Verifier-on-a-Leash: new schemes for verifiable delegated quantum computation, with quasilinear resources

Andrea Coladangelo[1], Alex B. Grilo[2], Stacey Jeffery[2], and Thomas Vidick[1]

[1] Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, USA. CMS, Caltech, Pasadena, USA
{acoladan, vidick}@cms.caltech.edu
[2] QuSoft and CWI, Amsterdam, the Netherlands
{alexg, jeffery}@cwi.nl

Abstract. The problem of reliably certifying the outcome of a computation performed by a quantum device is rapidly gaining relevance. We present two protocols for a classical verifier to verifiably delegate a quantum computation to two non-communicating but entangled quantum provers. Our protocols have near-optimal complexity in terms of the total resources employed by the verifier and the honest provers, with the total number of operations of each party, including the number of entangled pairs of qubits required of the honest provers, scaling as $O(g \log g)$ for delegating a circuit of size g. This is in contrast to previous protocols, whose overhead in terms of resources employed, while polynomial, is far beyond what is feasible in practice. Our first protocol requires a number of rounds that is linear in the depth of the circuit being delegated, and is blind, meaning neither prover can learn the circuit or its input. The second protocol is not blind, but requires only a constant number of rounds of interaction.
Our main technical innovation is an efficient rigidity theorem which allows a verifier to test that two entangled provers perform measurements specified by an arbitrary m-qubit tensor product of single-qubit Clifford observables on their respective halves of m shared EPR pairs, with a robustness that is independent of m. Our two-prover classical-verifier delegation protocols are obtained by combining this rigidity theorem with a single-prover quantum-verifier protocol for the verifiable delegation of a quantum computation, introduced by Broadbent.

1 Introduction

Quantum computers hold the potential to speed up a wide range of computational tasks (see, for example, [Mon16]). Recent progress towards implementing limited quantum devices has added urgency to the already important question of how a classical verifier can test a quantum device. This verifier could be an experimentalist running a new experimental setup; a consumer who has purchased a purported quantum device; or a client who wishes to delegate some task to a quantum server. In all cases, the user would like to exert some form of control over the quantum device. For example, the experimentalist may think that she is testing that a particular experiment prepares a certain quantum state by performing a series of measurements, i.e. by state tomography, but this assumes some level of trust in the measurement apparatus being used.

For a classical party to truly test a quantum system, that system should be modeled in a device-independent way, having classical inputs (e.g. measurement settings) and classical outputs (e.g. measurement results).

Tests of quantum mechanical properties of a system first appeared in the form of Bell tests [Bel64,CHSH69]. In a Bell test, a verifier asks classical questions to a quantum-device and receives classical answers. These tests make one crucial assumption on the system to be tested: that it consists of two spatially isolated components that are unable to communicate throughout the experiment. One can then upper bound the value of some statistical quantity of interest subject to the constraint that the two devices do not share any entanglement. Such a bound is referred to as a Bell inequality. While the violation of a Bell inequality can be seen as a certificate of entanglement, the area of self-testing, first introduced in [MY04], allows for the certification of much stronger statements, including about which measurements are being performed, and on which state. Informally, a *robust rigidity theorem* is a statement about which kind of apparatus, quantum state and measurements, must be used by a pair of isolated devices in order to succeed in a given statistical test. Following a well-established tradition, we will refer to such tests as *games*, call the devices *players* (or *provers*), and the quantum state and measurements that they implement the *strategy* of the players. A rigidity theorem is a statement about the necessary structure of near-optimal strategies for a game.

In 2012, Reichardt, Unger and Vazirani proved a robust rigidity theorem for playing a sequence of n CHSH games [RUV13]. Aside from its intrinsic interest, this rigidity theorem had two important consequences. One was the first device-independent protocol for quantum key distribution. The second was a protocol whereby a completely classical verifier can test a universal quantum computer consisting of two non-communicating devices. The resulting protocol for delegating quantum computations has received a lot of attention as the first classical-verifier delegation protocol. The task is well-motivated: for the foreseeable future, making use of a quantum computer will likely require delegating the computation to a potentially untrusted cloud service, such as that announced by IBM [Cas17].

Unfortunately, the complexity overhead of the delegation protocol from [RUV13], in terms of both the number of EPR pairs needed for the provers and the overall time complexity of the provers as well as the (classical) verifier, while polynomial, is prohibitively large. Although the authors of [RUV13] do not provide an explicit value for the exponent, in [HPDF15] it is estimated that their protocol requires resources that scale like $\Omega(g^{8192})$, where g is the number of gates in the delegated circuit (notwithstanding the implicit constant, this already makes the approach thoroughly impractical for even a 2-gate circuit!). The large overhead is in part due to a very small (although still inverse polynomial) gap between the completeness and soundness parameters of the rigidity theorem; this requires the verifier to perform many more Bell tests than the actual number of EPR pairs needed to implement the computation, which would scale linearly with the circuit size.

