider the juxtaposition of the quantum state of an ℓ-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 5) the property of being a valid "quantum tableau" cannot be enforced using a *local* Hamiltonian! 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 the physicist 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 5). 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

$$|\psi_{hist}\rangle \;=\; \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle|\psi_t\rangle \;. \tag{3.4}$$

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, in a manner very similar to what we did for classical tableaus: 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 $\{H|0\rangle, H|1\rangle\}$ 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 for the classical case by techniques from perturbation theory, it is possible to write the resulting Hamiltonian as a linear combination of local terms that all take the form of the EPR Hamiltonian (3.3). (Such a Hamiltonian is called a Hamiltonian "in XZ form", for obvious reasons.) The result is the following theorem from [14].

Theorem 3.1. *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 $\mathcal{C} = C_1 \cdots C_T$ 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*

$$H_{\mathcal{C}} \;=\; -\sum_{i,j} \frac{J_{ij}}{2}\left(\sigma_X^i \sigma_X^j + \sigma_Z^i \sigma_Z^j\right) , \tag{3.5}$$

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 3.2. It is possible to modify Theorem 3.1 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 3.1 can be adapted to show that the problem of estimating the minimal energy of a Hamiltonian of the form (3.5) is a QMA-complete problem.

Theorem 3.1 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. 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.

Such a verification protocol was introduced by Fitzsimons, Hadjušek and Morimae [16], and is summarized in Figure 6. In the protocol, the prover is required to prepare a smallest eigenstate of the Hamiltonian $H_{\mathcal{C}}$ given in (3.5). While it may not be immediately obvious at the level of our description, it is possible to prepare such a "history state" (3.4) 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 (3.5).

(1) The verifier initializes a counter γ to 0. She executes the following interaction with the prover independently $N = C/\delta^4$ 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 unpon reception. Let $b_{W,i} \in \{-1, 1\}$ be the outcome for the i-th qubit.
   (d) The verifier selects (i, j) uniformly at random among those pairs such that $J_{ij} \neq 0$. She updates her counter $\gamma \leftarrow \gamma - J_{ij} b_{W,i} b_{W,j}$.

(2) If $\gamma/N \leq a + \delta/2$ the verifier accepts the interaction. Otherwise, she rejects.

FIGURE 6. The Fitzsimons-Hadjuček-Morimae verification protocol.

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 $|\psi\rangle$ 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. The major achievement in Mahadev's work is a method to do just this; it is the topic of the next section.

4. Commitments

The key ingredient in Mahadev's verification protocol for quantum computations is a commitment scheme that allows a quantum prover to "commit" to a quantum state using only classical information, thereby providing the means to remove the need for quantum communication in the protocol described at the end of the previous section. Before introducing her commitment scheme, we review the classical notion of commitment and show how it can be implemented using collision-resistant hash functions.

4.1. Classical commitments.

Consider the following toy task of "coin-flipping over the phone" [8]: Alice is at work; Bob is at home; they would like to decide over the phone who will cook dinner tonight. Neither volunteers: they need to flip a coin. Clearly neither of them trusts the other to do this properly, so they need a protocol that makes it infeasible for either party to bias the outcome in their favor. Here is a way to achieve this using "commitments". Bob chooses a value $b \in \{0,1\}$ — ideally, he chooses it uniformly at random, but this is up to him. He then "commits" to b by sending Alice some information c — think of Bob as inserting a piece of paper with b written on it in a large safe, handing the safe to Alice, but keeping the key to himself. Then, Alice herself chooses a bit $a \in \{0,1\}$, and announces it directly to Bob. Finally, Bob reveals his bit b by giving Alice the "key" r to the safe. Alice uses r to open the safe and check Bob's claimed value for b. If the check goes through, they jointly agree that the bit $d = a \oplus b$ is unbiased. Finally, they use d to designate the night's cook-in-chief.

The properties of the coin-flipping protocol described in the previous paragraph —informally, that neither user has the ability to bias the outcome of the coin flip, provided the other user behaves honestly — suggest the following two requirements for the commitment scheme: it should be *hiding* (Alice does not learn b, unless Bob explicitly reveals it to her) and *binding* (once he has committed, Bob cannot "reveal" any value other than the one he committed to). Formally, a commitment scheme is defined as follows.

Definition 4.1 (Classical commitment scheme)**.** A (non-interactive) *commitment scheme* (for a message space M) is a triple of probabilistic polynomial-time procedures $(\textsc{Gen}, \textsc{Commit}, \textsc{Reveal})$ such that

- $k \leftarrow \textsc{Gen}(1^\lambda)$ generates a *key* k, when provided as input an integer λ written in unary.[15]

[15] λ is referred to as the *security parameter*. The reason it is provided in unary is that this guarantees, by the polynomial-time requirement on Gen, that the bit length of k is polynomial in the integer λ.

- COMMIT takes as input a key k and an $m \in M$ and returns a pair $(c, d) \leftarrow \text{COMMIT}_k(m)$ that consists of a *commitment value* c, and a *reveal value* d.[16]
- REVEAL takes as input a key k and a pair (c, d) and returns a value $m' \leftarrow \text{REVEAL}_k(c, d)$ such that $m' \in M \cup \{\perp\}$, where $\perp$ is a special "failure" symbol.
- *(Correctness:)* For any $m \in M$, if $(c, d) \leftarrow \text{COMMIT}_k(m)$ then it holds that $\text{REVEAL}_k(c, d) = m$.
- *(Statistical hiding:)* for any two $m \neq m' \in M$ the distributions of $c \leftarrow \text{COMMIT}_k(m)$ and $c' \leftarrow \text{COMMIT}_k(m')$ are identical.
- *(Computational binding:)* No probabilistic polynomial-time procedure $\mathcal{A}$ can, given k as input, return a triple (c, d, d') such that $m \leftarrow \text{REVEAL}_k(c, d)$ and $m' \leftarrow \text{REVEAL}_k(c, d')$ are such that $m \neq m' \in M$ with non-negligible probability.

The definition refers to a scheme that is *statistically* hiding and *computationally* binding. It is possible to consider schemes with the qualifiers inverted, i.e. computationally hiding and statistically binding, but we will not make use of such a scheme here. Note that, under the assumption that the scheme is statistically hiding, the binding property can only be required to hold under a computational assumption. This is because if the distribution of the commitment value c does not depend on whether the message $m = 0$ or $m = 1$, then for any c it is possible to find d_0 and d_1 that reveal to $m' = 0$ and $m' = 1$ respectively. The computational binding requirement is that, even though these values must exist, they should be hard to find.

We end this section by presenting a construction of a commitment scheme whose computational binding property rests on the assumption that there exists a family of *collision-resistant hash functions* (CRHF). A CRHF is a family of functions $\{f_{k(\lambda)} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$, for polynomially bounded functions $k(\lambda)$, $n(\lambda)$, $m(\lambda)$ of λ, such that given $k(\lambda)$ for any x the value $f_k(x)$ can be evaluated efficiently, but it is impossible for any probabilistic polynomial-time adversary $\mathcal{A}$ to find a pair of inputs $x \neq x'$ such that $f_k(x) = f_k(x')$ with non-negligible probability. Collision-resistant hash functions are widely used in cryptography, and many constructions are known based on assumptions such as the Diffie-Hellman assumption about hardness of the discrete logarithm problem or the Learning with Errors problem about hardness of solving noisy linear equations. (*Unconditionally* proving the existence of a CRHF would imply that P$\neq$NP,[17] so we have to rely on computational assumptions.) For reasons that will become clear when we discuss the extension to quantum commitments, it is convenient for us to make the additional assumption that the functions f_k are 2-to-1. Specifically,

> *Assumption (2TO1):* For each $k = k(\lambda)$, both functions $f_{k,0} : r \mapsto f_k(0\|r)$ and $f_{k,1} : r \mapsto f_k(1\|r)$ are injective, and they have the same range.