Subsequent work has presented significantly more efficient protocols for achieving the same, or similar, functionality [McK16,GKW15,HPDF15]. We refer to Table 1 for a summary of our estimated lower bounds on the complexity of each of these results

(not all papers provide explicit bounds, in which case our estimates, although generally conservative, should be taken with caution). Prior to our work, the best two-prover delegation protocol required resources scaling like g^{2048} for delegating a g-gate circuit. Things improve significantly if we allow for more than two provers, however, the most efficient multi-prover delegation protocols still required resources that scale as at least $\Omega(g^4 \log g)$ for delegating a g-gate circuit on n qubits. Since we expect that in the foreseeable future most quantum computations will be delegated to a third-party server, even such small polynomial overhead is unacceptable, as it already negates the quantum advantage for a number of problems, such as quantum search.

The most efficient classical-verifier delegation protocols known [FH15,NV17], with $\mathrm{poly}(n)$ and 7 provers, respectively, require resources that scale as $O(g^3)$, but this efficiency comes at the cost of a technique of "post-hoc" verification. In this technique, the provers must learn the verifier's input even before they are separated, so that they can prepare the history state for the computation.[1] As a result, these protocols are not blind[2]. Moreover, while the method does provide a means for verifying the outcome of an arbitrary quantum computation, in contrast to [RUV13] it does not provide a means for the verifier to test the provers' implementation of the required circuit on a gate-by-gate basis. Other works, such as [HH16], achieve two-prover verifiable delegation with complexity that scales like $O(g^4 \log g)$, but in much weaker models; for example, in [HH16] the provers' private system is assumed a priori to be in tensor product form, with well-defined registers. General techniques are available to remove the strong assumption, but they would lead to similar large overhead as previous results.

In contrast, in the setting where the verifier is allowed to have some limited quantum power, such as the ability to generate single-qubit states and measure them with observables from a small finite set, efficient schemes for blind verifiable delegation do exist [ABE10,FK17,Mor14,Bro18,HM15,MF16,FH17,MTH17] (see also [Fit16] for a recent survey). In this case, only a single prover is needed, and the most efficient *single-prover quantum-verifier* protocols can evaluate a quantum circuit with g gates in time $O(g)$. The main reason these protocols are much more efficient than the classical-verifier multi-prover protocols is that they avoid the need for directly testing any of the qubits used by the prover, instead requiring the trusted verifier to directly either prepare or measure the qubits used for the computation.

New rigidity results. We overcome the efficiency limitations of multi-prover delegation protocols by introducing a new robust rigidity theorem. Our theorem allows a classical verifier to certify that two non-communicating provers apply a measurement associated with an arbitrary m-qubit tensor product of single-qubit Clifford observables on their respective halves of m shared EPR pairs. This is the first result to achieve self-testing for such a large class of measurements. The majority of previous works in self-testing have been primarily concerned with certifying the state and were limited to simple

[1] Using results of Ji [Ji16], this allows the protocol to be single-round. Alternatively, the state can be created by a single prover and teleported to the others with the help of the verifier, resulting in a two-round protocol.

[2] *Blindness* is a property of delegation protocols, which informally states that the prover learns nothing about the verifier's private circuit.

	Provers	Rounds	Total Resources	Blind
RUV 2012 [RUV13]	2	$\text{poly}(n)$	$\geq g^{8192}$	yes
McKague 2013 [McK16]	$\text{poly}(n)$	$\text{poly}(n)$	$\geq 2^{153} g^{22}$	yes
GKW 2015 [GKW15]	2	$\text{poly}(n)$	$\geq g^{2048}$	yes
HDF 2015 [HPDF15]	$\text{poly}(n)$	$\text{poly}(n)$	$\Theta(g^4 \log g)$	yes
Verifier-on-a-Leash Protocol (Section 4)	2	$O(\text{depth})$	$\Theta(g \log g)$	yes
Dog-Walker Protocol (Section 5)	2	$O(1)$	$\Theta(g \log g)$	no

Table 1: Resource requirements of various delegation protocols in the multi-prover model. We use n to denote the number of qubits and g the number of gates in the delegated circuit. "depth" refers to the depth of the delegated circuit. "Total Resources" refers to the gate complexity of the provers, the number of EPR pairs of entanglement needed, and the number of bits of communication in the protocol. To ensure fair comparison, each protocol is required to produce the correct answer with probability 99%. For all protocols except our two new protocols, this requires a polynomial number of sequential repetitions, which is taken into account when computing the total resources.

single-qubit measurements in the X-Z plane. Prior self-testing results for multi-qubit measurements only allow to test for tensor products of σ_X and σ_Z observables. While this is sufficient for verification in the post-hoc model of [FH15], testing for σ_X and σ_Z observables does not directly allow for the verification of a general computation (unless one relies on techniques such as process tomography [RUV13], which introduce substantial additional overhead).

Our first contribution is to extend the "Pauli braiding test" of [NV17], which allows to test tensor products of σ_X and σ_Z observables with constant robustness, to allow for σ_Y observables as well. This is somewhat subtle due to an ambiguity in the complex phase that cannot be detected by any classical two-player test; we formalize the ambiguity and show how it can be effectively accounted for. Our second contribution is to substantially increase the set of elementary gates that can be tested, to include arbitrary m-qubit tensor products of single-qubit Clifford observables. This is achieved by introducing a new "conjugation test", which tests how an observable applied by the provers acts on the Pauli group. The test is inspired by general results of Slofstra [Slo16], but is substantially more direct.

A key feature of our rigidity results is that their robustness scales independently of the number of EPR pairs tested, as in [NV17]. This is crucial for the efficiency of our delegation protocols. The robustness for previous results in parallel self-testing typically had a polynomial dependence on the number of EPR pairs tested. We give an informal statement of our robust rigidity theorem.

Theorem 1 (Informal). *Let $m \in \mathbb{Z}_{>0}$. Let $\mathcal{G}$ be a fixed, finite set of single-qubit Clifford observables. Then there exists an efficient two-prover test* $\text{RIGID}(\mathcal{G}, m)$ *with $O(m)$-bit questions (a constant fraction of which are of the form $W \in \mathcal{G}^m$) and answers such that the following properties hold:*

- *(Completeness) There is a strategy for the provers that uses $m+1$ EPR pairs and succeeds with probability at least $1-e^{-\Omega(m)}$ in the test.*
- *(Soundness) For any $\varepsilon > 0$, any strategy for the provers that succeeds with probability $1-\varepsilon$ in the test must be $\mathrm{poly}(\varepsilon)$-close, up to local isometries, to a strategy in which the provers begin with $(m+1)$ EPR pairs and is such that upon receipt of a question of the form $W \in \mathcal{G}^m$ the prover measures the "correct" observable W.*

Although we do not strive to obtain the best dependence on ε, we believe it should be possible to obtain a scaling of the form $C\sqrt{\varepsilon}$ for a reasonable constant C. We discuss the test in Section 3. The complete analysis can be found in the full version of the paper.

New delegation protocols. We employ the new rigidity theorem to obtain two new efficient two-prover classical-verifier protocols in which the complexity of verifiably delegating a g-gate quantum circuit solving a BQP problem scales as $O(g \log g)$.[3]

We achieve our protocols by adapting the efficient single-prover quantum-verifier delegation protocol introduced by Broadbent [Bro18] (we refer to this as the "EPR protocol"), which has the advantage of offering a direct implementation of the delegated circuit, in the circuit model of computation and with very little modification needed to ensure verifiability, as well as a relatively simple and intuitive analysis.

Our first protocol is blind, and requires a number of rounds of interaction that scales linearly with the depth of the circuit being delegated. The second protocol is not blind, but only requires a constant number of rounds of interaction with the provers. Our work is the first to propose verifiable two-prover delegation protocols that overcome the prohibitively large resource requirements of all previous multi-prover protocols, requiring only a quasilinear amount of resources, in terms of number of EPR pairs and time. However, notwithstanding our improvements, a physical implementation of verifiable delegation protocols remains a challenging task for the available technology.

We introduce the protocols in more detail. The protocols provide different methods to delegate the quantum computation performed by the quantum verifier from [Bro18] to a second prover (call him PV for Prover V). The rigidity test is used to verify that the second prover indeed performs the same actions as the honest verifier, which are sequences of single-qubit measurements of Clifford observables from the set $\Sigma = \{X, Y, Z, F, G\}$ (where F and G are defined in (2)).

In the first protocol, one of the provers plays the role of Broadbent's prover (call him PP for Prover P), and the other plays the role of Broadbent's verifier (PV). As PV just performs single-qubit and Bell-basis measurements, universal quantum computational power is not needed for this prover. The protocol is divided into two sub-games; which game is played is chosen by the verifier by flipping a biased coin with appropriately chosen probabilities.