In Section 7 we sketch a construction of a CRHF family that is *almost* 2-to-1, in a sense that we will make precise, based on the learning with errors problem.

Let $\{f_k : \{0,1\}^n \to \{0,1\}^{n/2}\}$ be a 2-to-1 CRHF family (this is also called a "claw-free" family, where any triple (r_0, r_1, y) such that $f_k(0\|r_0) = f_k(1\|r_1) = y$

[16] We write the input key k to COMMIT as a subscript that is often omitted for convenience.

[17] The converse implication is not known to hold.

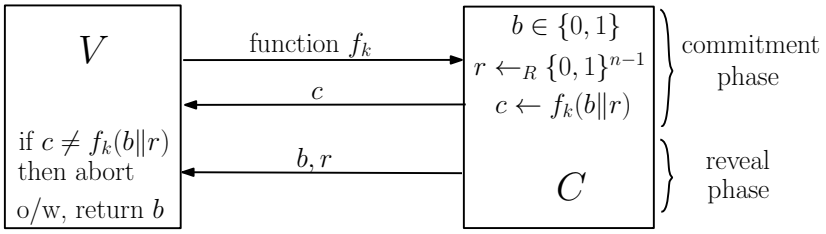

FIGURE 7. A computationally binding commitment protocol based on the use of a CRHF family $\{f_{k(\lambda)}\}_\lambda$. We refer to the party performing the commitment as the *committer*, and the party receiving the commitment as the *verifier*. The symbol $x \leftarrow_R S$ means that x is selected uniformly at random from the finite set S.

forms a claw). Here is a commitment scheme based on f_k (see also Figure 7). The scheme allows commitments of single-bit messages, $M = \{0,1\}$. Both parties agree on a security parameter $\lambda \geq 1$. To commit to a bit $b \in M$, the committer selects a uniformly random $r \in \{0,1\}^{n-1}$ and sends $c = f_k(b\|r)$ to the verifier, where the symbol $\|$ is used to denote string concatenation. To reveal b, the committer sends both b and r to the verifier, who checks that $c = f_k(b\|r)$. This scheme is computationally binding, because to "change his mind" the committer needs to identify r_0 and r_1 such that $f_k(0\|r_0) = f_k(1\|r_1)$, which is a collision. And the scheme is statistically hiding, because due to the 2-to-1 property any commitment c from the committer has exactly one preimage under $f_{k,0}$ and one preimage under $f_{k,1}$, that the verifier has no basis to prefer over each other.

4.2. **Quantum commitment schemes.** We now devise an analogue of Definition 4.1 that allows for committing to quantum states. Keeping in mind the goal of using such a commitment scheme for classical verification, we wish to keep the actions of the verifier classical. It is then no longer meaningful to expect that the scheme would allow the committer to "reveal" its quantum state to the verifier. Instead of revealing the quantum state itself, we settle for the ability for the committer to reveal *measurement outcomes* performed on the state to which it has committed to. Importantly, we should allow the verifier to request measurement outcomes in a choice of at least two incompatible bases, so that the committer couldn't simply have measured its state first and then classically committed to the measurement outcome. Intuitively, the "binding" requirement of the scheme should be that it is impossible for the committer to report measurement outcomes that are obtained by measuring a state that depends on the measurement basis requested by the verifier. For example, it should be impossible for the committer to "commit" to a state ρ such that it later has the ability to both "reveal" to the verifier that a measurement of ρ in the computational, or in the Hadamard, basis yields an outcome $+1$ — as indeed, there is no such state.

We proceed with a formal definition that mimics Definition 4.1 with these desiderata in mind.

Definition 4.2 (Qubit commitment scheme)**.** A *qubit commitment scheme* for a pair of observables (X, Z) on a Hilbert space $\mathcal{H}_{\mathsf{B}}$ is a triple of classical probabilistic polynomial-time procedures $(\textsc{Gen}, \textsc{Open}_Z, \textsc{Open}_X)$ and a triple of quantum polynomial-time procedures $(\textsc{Commit}, \textsc{Meas}_Z, \textsc{Meas}_X)$ such that

- $k \leftarrow \textsc{Gen}(1^\lambda)$ generates a *key* k, when provided as input an integer λ written in unary.
- $\textsc{Commit}$ is a quantum polynomial-time procedure
$$\textsc{Commit}_k : \mathcal{H}_{\mathsf{B}} \to \mathcal{H}_{\mathsf{C}} \otimes \mathcal{H}_{\mathsf{B}'}$$
such that for any density matrix ρ_{BR} on $\mathcal{H}_{\mathsf{B}} \otimes \mathcal{H}_{\mathsf{R}}$, where R is an arbitrary reference system, the state $(\textsc{Commit}_k \otimes \mathrm{Id}_{\mathsf{R}})(\rho_{\mathsf{BR}})$ is classical on C (i.e. the first output register, C, has been measured). We call the post-measurement state on B' the *post-commitment state*.
- $\textsc{Meas}_Z$ and $\textsc{Meas}_X$ are quantum measurements on B' that return classical strings d_Z and d_X as their output, respectively.
- $\textsc{Open}_Z$ and $\textsc{Open}_X$ take as input a pair (c, d_Z) and (c, d_X) and return a value $a_Z \in \{0,1\} \cup \{\perp\}$ and $a_X \in \{0,1\} \cup \{\perp\}$ respectively.
- *(Correctness:)* For any state ρ_{BR}, the distribution of
$$a_Z \leftarrow \big(\textsc{Open}_Z \otimes \mathrm{Id}_{\mathsf{R}}\big) \circ \big(\textsc{Meas}_Z \otimes \mathrm{Id}_{\mathsf{CR}}\big) \circ \big(\textsc{Commit}_k \otimes \mathrm{Id}_{\mathsf{R}}\big)(\rho) \tag{4.1}$$
is statistically indistinguishable from the distribution obtained by measuring the register B of ρ_{BR} using observable Z.[18] A similar condition holds for the combination of $\textsc{Meas}_X$ and $\textsc{Open}_X$.
- *(Statistical hiding:)* For every ρ_{BR} and ρ'_{BR}, the distributions of c and c' obtained from $\textsc{Commit}_k(\rho)$ and $\textsc{Commit}_k(\rho')$ respectively are statistically indistinguishable.
- *(Computational binding:)* For any triple of quantum polynomial-time procedures $(\textsc{Commit}^*, \textsc{Meas}^*_Z, \textsc{Meas}^*_X)$ and efficiently preparable state ρ^*_{BR} such that the values a^*_Z and a^*_X defined as in (4.1) (with $\textsc{Commit}^*$ and $\textsc{Meas}^*$ replacing $\textsc{Commit}$ and $\textsc{Meas}$ respectively) are different from $\perp$ with non-negligible probability, there exists a state σ_{B} such that the distribution of $(-1)^{a^*_Z}$ and $(-1)^{a^*_X}$, conditioned on $a^*_Z \neq \perp$ or $a^*_X \neq \perp$ respectively, are computationally indistinguishable from the outcome distribution obtained by directly measuring σ_{B} using Z and X respectively.