- The first game is a sequential version of the rigidity game $\textsc{rigid}(\Sigma, m)$ (from Theorem 1) described in Figure 9. This aims to enforce that PV performs precisely the right measurements;

[3] The $\log g$ overhead is due to the complexity of sampling from the right distribution in rigidity tests. We leave the possibility of removing this by derandomization for future work. Another source of overhead is in achieving blindness: in order to hide the circuit, we encode it as part of the input to a universal circuit, introducing a factor of $O(\log g)$ overhead.

– The second game is the delegation game, described in Figures 6, 7, and 8, and whose structure is summarized in Figure 4. Here the verifier guides PP through the computation in a similar way as in the EPR Protocol.

We remark that in both sub-games, the questions received by PV are of the form $W \in \Sigma^m$, where $\Sigma = \{X, Y, Z, F, G\}$ is the set of measurements performed by the verifier in Broadbent's EPR protocol. The questions for PV in the two sub-games are sampled from the same distribution. This ensures that the PV is not able to tell which kind of game is being played. Hence, we can use our rigidity result of Theorem 1 to guarantee honest behavior of PV in the delegation sub-game. We call this protocol *Verifier-on-a-Leash Protocol*, or "leash protocol" for short.

The protocol requires $(2d + 1)$ rounds of interaction, where d is the depth of the circuit being delegated (see Section 2.3 for a precise definition of how this is computed). The protocol requires $O(n + g)$ EPR pairs to delegate a g-gate circuit on n qubits, and the overall time complexity of the protocol is $O(g \log g)$. The input to the circuit is hidden from the provers, meaning that the protocol can be made blind by encoding the circuit in the input, and delegating a universal circuit. We note that using universal circuits incurs a $\log n$ factor increase in the depth of the circuit [BFGH10].

The completeness of the protocol follows directly from the completeness of [Bro18]. Once we ensure the correct behavior of PV using our rigidity test, soundness follows from [Bro18] as well, since the combined behavior of our verifier and an honest PV is nearly identical to that of Broadbent's verifier.

The second protocol also starts from Broadbent's protocol, but modifies it in a different way to achieve a protocol that only requires a constant number of rounds of interaction. The proof of security is slightly more involved, but the key ideas are the same: we use a combination of our new self-testing results and the techniques of Broadbent's protocol to control the two provers, one of which plays the role of Broadbent's verifier, and the other the role of the prover. Because of the more complicated "leash" structure in this protocol, we call it the *Dog-Walker Protocol*. Like the leash protocol, the Dog-Walker Protocol has overall time complexity $O(g \log g)$. Unlike the leash protocol, the Dog-Walker protocol is not blind. In particular, while PV and PP would have to collude after the protocol is terminated to learn the input in the leash protocol, in the Dog-Walker protocol, PV simply receives the input in clear.

Based on the Dog-Walker Protocol, it is possible to design a classical-verifier two-prover protocol for all languages in QMA. This is achieved along the same lines as the proof that QMIP = MIP* from [RUV13]. The first prover, given the input, creates the QMA witness and teleports it to the second prover with the help of the verifier. The verifier then delegates the verification circuit to the second prover, as in the Dog-Walker Protocol; the first prover can be re-used to verify the operations of the second one.

Subsequent work. Bowles et al. [BvCA18] have independently re-derived a variant of our rigidity test for multi-qubit σ_X, σ_Y and σ_Z observables in the context of entanglement certification protocols in quantum networks. Their self-test result has a slightly smaller set of questions but significantly weaker robustness bounds.

Recently [Gri17] proposed the first protocol for verifiable delegation of quantum computation by classical clients where such space-like separation can replace the non-communication assumption, but his protocol is not blind.

Open questions and directions for future work. We have introduced a new rigidity theorem and shown how it can be used to transform a specific quantum-verifier delegation protocol, due to Broadbent, into a classical-verifier protocol with an additional prover, while suffering very little overhead in terms of the efficiency of the protocol. We believe that a similar transformation could be performed starting from delegation protocols based on other models of computation, such as the protocol in the measurement-based model of [FK17] or the protocol based on computation by teleportation considered in [RUV13], and would lead to similar efficiency improvements.