Although the definition may at first seem complicated, it closely parallels Definition 4.1 of a classical commitment scheme. The main difference is that a quantum commitment scheme allows the committer to commit to a quantum state ρ in such a way that the verifier may request measurement outcomes obtained by measuring the state ρ using two different observables, Z and X. The observables Z and X may not be simultaneously measurable, so that it is not possible to implement

[18] Statistical indistinguishability between two families of distributions $\{D_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{D'_\lambda\}_{\lambda\in\mathbb{N}}$ means that there does not exist a procedure that, given as input 1^λ as well as a sample from either D_λ or D'_λ, can determine which is the case with a success probability that is larger than $\frac{1}{2} + \varepsilon(\lambda)$, for some non-negligible function $\varepsilon(\lambda)$. This is equivalent to the statistical distance between the two distributions being at most $\varepsilon(\lambda)$. Computational indistinguishability is defined similarly, except that the distinguishing procedure is restricted to run in probabilistic polynomial time. For distributions on a single bit, computational indistinguishability implies statistical indistinguishability, but this is no longer the case for distributions over larger domains.

a quantum commitment scheme by providing classical commitments to measurement outcomes in both bases simultaneously. The most important guarantee of the scheme is the binding property, that ensures that any committer is tied to reporting outcomes for both types of measurement requests that could, at least in principle, have been obtained from a single quantum state. For example, for the case of a single qubit commitment and Z and X corresponding to measurements in the computational and Hadamard basis respectively, it should not be possible to generate a commitment string c such that it is later possible to find d_Z and d_X that both yield $a_Z = 0$ and $a_Z = 1$ with high probability, as Heisenberg's uncertainty principle rules out the existence of a single-qubit state that yields the outcome 0 when measured in both the computational and the Hadamard basis.

A few additional remarks on the definition are in order. A first remark is that the definition does not restrict the dimension of the system B, so that it allows committing to a multiple-qubit state. However, we have restricted the reported outcomes to be from a pair of two-outcome observables only. It is possible to formulate a more general definition, that allows more than two measurements with outcomes in a larger set than $\{0, 1\}$. In fact, such a commitment scheme is eventually needed for Mahadev's protocol. For the sake of simplicity, we gave the simpler definition.

A second remark is that the definition is formulated as a "stand-alone" security definition, meaning that it does not attempt to guarantee security in a broader context where e.g. multiple commitments would be obtained in sequence, or the committer's post-measurement states could be used as part of a subsequent cryptographic procedure, etc. Giving a "universally composable" definition, as has been done for the case of classical commitments [13], would be desirable to make quantum commitments useful in a broader context; giving such a definition falls outside the scope of these notes (and is not needed for the analysis of Mahadev's protocol).

5. Constructing quantum commitments

We proceed to describe the key innovation that underlies Mahadev's protocol, a quantum commitment scheme that satisfies the properties of Definition 4.2.[19] To get started we consider the most naïve scheme, that consists in applying the commitment scheme described in Section 4.1 directly, in superposition, to a quantum state. For simplicity we consider the case of committing to a single-qubit state, when the observables Z and X are the single-qubit Pauli observables σ_Z and σ_X.

5.1. Key generation.

Let $n, m \geq 1$ be integer and

$$\left\{ f_{k(\lambda)} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)} \right\}_{\lambda \in \mathbb{N}}$$

a family of functions that satisfies assumption (2TO1) stated in Section 4.1. The public key $k \leftarrow \textsc{Gen}(1^\lambda)$ for the scheme contains a description of the function f_k that allows one to evaluate it on any input. For simplicity, throughout this section we fix λ, and often drop the index k, writing f for a function chosen according to the procedure Gen. In the coming subsections we progressively formulate requirements on f that are stronger than those of a 2-to-1 CRHF.

[19] As already hinted at, her scheme satisfies more, and in particular allows commitments to more than two measurements, each with more than two outcomes. Here we concentrate on the simplest non-trivial variant, that already demonstrates (almost) all the interesting features.

5.2. **Commitment procedure.** We start by describing the procedure $\textsc{Commit}_k$, for the case where the committer wishes to commit to a single-qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. As a first step, the committer evaluates the function f in superposition by creating an additional register that contains a uniform superposition over all $(n-1)$-bit strings and then computing the commitment in superposition into a third ancilla register initialized to $|0\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \mapsto \big(\alpha|0\rangle + \beta|1\rangle\big)\Big(\frac{1}{\sqrt{2^{n-1}}}\sum_{r\in\{0,1\}^{n-1}}|r\rangle\Big)|0^n\rangle$$

$$\mapsto |\psi'\rangle = \frac{1}{\sqrt{2^{n-1}}}\sum_{r\in\{0,1\}^{n-1}}\alpha|0\rangle|r\rangle|f(0\|r)\rangle + \beta|1\rangle|r\rangle|f(1\|r)\rangle\ . \tag{5.1}$$

The committer then measures the last register in the computational basis to obtain a classical commitment string $c \in \{0,1\}^m$. The string c is the outcome returned by the procedure $\textsc{Commit}_k$. The remaining two registers, the first qubit and the register containing r, form the register B' that contains the post-commitment state.

At this point it is worth noting explicitly that the (2TO1) property is crucial in ensuring that the post-measurement state retains the same "structure" as the original state $|\psi\rangle$ that was committed to, i.e. the commitment procedure does not "damage" the quantum state to which it is applied. Recall from Section 2.1 that in general any measurement induces a collapse of the state. Informally, the reason that this does not happen here is that since the classical commitment scheme is statistically hiding, the commitment string reveals no information about the committed bit, so that measuring the commitment can be done without perturbing the committed state. Thus when the commitment string is measured, the state $|\psi'\rangle$ in (5.1) *partially* collapses to the post-measurement state consistent with the outcome obtained,

$$|\psi''\rangle = \big(\alpha|0\rangle|r_0\rangle + \beta|1\rangle|r_1\rangle\big)|c\rangle\ . \tag{5.2}$$

We pause for a moment and argue that the state $|\psi''\rangle$ is a plausible "commitment" to $|\psi\rangle$, and in particular why the operations performed so far may yield a scheme that has the binding property. Observe that $|\psi''\rangle$ is very similar to $|\psi\rangle$, except that the initial superposition that defined $|\psi\rangle$ got slightly "muddled" by the inclusion of r_0 and r_1. Nevertheless, morally the superposition is preserved: most importantly, the coefficients α, β that define it are unchanged. In fact, the state $|\psi''\rangle$ is unitarily related to $|\psi\rangle$, by the simple unitary

$$U_c: \quad |0\rangle|r_0\rangle \mapsto |0\rangle|0\rangle\ , \quad |1\rangle|r_1\rangle \mapsto |1\rangle|0\rangle \tag{5.3}$$

(extended in an arbitrary way to the whole space). However, although this unitary exists, it may not be easy for the prover to implement it! This is because doing so seems to require the identification of both r_0 and r_1 from c, which would require the prover to find a collision for f. (See Figure 8.)

Note that we haven't made the requirement that f should be collision-resistant yet. Even though it is only needed for the analysis of a multi-qubit commitment scheme, for completeness we introduce an assumption on f that is known as *collapsing* and is a stronger property than being collision resistant.

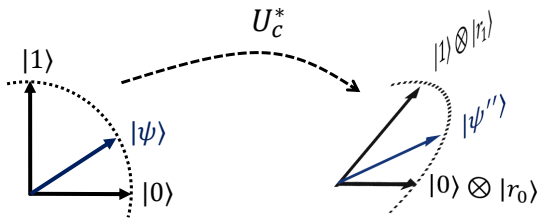

FIGURE 8. An illustration of the commitment procedure. On the left, the state $|\psi\rangle$ is expressed in the known basis $(|0\rangle, |1\rangle)$. On the right, the basis is an unknown (to the committer) pair of orthonormal vectors $(|0\rangle \otimes |r_0\rangle, |1\rangle \otimes |r_1\rangle)$ in a high-dimensional space.

Assumption (C): Consider the following abstract game between an arbitrary " prover" and a trusted (quantum) "challenger".[20] First, the prover is 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. The prover hands the state $|\phi\rangle$ over to the challenger, who evaluates f in superposition on $|\phi\rangle$ and measures the image register, obtaining a c in the range of f and the (suitably normalized) post-measurement state $|\phi'\rangle = \sum_{x:f(x)=c} \alpha_x |x\rangle$. The challenger returns to the prover 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 prover wins if it correctly guesses which is the case. Assumption (C) on the function f states that no quantum polynomial-time prover can succeed in this game with probability non-negligibly larger than $\frac{1}{2}$.