Recently, [HZM+17] provided an experimental demonstration of a two-prover delegation protocol based on [RUV13] for a 3-qubit quantum circuit based on Shor's algorithm to factor the number 15; in order to obtain an actual implementation, necessitating "only" on the order of 6000 CHSH tests, the authors had to make the strong assumption that the devices behave in an i.i.d. manner at each use, and could not use the most general testing results from [RUV13]. We believe that our improved rigidity theorem could lead to an implementation that does not require any additional assumption. We also leave as an open problem investigating whether (a variant of) our protocol can be made fault-tolerant, making it more suitable for future implementation.

We note that our protocols require the verifier to communicate with one prover after at least one round of communication with the other has been completed. Therefore, the requirement that the provers do not communicate throughout the protocol cannot be enforced through space-like separation, and must be taken as an a priori assumption. Since the protocol of [Gri17] is not blind, it is an open question whether there exists a two-prover delegation protocol that consists of a single round of simultaneous communication with each prover, and is blind and verifiable. We also wonder if the fact that blindness is compromised after the provers collude is unavoidable in this model. A different avenue to achieve this is to rely on computational assumptions on the power of the provers to achieve protocols with more properties (non-interactive, blind, verifiable) [DSS16,ADSS17,Mah17,Mah18], albeit not necessarily in a truly efficient manner.

Finally, due to its efficiency and robustness, our ridigity theorem is a potentially useful tool in many other cryptographic protocols. For instance, an interesting direction to explore is the possibility of exploiting our theorem to achieve more efficient protocols for device-independent quantum key distribution, entanglement certification or other cryptographic protocols involving more complex untrusted computation of the users.

Organization. In Section 2, we give the necessary preliminaries, including outlining Broadbent's EPR Protocol (Section 2.3). In Section 3, we introduce our new rigidity theorems. In Section 4, we present our first protocol, the leash protocol, and in Section 5, we discuss our second protocol, the Dog-Walker Protocol.

Acknowledgments. We thank Anne Broadbent for useful discussions in the early stages of this work. All authors acknowledge the IQIM, an NSF Physics Frontiers Center at the California Institute of Technology, where this research was initiated. AC is supported by AFOSR YIP award number FA9550-16-1-0495. AG is supported by ERC Consolidator Grant 615307-QPROGRESS and was previously supported by ERC QCC when AG was a member of IRIF (CNRS/Université Paris Diderot). SJ is supported by an NWO

WISE Grant. TV is supported by NSF CAREER Grant CCF-1553477, MURI Grant FA9550-18-1-0161, AFOSR YIP award number FA9550-16-1-0495, and the IQIM, an NSF Physics Frontiers Center (NSF Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028).

2 Preliminaries

2.1 Notation

We often write $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ for a string of bits, and $W = W_1 \cdots W_m \in \Sigma^m$ for a string, where Σ is a finite alphabet. If $S \subseteq \{1, \ldots, m\}$ we write W_S for the sub-string of W indexed by S. For an event E, we use 1_E to denote the indicator variable for that event, so $1_E = 1$ if E is true, and otherwise $1_E = 0$. We write $\mathrm{poly}(\varepsilon)$ for $O(\varepsilon^c)$, where c is a universal constant that may change each time the notation is used.

$\mathcal{H}$ is a finite-dimensional Hilbert space. We denote by $\mathrm{U}(\mathcal{H})$ the set of unitary operators, $\mathrm{Obs}(\mathcal{H})$ the set of binary observables (we omit the term "binary" from here on; in this paper all observables are binary) and $\mathrm{Proj}(\mathcal{H})$ the set of projective measurements on $\mathcal{H}$ respectively. We let $|\mathrm{EPR}\rangle$ denote an EPR pair:

$$|\mathrm{EPR}\rangle \;=\; \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right).$$

Observables. We use capital letters $X, Z, W, \ldots$ to denote observables. We use greek letters σ, τ with a subscript σ_W, τ_W, to emphasize that the observable W specified as subscript acts in a particular basis. For example, X is an arbitrary observable but σ_X is specifically the Pauli X matrix defined in (1).

For $a \in \{0,1\}^n$ and commuting observables $\sigma_{W_1}, \ldots, \sigma_{W_n}$, we write $\sigma_W(a) = \prod_{i=1}^n (\sigma_{W_i})^{a_i}$. The associated projective measurements are $\sigma_{W_i} = \sigma^0_{W_i} - \sigma^1_{W_i}$ and $\sigma^u_W = \mathrm{E}_a (-1)^{u \cdot a} \sigma_W(a)$. Often the σ_{W_i} will be single-qubit observables acting on distinct qubits, in which case each is implicitly tensored with identity outside of the qubit on which it acts.