The reason that Assumption (C) implies collision resistance is that, if the function were not collision resistant, the prover 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 prover guesses at random.

5.3. **Statistical hiding.** The statistical hiding property is straightforward, and follows from the $(2TO1)$ property as in the case of the classical commitment scheme: any measurement outcome c that the prover could obtain is such that $c = f(0\|r_0) = f(1\|r_1)$ for some $r_0, r_1 \in \{0,1\}^{n-1}$, and in fact the distribution of c is uniform over the range of the function f.

5.4. **Revealing measurement outcomes in the computational basis.** We describe the procedures MEAS_Z and OPEN_Z together. Recall that after having performed the commitment procedure, the committer's state takes the form of $|\psi''\rangle$ in (5.2). The measurement MEAS_Z simply consists in a measurement of the

[20]This game is not meant to be executed in the protocol; rather, it is meant to indicate a task that the committer, impersonating the "prover", should *not* be able to complete.

first two registers of $|\psi''\rangle$ in the computational basis. This results in an outcome $d_Z = (b, r_b) \in \{0,1\} \times \{0,1\}^{n-1}$ that the committer reports to the verifier.

The verifier's procedure $\textsc{Open}_Z$ is defined as follows. Given c and $d_Z = (b, r_b)$, the verifier first checks that $b\|r_b$ is a preimage of c under f, i.e. that $f(b\|r_b) = c$. We refer to this test as the *preimage test*. In case the test fails, the verifier sets $a_Z \leftarrow \perp$. (Note that this test is identical to the test performed by the verifier in the reveal phase of the classical commitment protocol described in Section 4.1.) In case the test succeeds, the verifier sets $a_Z \leftarrow b$.

The completeness property for this pair of procedures is clear. Given a state of the form (5.2) that has been obtained by the correct application of the procedure $\textsc{Commit}_k$, a measurement of the first n qubits always yields a pair (b, r_b) such that $f(b\|r_b) = c$. Moreover, the distribution of the bit b is $\Pr(b = 0) = |\alpha|^2$, $\Pr(b = 1) = |\beta|^2$, which is exactly the distribution of a measurement of the qubit $|\psi\rangle$ in the computational basis.

5.5. **Revealing measurement outcomes in the Hadamard basis.** We turn to the procedures $\textsc{Meas}_X$ and $\textsc{Open}_X$. The measurement $\textsc{Meas}_X$ consists in a measurement of the first two registers of $|\psi''\rangle$ in the Hadamard basis. This results in an outcome $d_X = (u, t) \in \{0,1\} \times \{0,1\}^{n-1}$ that the committer reports to the verifier.

The verifier's procedure $\textsc{Open}_X$ is defined as follows. Given c and $d_X = (u, t)$ the verifier first computes the two preimages $0\|r_0$ and $1\|r_1$ of c under f. For this to be possible we make the following assumption on the function f:

> *Assumption (T):* There is "trapdoor information" such that, given the trapdoor, it is possible to efficiently invert f, i.e. given c, recover r_0, r_1 such that $f(0\|r_0) = f(1\|r_1) = c$.

Functions that are easy to compute and hard to invert, but easy to invert given a trapdoor, are widely used in cryptography; they form the bedrock of public-key cryptography. We will say more about how such a trapdoor can be "planted" in an efficiently computable function in Section 7.[21] For the time being, we proceed with the description of $\textsc{Open}_X$. Having determined r_0 and r_1, the verifier sets

$$a_X \leftarrow u \oplus t \cdot (r_0 \oplus r_1) \in \{0,1\} \,, \tag{5.4}$$

except if $t = 0^{n-1}$, in which case the verifier sets $a_X \leftarrow \perp$ (the motivation for this condition will become clear later, when we discuss Assumption (HC)).

The completeness property of this pair of procedures can be verified by direct calculation. To provide some intuition for the expression (5.4), we verify completeness for the case when $\alpha = \beta = \frac{1}{\sqrt{2}}$, i.e. $|\psi\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Applying the H gate on the first n qubits of the state $|\psi''\rangle$ in (5.2) yields (omitting the last register,

[21] Technically the use of a trapdoor requires a modification of the $\textsc{Gen}$ procedure, so that it is allowed to return a (public key, private key) pair such that the public key is given to the committer and allows evaluation of f, while the private key is given to the verifier only, and allows inversion of f.

that contains $|c\rangle$)

$$H^{\otimes n}|\psi''\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{b\in\{0,1\}} \Big(\sum_{u\in\{0,1\}} (-1)^{ub}|u\rangle \Big) \otimes \Big(\sum_{t\in\{0,1\}^{n-1}} \big((-1)^{t\cdot r_0} + (-1)^{t\cdot r_1}\big)|t\rangle \Big)$$
$$= \frac{1}{\sqrt{2^{n-1}}} \sum_{u\in\{0,1\}, t\in\{0,1\}^{n-1}} 1_{t\cdot r_0 = u\oplus(t\cdot r_1)}|u\rangle|t\rangle \ .$$

Thus in this case, for any outcome (u,t) that can be obtained with non-zero probability by the committer's procedure $\textsc{Meas}_X$, the verifier's decoded bit is $a_X = u \oplus t\cdot(r_0 \oplus r_1) = 0$, which agrees with the outcome of a measurement of the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ in the Hadamard basis. (Moreover, the probability of obtaining $t = 0^{n-1}$ is exponentially small, so that the committer has only an exponentially small chance of leading to an abort.) Informally, the addition of $t\cdot(r_0\oplus r_1)$ to the bit u acts as a "decoding" operation that accounts for interference created by the strings r_0, r_1 that have been appended to the prover's qubit in the commitment phase.

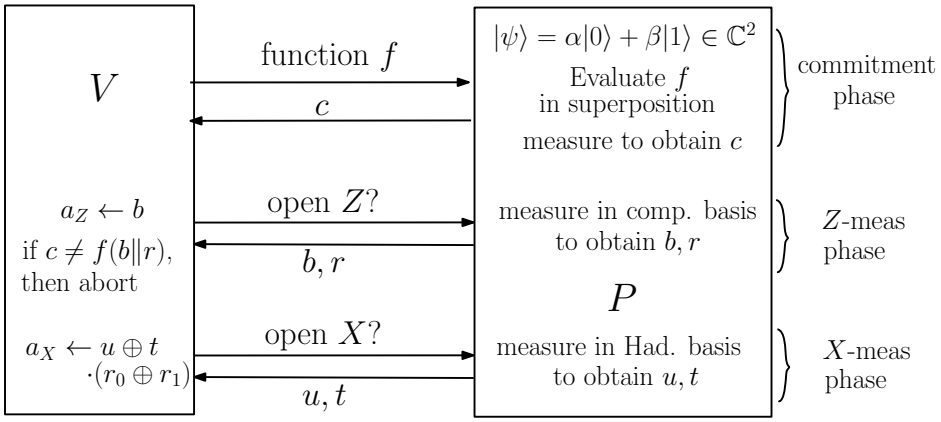

FIGURE 9. Committing to a qubit.

5.6. **The committed qubit.** We have defined all procedures that underlie the qubit commitment scheme, and verified that these procedures satisfy the required completeness requirement, as well as the property of statistical hiding. (The protocol is summarized in Figure 9.) It remains to show the computational binding property, which is the heart of the matter. The property requires to show the existence of a state σ having certain properties in regard to the outcomes recorded by the verifier in the protocol. In this section, we define the state σ, that we refer to as the "commited qubit". In the next section we argue the required properties.

Note that we have to be careful with the meaning that we ascribe to any definition of a "committed qubit". Indeed, the attentive reader will have observed that it is straightforward for a classical committer to "succeed" in the commitment protocol, by selecting an arbitrary (b, r_b), setting $c = f(b\|r_b)$, and sending $d_Z = (b, r_b)$ and d_X a uniformly random n-bit string when asked to do so. Indeed, that this is possible should come as no surprise: it exactly corresponds to the honest behavior for a committer desiring to commit to the "classical" qubit $|b\rangle$! In fact, one may

consider the fact that the commmitment scheme can be implemented by a classical verifier, whenever the information to be committed to is classical, as a useful feature of the scheme.