Pauli and Clifford groups. Let

$$\sigma_I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \text{and} \quad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1}$$

denote the standard Pauli matrices acting on a qubit. The single-qubit Weyl-Heisenberg group

$$\mathcal{H}^{(1)} = H(\mathbb{Z}_2) = \left\{ (-1)^c \sigma_X(a) \sigma_Z(b),\ a, b, c \in \{0,1\} \right\}$$

is the matrix group generated by the Pauli σ_X and σ_Z. We let $\mathcal{H}^{(n)} = H(\mathbb{Z}_2^n)$ be the direct product of n copies of $\mathcal{H}^{(1)}$. The n-qubit Clifford group is the normalizer of $\mathcal{H}^{(n)}$ in the unitary group, up to phase:

$$G_{\mathcal{C}}^{(n)} = \left\{ G \in \mathrm{U}((\mathbb{C}^2)^{\otimes n}) : G \sigma G^\dagger \in \mathcal{H}^{(n)} \quad \forall \sigma \in \mathcal{H}^{(n)} \right\}.$$

Some Clifford observables we will use include

$$\sigma_H = \frac{\sigma_X + \sigma_Z}{\sqrt{2}}, \quad \sigma_{H'} = \frac{\sigma_X - \sigma_Z}{\sqrt{2}}, \quad \sigma_F = \frac{-\sigma_X + \sigma_Y}{\sqrt{2}}, \quad \sigma_G = \frac{\sigma_X + \sigma_Y}{\sqrt{2}}. \quad (2)$$

Note that σ_H and $\sigma_{H'}$ are characterized by $\sigma_X \sigma_H \sigma_X = \sigma_{H'}$ and $\sigma_Z \sigma_H \sigma_Z = -\sigma_{H'}$. Similarly, σ_F and σ_G are characterized by $\sigma_X \sigma_F \sigma_X = -\sigma_G$ and $\sigma_Y \sigma_F \sigma_Y = \sigma_G$.

2.2 Quantum circuits

We use capital letters in sans-serif font to denote gates. We work with the universal quantum gate set $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{T}\}$, where the controlled-not gate is the two-qubit gate with the unitary action

$$\mathsf{CNOT}|b_1, b_2\rangle = |b_1, b_1 \oplus b_2\rangle,$$

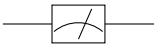
and the Hadamard and T gates are single-qubit gates with actions

$$\mathsf{H}|b\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^b|1\rangle\right) \text{ and } \mathsf{T}|b\rangle = e^{ib\pi/4}|b\rangle,$$

respectively. We will also use the following gates:

$$\mathsf{X}|b\rangle = |b \oplus 1\rangle, \ \mathsf{Z}|b\rangle = (-1)^b|b\rangle, \text{ and } \mathsf{P}|b\rangle = i^b|b\rangle.$$

Measurements in the Z basis (or computational basis) will be denoted by the standard measurement symbol:



To measure another observable, W, we can perform a unitary change of basis U_W before the measurement in the computational basis.

We assume that every circuit has a specified output wire, which is measured at the end of the computation to obtain the output bit. Without loss of generality, we can assume this is always the first wire. For an n-qubit system, we let Π_b, for $b \in \{0,1\}$, denote the orthogonal projector onto states with $|b\rangle$ in the output wire: $|b\rangle\langle b| \otimes \mathrm{Id}$. For example, the probability that a circuit Q outputs 0 on input $|\boldsymbol{x}\rangle$ is $\|\Pi_0 Q|\boldsymbol{x}\rangle\|^2$.

We can always decompose a quantum circuit into layers such that each layer contains at most one T gate applied to each wire. The minimum number of layers for which this is possible is called the T *depth* of the circuit. We note that throughout this work, we will assume circuits are compiled in a specific form that introduces extra T gates (see the paragraph on the H gadget in Section 2.3). The T depth of the resulting circuit is proportional to the depth of the original circuit.

2.3 Broadbent's EPR Protocol

In this section we summarize the main features of a delegation protocol introduced in [Bro18], highlighting the aspects that will be relevant to understanding our subsequent adaptation into two-prover protocols. The "EPR Protocol" from [Bro18] involves

the interaction between a verifier V_{EPR} and a prover P. We write P_{EPR} for the "honest" behavior of the prover. The verifier V_{EPR} has limited quantum powers. Her goal is to delegate a BQP computation to the prover P in a verifiable way. Specifically, the verifier has as input a quantum circuit Q on n qubits and an input string $\boldsymbol{x} \in \{0,1\}^n$, and the prover gets as input Q. The verifier and prover interact. At the end of the protocol, the verifier outputs either accept or reject. The protocol is such that there exist values p_{sound} and p_{compl} with $p_{\text{sound}} < p_{\text{compl}}$ such that $p_{\text{compl}} - p_{\text{sound}}$, called the *soundness-completeness gap*, is a constant independent of input size, and moreover:

Completeness: If the prover is honest and $\|\Pi_0 Q|\boldsymbol{x}\rangle\|^2 \geq 2/3$, then the verifier outputs accept with probability at least p_{compl};

Soundness: If $\|\Pi_0 Q|\boldsymbol{x}\rangle\|^2 \leq 1/3$, then the probability the verifier outputs accept is at most p_{sound}.