In general, the actions of an arbitrary committer can be modeled by two measurement procedures $\textsc{Meas}_Z^*$ and $\textsc{Meas}_X^*$ such that each measurement has outcomes that range in the set of n-bit strings. Up to a change of basis for the committer's post-commitment state we may assume without loss of generality that the measurement $\textsc{Meas}_Z^*$ consists of a measurement of the first n qubits of the prover's post-commitment state in the computational basis. In other words, we may fix a basis in which the post-commitment state can be expressed as

$$|\tilde{\psi}\rangle = \sum_{b\in\{0,1\},r\in\{0,1\}^{n-1}} \tilde{\alpha}_{b,r}\,|b\rangle|r\rangle|\phi_{b,r}\rangle\ , \tag{5.5}$$

for arbitrary coefficients $\tilde{\alpha}_{b,r}$ and normalized states $|\phi_{b,r}\rangle$, and such that moreover it holds that $\textsc{Meas}_Z^* = \textsc{Meas}_Z$.[22]

Having fixed a basis, the measurement $\textsc{Meas}_X^*$ can in general be expressed as the composition of an arbitrary unitary V acting on the entirety of the committer's space, followed by a measurement of the first n qubits in the Hadamard basis, i.e. $\textsc{Meas}_X^* = V^*(\textsc{Meas}_X \otimes \mathrm{Id})V$. Using that $\{\mathrm{Id}, \sigma_X, \sigma_Z, \sigma_X\sigma_Z\}$ form a basis for the space of linear operators on $\mathbb{C}^2$, any such unitary can be uniquely expanded according to its action on the first qubit as

$$V \;=\; \mathrm{Id}\otimes V_I + \sigma_X \otimes V_X + \sigma_Z \otimes V_Z + \sigma_X\sigma_Z \otimes V_{XZ}\ .$$

With this notation it is not hard to verify that the linear map defined by

$$\tilde{V} : |b\rangle|\phi\rangle \mapsto \big(|b\rangle V_I|\phi\rangle + (-1)^b|b\rangle V_Z|\phi\rangle\big)|0\rangle + \big(|b\rangle V_X|\phi\rangle + (-1)^b|b\rangle V_{XZ}|\phi\rangle\big)|1\rangle\ , \tag{5.6}$$

where $b \in \{0,1\}$ and $|\phi\rangle$ is arbitrary, is an isometry, hence is an admissible operation in quantum mechanics.[23] (Note that $\tilde{V}$ increases dimension by a factor 2. The new qubit register, in third position, is called a "purifying system".) We are ready to make a key definition.

Definition 5.1 (Committed qubit)**.** Given a commitment string $c \in \{0,1\}^m$ and an arbitrary post-commitment state for the prover of the form (5.5), let σ be the single-qubit state obtained from $|\tilde{\psi}\rangle$ by performing the following operations:

(i) Apply the isometry $\tilde{V}$ defined in (5.6) to $|\tilde{\psi}\rangle$, yielding a state on $(\mathbb{C}^2)^{\otimes n}\otimes\mathcal{H}'$ (for some $\mathcal{H}'$).
(ii) Measure qubits 2 to n in the Hadamard basis, obtaining an outcome string $t \in \{0,1\}^{n-1}$.
(iii) If $t\cdot(r_0 \oplus r_1) = 1 \bmod 2$, apply a σ_Z operator to the first qubit. Here $0\|r_0$ and $1\|r_1$ are the two preimages of the commitment string c under f.
(iv) Return the first qubit.

[22] We have not entirely justified why this assumption is without loss of generality. It requires the use of the Stinespring dilation theorem, that guarantees that any measurement with n-bit outcomes can be expressed, up to isometry, as a measurement of n qubits in the computational basis.

[23] For the expert, $\tilde{V}$ is obtained from V by a σ_Z-twirl, followed by a conditional σ_X bit-flip. The motivation for this definition will become clear in the proof of the computational binding property.

Note that the verifier does not know the state σ; in fact, strictly speaking σ is not present on the committer's space at any time. The point is that σ exists, and is well-defined (mathematically) as a function only of the commitment string c, the post-commitment state $|\tilde{\psi}\rangle$, and the unitary V.

5.7. **Computational binding.** The discussion in the preceding section establishes a definition of a state σ, that may not exist directly on the provers' space at any time in the protocol, but that is explicitly defined from states and operators that are a function of the committer's. To establish the binding property, it remains to show that the distribution of outcomes obtained by measuring σ in the computational and Hadamard bases is computationally indistinguishable from the distribution of outcomes computed by the verifier using the $\textsc{Open}_Z$ and $\textsc{Open}_X$ procedures respectively, under the assumption that the procedures lead to non-$\perp$ outcomes with high probability.

We start with the case of the computational basis, that is easier. Recall that we made a choice of basis for the committer's space such that its post-commitment state is of the form $|\tilde{\psi}\rangle$ in (5.5), and moreover the measurement $\textsc{Meas}_Z^* = \textsc{Meas}_Z$ measures the first n qubits of $|\tilde{\psi}\rangle$ in the computational basis and returns the outcome (b, r), from which the verifier obtains $a_Z = b$ (provided the preimage test succeeds).

According to Definition 5.1, the committed qubit σ is defined from $|\tilde{\psi}\rangle$ by applying the isometry $\tilde{V}$ in (5.6) and returning the first qubit, to which may have been applied a σ_Z operator, depending on the string t. We make two observations. First, note that $\tilde{V}$ has a block-diagonal form: it stabilizes the spaces $|0\rangle \otimes \mathcal{H}''$ and $|1\rangle \otimes \mathcal{H}''$, where $\mathcal{H}'' = (\mathbb{C}^2)^{\otimes(n-1)} \otimes \mathcal{H}'$ is the Hilbert space associated with all but the committer's first qubit. As a result the outcome of a measurement of the first qubit of $|\tilde{\psi}\rangle$, or of $\tilde{V}|\tilde{\psi}\rangle$, in the computational basis, are identically distributed. Second, the σ_Z operator has no effect on a measurement in the computational basis. These two observations together establish that the verifier's outcome $a_Z = b$ has exactly the right distribution.

Before we consider measurements in the Hadamard basis, we make a simplifying assumption. Note that an honest committer, whose state $|\tilde{\psi}\rangle$ is the state $|\psi''\rangle$ in (5.2), always passes the preimage test. For the remainder of the analysis we make the assumption that, in case the verifier decides to execute the $\textsc{Open}_Z$ procedure, the committer's measurement procedure $\textsc{Meas}_Z^*$ *always* returns a pair (b, r) that passes the preimage test.[24] As a consequence of this assumption the expression for the state $|\tilde{\psi}\rangle$ simplifies to

$$|\tilde{\psi}\rangle = \sum_{b\in\{0,1\}} \tilde{\alpha}_b \, |b\rangle|r_b\rangle|\phi_b\rangle \;, \tag{5.7}$$

where r_0 and r_1 are such that $f(0\|r_0) = f(1\|r_1) = c$, since using Assumption (2TO1) all other (b, r) would be rejected by the preimage test.

Our task now is to argue that the verifier's bit a_X, as obtained from the $\textsc{Open}_X$ procedure described in (5.4), has a distribution that is computationally indistinguishable from that of a Hadamard measurement of the committed qubit.

[24]This assumption requires that in any execution of the commitment protocol there is a positive probability that the verifier executes the preimage test. In the case of Mahadev's verification protocol, each of the two reveal phases is chosen with probability 1/2, so that a simple reduction allows us to make the assumption without loss of generality.

We start from the distribution of a_X. Recall from Section 2.1 that given a quantum state $\tilde{\rho} = |\tilde{\psi}\rangle\langle\tilde{\psi}|$ the expectation value of an observable O is given by $\mathrm{Tr}(O\tilde{\rho}) = \langle\tilde{\psi}|O|\tilde{\psi}\rangle$. Here, using $\textsc{Meas}_X^* = V^*(\textsc{Meas}_X \otimes \mathrm{Id})V$, the observable O is obtained by first applying V, followed by a measurement in the Hadamard basis of all qubits but the first to obtain the string t (this corresponds to applying the projection $|t\rangle\langle t|$), then a σ_Z bit-flip on the first qubit, as a function of the outcome t obtained (this corresponds to applying the unitary $\sigma_Z^{t\cdot(r_0\oplus r_1)}$), and finally a measurement of the first qubit in the Hadamard basis (this corresponds to the observable σ_X). As a result, the expectation value of $(-1)^{a_X}$ can be expressed as

$$\mathrm{E}\big[(-1)^{a_X}\big] = \sum_{t\in\{0,1\}^{n-1}} \langle\tilde{\psi}|V^\dagger H^{\otimes n}\big((\sigma_X^{t\cdot(r_0\oplus r_1)}\sigma_Z\sigma_X^{t\cdot(r_0\oplus r_1)})\otimes|t\rangle\langle t|\big)H^{\otimes n}V|\tilde{\psi}\rangle\ ,$$

where for later convenience we have used $\sigma_X = H\sigma_Z H$ to rewrite the observable applied on the first qubit.

We now turn to the expectation value of $(-1)^b$, where b is the outcome of a measurement of the committed qubit σ in the Hadamard basis. Using the definition of the committed qubit this can be expressed in a similar fashion, except that due to the use of the isometry $\tilde{V}$ in the definition of σ the unitary V has been conjugated by a random σ_Z operator:[25]

$$\begin{aligned}\mathrm{E}\big[(-1)^b\big] &= \frac{1}{2}\mathrm{E}\big[(-1)^{a_X}\big]\\ &+\frac{1}{2}\sum_{t\in\{0,1\}^{n-1}} \langle\tilde{\psi}|\sigma_Z V^\dagger\sigma_Z H^{\otimes n}\big((\sigma_X^{t\cdot(r_0\oplus r_1)}\sigma_Z\sigma_X^{t\cdot(r_0\oplus r_1)})\otimes|t\rangle\langle t|\big)H^{\otimes n}\sigma_Z V\sigma_Z|\tilde{\psi}\rangle\\ &= \frac{1}{2}\mathrm{E}\big[(-1)^{a_X}\big]\\ &-\frac{1}{2}\sum_{t\in\{0,1\}^{n-1}} \langle\tilde{\psi}|\sigma_Z V^\dagger H^{\otimes n}\big((\sigma_X^{t\cdot(r_0\oplus r_1)}\sigma_Z\sigma_X^{t\cdot(r_0\oplus r_1)})\otimes|t\rangle\langle t|\big)H^{\otimes n}V\sigma_Z|\tilde{\psi}\rangle\ ,\end{aligned}$$

where all σ_Z operators act on the first qubit, and for the second line we used $\sigma_Z H = H\sigma_X$ to commute the innermost σ_Z all the way to the middle, where we simplified $\sigma_X\sigma_Z\sigma_X = -\sigma_Z$. Taking the difference between the two terms, simplifying the middle expression using anti-commutation, and cancelling out cross-terms gives

$$\begin{aligned}&\Big|\mathrm{E}\big[(-1)^{a_X}\big] - \mathrm{E}\big[(-1)^b\big]\Big|\\ &= \frac{1}{2}\Big|\sum_{t\in\{0,1\}^{n-1}} (-1)^{t\cdot(r_0\oplus r_1)}\big(\langle 0,r_0,\phi_0|V^\dagger H^{\otimes n}\big(\sigma_Z\otimes|t\rangle\langle t|\big)H^{\otimes n}V|0,r_0,\phi_0\rangle\\ &\qquad + \langle 1,r_1,\phi_1|V^\dagger H^{\otimes n}\big(\sigma_Z\otimes|t\rangle\langle t|\big)H^{\otimes n}V|1,r_1,\phi_1\rangle\big)\Big|\ ,\end{aligned} \tag{5.8}$$

where to write this last expression we used the assumption that the state $|\tilde{\psi}\rangle$ can be expressed as in (5.7), i.e. that the committer succeeds in the verifier's preimage test with probability 1. To argue that the right-hand side of (5.8) cannot be large, we make the following final assumption.

[25]For the second part of the state on the right-hand side of (5.6), corresponding to the last qubit being in state $|1\rangle$, there is also a missing "σ_X" operator on the first qubit, labeled $|b\rangle$. A σ_X has no effect on the outcome of a measurement in the Hadamard basis, so it can be ignored here.

> *Assumption (HC):* No quantum polynomial-time procedure can, given as input a description of f, return a quadruple (c, r, u, t) such that $f(b\|r) = c$ for some $b \in \{0,1\}$, $t \neq 0^{n-1}$, and $u = t \cdot (r_0 \oplus r_1)$, where r_0, r_1 are the two preimages of c, with probability non-negligibly larger than $\frac{1}{2}$.

If the expression on the right-hand side of (5.8) were non-negligible, there would be a violation of Assumption (HC): a quantum polynomial-time "adversary" (to the assumption) could simulate the prover to prepare $|\tilde{\psi}\rangle$ and measure the first n qubits register to obtain (b, r_b). It would then apply the unitary V and measure the first n qubits in the Hadamard basis to obtain a string (u, t). Finally, the adversary would return the tuple (c, r, u, t); (5.8) exactly measures the correlation of the bit u with the correct value $t \cdot (r_0 \oplus r_1)$. Note that the requirement that $t \neq 0^{n-1}$ is necessary for the assumption to be reasonable, as for $t = 0^{n-1}$ the value $u = 0$ is easy to determined. The adversary described above obtains $t = 0^{n-1}$ with probability $2^{-(n-1)}$.

Thus Assumption (HC) guarantees that the expression in (5.8) is negligibly small, completing the proof of the binding property for the case of a Hadamard basis measurement.

5.8. **Summary.** The preceding sections have introduced a quantum commitment scheme that allows a committer to commit to a single-qubit state in a way that is perfectly hiding (this guarantees to the committer that committing does not leak information), and later reveal the outcome of a measurement of the qubit in the computational or Hadamard basis in a way that is computationally binding (this guarantees to the verifier that the outcomes a_Z or a_X obtained from the committer's reported strings d_Z or d_X respectively are consistent with measurements on a single state).

The use of a quantum commitment scheme in Mahadev's verification protocol, to be described in the next section, requires the prover to commit to an n'-qubit state. This is done by requiring n' commitment strings $c_1, \ldots, c_{n'}$. There are still only two possible measurements: the first reports the n'-bit outcome obtained by measuring the n'-qubit state in the computational basis, and the second does the same for the Hadamard basis. The construction and analysis of a quantum commitment scheme that accommodates this is very similar to the construction and analysis given in this section, except that the proof of the computational binding property explicitly requires the assumption (C) introduced in Section 5.2.

Our description would not be satisfying if we did not discuss assumptions (2TO1), (C), (T), and (HC). Are these assumptions reasonable? While (2TO1) and (T) are fairly standard assumptions in classical cryptography, for which it is possible to find multiple constructions, Assumption (C) is less common (though it has been used in different contexts in quantum cryptography), and Assumption (HC) is even less usual (though it can be seen as a strengthening of a more standard "(non-adaptive) hardcore bit property"). In Section 7 we sketch a construction of a function family satisfying all four assumption simultaneously, based on the computational hardness of the "Learning with Errors" problem in cryptography.

6. Verifying quantum computations

In Section 3 we reduced the problem of verifying the outcome of an arbitrary polynomial time quantum computation to the following decision problem.