In the EPR protocol, V_{EPR} and P_{EPR} are assumed to share $(n+t)$ EPR pairs at the start of the protocol, where t is the number of T gates in Q and n the number of input bits. (In [Bro18] the EPR protocol is only considered in the analysis, and it is assumed that the EPR pairs are prepared by the verifier.) The first n EPR pairs correspond to the input to the computation; they are indexed by $N = \{1,\ldots,n\}$. The remaining pairs are indexed by $T = \{n+1,\ldots,n+t\}$; they will be used as ancilla qubits to implement each of the T gates in the delegated circuit.

The behavior of V_{EPR} depends on a *round type* randomly chosen by V_{EPR} after her interaction with P_{EPR}. There are three possible round types:

- Computation round ($r = 0$): the verifier delegates the computation to P_{EPR}, and at the end of the round can recover its output if P_{EPR} behaves honestly;
- X-test round ($r = 1$) and Z-test round ($r = 2$): the verifier tests that P_{EPR} behaves honestly, and rejects if malicious behavior is detected.

For some constant p, V chooses $r = 0$ with probability p, and otherwise chooses $r \in \{1,2\}$ with equal probability. Since the choice of round type is made after interaction with P_{EPR}, P_{EPR}'s behavior cannot depend on the round type. In particular, any deviating behavior in a computation round is reproduced in both types of test rounds. The analysis amounts to showing that any deviating behavior that affects the outcome of the computation will be detected in at least one of the test rounds.

In slightly more detail, the high-level structure of the protocol is the following. V_{EPR} measures her halves of the n qubits in N in order to prepare the input state on P_{EPR}'s system. As a result the input is quantum one-time padded with keys that depend on V_{EPR}'s measurement results. For example, in a computation round, V_{EPR} measures each input qubit in the Z basis, and gets some result $\boldsymbol{d} \in \{0,1\}^n$, meaning the input on P_{EPR}'s side has been prepared as $\mathsf{X}^{\boldsymbol{d}}|0\rangle^{\otimes n}$. In [Bro18], the input is always considered to be $\mathbf{0}$, but we can also prepare an arbitrary classical input $\boldsymbol{x} \in \{0,1\}^n$ by reinterpreting the one-time pad key as $\boldsymbol{a} = \boldsymbol{d} \oplus \boldsymbol{x}$ so that the input state on P_{EPR}'s side is $\mathsf{X}^{\boldsymbol{a}}|\boldsymbol{x}\rangle$. In a test round, on the other hand, the input is prepared as the one-time pad of either $|0\rangle^{\otimes n}$ or $|+\rangle^{\otimes n}$. Note that as indicated in Figure 2 this choice of measurements will be made after the interaction with P_{EPR} has taken place.

The honest prover P_{EPR} applies the circuit Q, which we assume is compiled in the universal gate set $\{\mathsf{H},\mathsf{T},\mathsf{CNOT}\}$, to his one-time padded input. We will shortly

describe gadgets that P_{EPR} can apply in order to implement each of the three gate types. The gadgets are designed in a way that in a test round each gadget amounts to an application of an identity gate; this is what enables V_{EPR} to perform certain tests in those rounds that are meant to identify deviating behavior of a dishonest prover. After each gadget, the one-time padded keys can be updated by V_{EPR}, who is able to keep track of the keys at any point in the circuit using the *update rules* in Table 2.

		Key Update Rule
T	Computation Round	$(a_j, b_j) \leftarrow (a_j + c_i, b_j + e_i + a_j + c_i + (a_j + c_i)z_i)$
	X-Test, even parity; or Z-test, odd parity	$(a_j, b_j) \leftarrow (e_i, 0)$
	Z-Test, even parity; or X-test, odd parity	$(a_j, b_j) \leftarrow (0, b_j + e_i + z_i)$
H		$(a_j, b_j) \leftarrow (b_j, a_j)$
CNOT		$(a_j, b_j, a_{j'}, b_{j'}) \leftarrow (a_j, b_j + b_{j'}, a_j + a_{j'}, b_{j'})$

Table 2: Rules for updating the one-time-pad keys after applying each type of gate in the EPR Protocol, in particular: after applying the i-th T gate to the j-th wire; applying an H gate to the j-th wire; or applying a CNOT gate controlled on the j-th wire and targeting the j'-th wire.