> *Input:* An integer n', the description of an n'-qubit Hamiltonian H in XZ form,
>
> $$H_{\mathcal{C}} = -\sum_{1\leq i<j\leq n} \frac{J_{ij}}{2}\left(\sigma_X^i\sigma_X^j + \sigma_Z^i\sigma_Z^j\right), \tag{6.1}$$
>
> a real number a, and $\delta > 0$.
> *Promise:* The smallest eigenvalue of H is either less than a, or at least $a+\delta$.
> *Decision:* Accept if and only if the smallest eigenvalue of H is less than a. (We refer to such H as "YES" instances.)

The reduction guarantees that the "promise gap" δ can be taken to be at least some inverse polynomial in n'. For ease of exposition we further assume that all coefficients J_{ij} lie in $\{-1,0,1\}$, and that $a = -\sum |J_{ij}|$. In physical language this corresponds to a "frustration-free" Hamiltonian, meaning that in the case of a YES instance an eigenstate with smallest eigenvalue a is also an eigenstate with eigenvalue J_{ij} of each of the local terms $h_{ij} = \frac{1}{2}(\sigma_X^i\sigma_X^j + \sigma_Z^i\sigma_Z^j)$. (This last assumption *is* with loss of generality, as it is not hard to see that the resulting problem lies in NP; nevertheless, we make it because it helps simplify the presentation without hiding any interesting steps.)

Informally, Mahadev's protocol for deciding this problem combines the protocol described in Figure 6 in Section 3 with the quantum commitment scheme introduced in Section 5. This leads to the protocol summarized in Figure 10. (The protocol we present requires a polynomial number of rounds. We chose to do so to simplify the analysis. Mahadev's protocol can be executed in two rounds only, as stated in Theorem 2.3.)

In order to prove Theorem 2.3 we need to verify the completeness and soundness properties required of an argument system. The completeness property is straightforward. Assuming H has an eigenstate $|\psi\rangle$ with eigenvalue at most a, the prover can commit to that eigenstate and honestly report measurement outcomes. In this case, the verifier will never abort. Her counter γ is a random variable such that the probability of γ/N being larger than $a+\delta/2$ can be bounded by a small constant, less than $\frac{1}{3}$, using Hoeffding's inequality and assuming that the constant C is chosen large enough.

To establish the soundness property we first observe that we may assume that the prover's actions in, say, a fraction at least $1-\delta^2$ of the rounds, is such that, if $W = Z$ is chosen by the verifier in that round, then the probability of the verifier's reported outcome d_Z leading to $a_Z = \perp$ is very small, smaller than δ^2. Indeed, if this is not the case then the probability that the verifier obtains $a_Z = \perp$ at least once throughout the entire protocol will be at least $2/3$, which would immediately establish the soundness property.

Under that assumption, for those rounds we may make use of the computational binding property of the quantum commitment scheme, that establishes the existence of an n-qubit committed state ρ that underlies the distribution of measurement outcomes a_Z or a_X obtained by the verifier in those rounds. By assumption of a NO instance, for any such state the distribution of outcomes obtained when measuring

Let λ be a security parameter, $\mathcal{C}$ a quantum circuit provided as input, and $H_{\mathcal{C}}$ the n'-qubit Hamiltonian obtained from $\mathcal{C}$ as in (3.5). Let $(\textsc{Gen}, \textsc{Open}_Z, \textsc{Open}_X)$ be the verifier's procedures in a quantum commitment scheme such that the measurements Z and X correspond to n'-bit outcome n'-qubit measurements in the computational and Hadamard bases respectively. Let $(\textsc{Commit}, \textsc{Meas}_Z, \textsc{Meas}_Z)$ be the committer's procedures in the same scheme.

(1) The verifier initializes a counter γ to 0. She executes the following interaction with the prover independently $N = C/\delta^4$ times, where C is a large enough universal constant:
   (a) The verifier generates a (public key, private key) pair $(pk, sk) \leftarrow \textsc{Gen}(1^\lambda)$, and sends the public key pk to the prover. She keeps the secret key sk private.
   (b) The prover creates an eigenstate $|\psi\rangle$ of H with smallest eigenvalue. Let $\rho = |\psi\rangle\langle\psi|$ be the associated density matrix. The prover executes $c \leftarrow \textsc{Commit}_{pk}(\rho)$ and sends c to the verifier.
   (c) The verifier selects a measurement $W \in \{X, Z\}$ uniformly at random, and asks the prover to reveal measurement outcomes in the basis W.
   (d) The prover performs the measurement $\textsc{Meas}_W$ on the post-commitment state, and returns the outcome d_W to the verifier.
   (e) The verifier computes $a_W \leftarrow \textsc{Open}_W(c, d_W)$. If $a_W = \perp$ the verifier aborts the protocol. Otherwise, a_W is an n-bit string. The verifier selects (i, j) uniformly at random among those pairs such that $J_{ij} \neq 0$. She updates her counter $\gamma \leftarrow \gamma - J_{ij}(-1)^{a_{W,i}+a_{W,j}}$.
(2) If $\gamma/N \leq a + \delta/2$ the verifier accepts the interaction. Otherwise, she rejects.

FIGURE 10. The Mahadev verification protocol.

in the computational or Hadamard basis must, on average, lead to a value for the verifier's variable γ that averages, for those rounds, to at least $a + \delta - O(\delta^2)$. Taking into account those rounds where we do not have control of the committer (because its behavior has a larger probability of leading to an abort), the probability that the verifier's final count γ/N is lower than $a + \delta/2$ can be made less than $\frac{1}{3}$ provided that the constant C is chosen large enough.

The argument completes the proof sketch for the main theorem exposed here, Theorem 2.3. In the next section we conclude by outlining a construction of a family of functions that satisfies the requirements for implementing the quantum commitment scheme described in Section 5, under the learning with errors assumption.

7. A construction based on the learning with errors problem

In Section 5 we have identified four assumptions on a family of functions $\{f_{k(\lambda)} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{\lambda \in \mathbb{N}}$ that are sufficient for the resulting quantum commitment scheme to be computationally binding. Can the four 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 [9], and a mild modification of it is used in Mahadev's protocol. Even

though the assumptions introduced in the Section 5 will not all be strictly satisfied by the construction, it is possible to verify that the protocol itself remains sound.

7.1. **The LWE problem.** Our starting point is the *Learning with Errors* (LWE) problem, introduced by Regev [28]. 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 [10], attribute-based encryption [19], program obfuscation [32, 20], traitor tracing [21], 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 for the verification protocol presented 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 error distribution χ over $\mathbb{Z}_q$. 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)* 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).

We include a few words of explanation for the reader unaccustomed with the notion of computational indistinguishibility between ensembles of distributions. 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.[26] 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. To the reader with a geometric mind, it might help to picture a discrete lattice (all integer linear combinations of the columns of A, as a subset of $\mathbb{R}^m$) such that to each lattice point is added a little noise, in the form of a discrete Gaussian distribution with small

[26]Computational hardness only makes sense as the input size goes to infinity, which is why to be precise we should consider a family of distributions, parametrized by an integer λ, and argue that the samples become harder and harder to distinguish as $\lambda \to \infty$.

variance centered at the lattice point. Even though all the Gaussian "blobs" thus obtained are well separated, given a point in any one of them, it is (conjecturally) hard to recover the center of the blob, i.e. the closest lattice vector.

We comment briefly on the choice of parameters. 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 [26].

7.2. **Construction.** To specify the function f we 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 (HC)).

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.[27] 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 [25]. 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 $1/(\sqrt{mn\log q})$,[28] are sampled. The public information is $(A, y = As + e)$. The private information is the pair (R, s).

Next, we discuss how the function can be evaluated, given the public parameters (A, y). We define two functions f_0, f_1 that should be understood as $f(0\|\cdot)$ and $f(1\|\cdot)$ respectively. 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 $\mathbb{Z}_2^{wn}$ for $w = \lceil \log q \rceil$) and returns $Ax + e' + by$, 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 6. A formal way around this is to think of f_b as the function that returns not $Ax + e' + by$, but the *distribution* of $Ax + e' + by$, 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, i.e. Assumption (2TO1). Injectivity

[27] 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.

[28] The precise choice of α is delicate, and the parameters given here should only be treated as indicative; we refer to [9, Section 8] for the right setting of parameters.

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 $Ax+e'+by$ (with high probability over the choice of e'). To make the function ranges overlap, we need the distribution of $Ax+e'$ to have the same support as the distribution of $Ax'+e'+y = A(x'+s)+(e'+e)$. 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 (2TO1). Property (T) follows from the existence of the secret information (R, s). Given a $b \in \{0,1\}$ and an element $c = Ax + e' + by = A(x+bs) + (e'+be)$ 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 c under f_b.

The two remaining properties, the collapsing property (C) and the hardcore bit property (HC), require more work, and we refer to [9] for a detailed exposition. We remark that the two properties are not entirely new. Property (C) was been 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 [31]. Similar "hardcore bit" properties to (HC) have been shown for many LWE-based cryptographic schemes (see e.g. [4]). Usually the property states that "for any vector $t \in \mathbb{Z}_q^n \backslash \{0\}$, the value $t \cdot s \in \mathbb{Z}_q$ is indistinguishable from uniform, even given a sample $(A, As+e)$". Our property (HC) is subtly stronger, in that the adversary may choose the vector t itself, possibly as a function of the sample $(A, As+e)$. An additional difficulty stems from the specific "bit" that the adversary predicts in our setting. In the definition of Assumption (HC) this bit is the value $u = t \cdot (r_0 \oplus r_1)$, where r_0, r_1 are the *binary representation* of the two preimages in $\mathbb{Z}_q^n$, x_0 and $x_1 = x_0 - s$, of the prover's commitment string $c \in \mathbb{Z}_q^m$. (Recall that the use of the binary representation came from the requirements on the honest prover, that is asked to perform a measurement in the Hadamard basis, yielding a binary string of outcomes.) It is in order to complete the argument showing that a procedure that returns the information asked for in Assumption (HC), i.e. the quadruple (c, r, u, t), can be turned into a procedure that breaks the decisional LWE assumption, that we need to assume that the secret vector s is a binary vector. The result is a somewhat roundabout construction that we may hope will be simplified in future work.

Acknowledgments. I am indebted to Urmila Mahadev for numerous conversations that helped clarify her work. I thank Victor Albert, Alexandru Georghiu, Urmila Mahadev and Oded Regev for comments on earlier versions of these notes.

References

1. Scott Aaronson and Andris Ambainis, *Forrelation: A problem that optimally separates quantum from classical computing*, SIAM Journal on Computing **47** (2018), no. 3, 982–1038.
2. Dorit Aharonov, Micahel Ben-Or, and Elad Eban, *Interactive Proofs For Quantum Computations*, Arxiv preprint arXiv:0810.5375 (2008).

3. Dorit Aharonov and Ayal Green, *A quantum inspired proof of $P^{\#p} \subseteq IP$*, arXiv preprint arXiv:1710.09078 (2017).
4. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan, *Simultaneous hardcore bits and cryptography against memory attacks*, Theory of Cryptography Conference, Springer, 2009, pp. 474–495.
5. László Babai, *Trading group theory for randomness*, Proceedings of the seventeenth annual ACM symposium on Theory of computing, ACM, 1985, pp. 421–429.
6. Francisco Barahona, *On the computational complexity of ising spin glass models*, Journal of Physics A: Mathematical and General **15** (1982), no. 10, 3241.
7. Hannes Bernien, Sylvain Schwartz, Alexander Keesling, Harry Levine, Ahmed Omran, Hannes Pichler, Soonwon Choi, Alexander S Zibrov, Manuel Endres, Markus Greiner, et al., *Probing many-body dynamics on a 51-atom quantum simulator*, Nature **551** (2017), no. 7682, 579.
8. Manuel Blum, *Coin flipping by telephone a protocol for solving impossible problems*, ACM SIGACT News **15** (1983), no. 1, 23–27.
9. Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick, *Certifiable randomness from a single quantum device*, arXiv preprint arXiv:1804.00640 (2018).
10. Zvika Brakerski and Vinod Vaikuntanathan, *Efficient fully homomorphic encryption from (standard) lwe*, SIAM Journal on Computing **43** (2014), no. 2, 831–871.
11. Gilles Brassard, David Chaum, and Claude Crépeau, *Minimum disclosure proofs of knowledge*, Journal of Computer and System Sciences **37** (1988), no. 2, 156–189.
12. Anne Broadbent, Joseph F. Fitzsimons, and Elham Kashefi, *Universal blind quantum computation*, Arxiv preprint arXiv:0807.4154 (2008).
13. Ran Canetti and Marc Fischlin, *Universally composable commitments*, Annual International Cryptology Conference, Springer, 2001, pp. 19–40.
14. Toby Cubitt and Ashley Montanaro, *Complexity classification of local hamiltonian problems*, SIAM Journal on Computing **45** (2016), no. 2, 268–316.
15. Richard P Feynman, *Simulating physics with computers*, International journal of theoretical physics **21** (1982), no. 6-7, 467–488.
16. Joseph F Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae, *Post hoc verification of quantum computation*, Physical review letters **120** (2018), no. 4, 040501.
17. Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi, *Verification of quantum computation: An overview of existing approaches*, arXiv preprint arXiv:1709.06984 (2017).
18. Shafi Goldwasser, Silvio Micali, and Charles Rackoff, *The knowledge complexity of interactive proof systems*, SIAM Journal on computing **18** (1989), no. 1, 186–208.
19. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee, *Attribute-based encryption for circuits*, Journal of the ACM (JACM) **62** (2015), no. 6, 45.
20. Rishab Goyal, Venkata Koppula, and Brent Waters, *Lockable obfuscation*, Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on, IEEE, 2017, pp. 612–621.
21. ———, *Collusion resistant traitor tracing from learning with errors*, Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, ACM, 2018, pp. 660–670.
22. Joe Kilian, *A note on efficient zero-knowledge proofs and arguments*, Proceedings of the twenty-fourth annual ACM symposium on Theory of computing, ACM, 1992, pp. 723–732.
23. Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan, *Algebraic methods for interactive proof systems*, Journal of the ACM (JACM) **39** (1992), no. 4, 859–868.
24. Urmila Mahadev, *Classical verification of quantum computations*, arXiv preprint arXiv:1804.01082 (2018).
25. Daniele Micciancio and Chris Peikert, *Trapdoors for lattices: Simpler, tighter, faster, smaller*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2012, pp. 700–718.
26. Chris Peikert et al., *A decade of lattice cryptography*, Foundations and Trends® in Theoretical Computer Science **10** (2016), no. 4, 283–424.
27. Ran Raz and Avishay Tal, *Oracle separation of BQP and PH*, Electronic Colloquium on Computational Complexity (ECCC), vol. 25, 2018, p. 107.
28. Oded Regev, *On lattices, learning with errors, random linear codes, and cryptography*, Journal of the ACM (JACM) **56** (2009), no. 6, 34.
29. Ben W Reichardt, Falk Unger, and Umesh Vazirani, *Classical command of quantum systems*, Nature **496** (2013), no. 7446, 456.

30. A. Shamir, *IP= PSPACE*, Journal of the ACM (JACM) **39** (1992), no. 4, 869–877.
31. Dominique Unruh, *Computationally binding quantum commitments*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2016, pp. 497–527.
32. Daniel Wichs and Giorgos Zirdelis, *Obfuscating compute-and-compare programs under LWE*, 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2017, pp. 600–611.

California Institute of Technology, Pasadena CA 91106, USA
E-mail address: vidick@cms.caltech.edu