We now describe the three gadgets, before giving a complete description of the protocol.

CNOT Gadget To implement a CNOT gate on wires j and j', P_{EPR} simply performs the CNOT gate on those wires of his input qubits. The one-time pad keys are changed by the update rule in Table 2, because $\mathsf{CNOT} \cdot \mathsf{X}^{a_j}\mathsf{Z}^{b_j} \otimes \mathsf{X}^{a_{j'}}\mathsf{Z}^{b_{j'}} = \mathsf{X}^{a_j}\mathsf{Z}^{b_j+b_{j'}} \otimes \mathsf{X}^{a_j+a_{j'}}\mathsf{Z}^{b_{j'}} \cdot \mathsf{CNOT}$. Note that $\mathsf{CNOT}|0\rangle|0\rangle = |0\rangle|0\rangle$ and $\mathsf{CNOT}|+\rangle|+\rangle = |+\rangle|+\rangle$, so in the test runs, P_{EPR} is applying the identity.

H Gadget To implement an H gate on wire j, P_{EPR} simply performs the H on wire j, and the one-time-pad keys are changed as in Table 2. Unlike CNOT, H does not act as the identity on $|0\rangle$ and $|+\rangle$, so it is not the identity in a test round. To remedy this, assume that Q is compiled so that every H gate appears in a pattern $\mathsf{H}(\mathsf{TTH})^k$, where the maximal such k is odd. This can be accomplished by replacing each H by HTTHTTHTTH, which implements the same unitary. In test rounds, the T gadget, described shortly, implements the identity, and since $\mathsf{H}(\mathrm{Id}\,\mathsf{H})^k$ for odd k implements the identity, $\mathsf{H}(\mathsf{TTH})^k$ will also have no effect in test rounds.

Parity of a T Gate Within a pattern $\mathsf{H}(\mathsf{TTH})^k$, the H has the effect of switching between an X-test round scenario (the state $|0\rangle$) and a Z-test round scenario (the state $|+\rangle$). In order to consistently talk about the type of a round while evaluating the circuit, we can associate a parity with each T gate in the circuit. The parity of the T gates that are not part of the pattern $\mathsf{H}(\mathsf{TTH})^k$ will be defined to be even. A H will always flip the parity, so that within such a pattern, the first two T gates will be odd, the next two will be even, etc., until the last two T gates will be odd again.

T Gadget The gadget for implementing the i-th T gate on the j-th wire is performed on P_{EPR}'s j-th input qubit, and his i-th auxiliary qubit (indexed by $n+i$), which we can think of as being prepared in a particular auxiliary state by V_{EPR} measuring her half of the corresponding EPR pair, as shown in Figure 1. The gadget depends on a random bit z_i that is chosen by V_{EPR} and sent to the prover.

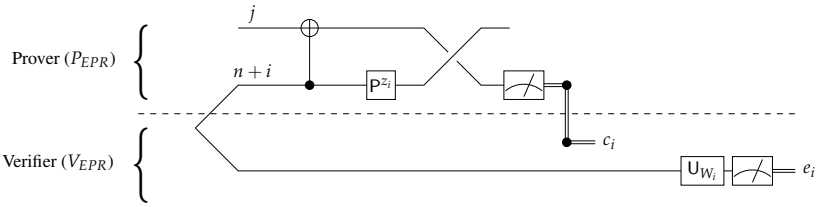


Fig. 1: The gadget for implementing the i-th T gate on the j-th wire. The gate U_{W_i} implementing the change of basis associated with observable W_i is applied as part of the procedure V^r_{EPR} (see Figure 3b) and is determined by the round type r, the parity of the i-th T gate, z_i, c_i, and a'_i (the X-key going into the i-th T gate), as in Table 3.

			U_{W_i} (observable W_i)
Computation Round	$a'_i \oplus c_i \oplus z_i = 0$		HT (observable G)
	$a'_i \oplus c_i \oplus z_i = 1$		HPT (observable F)
X-Test Round	even T gate		Id (observable Z)
	odd T gate	$z_i = 0$	H (observable X)
		$z_i = 1$	HP (observable Y)
Z-Test Round	odd T gate		Id (observable Z)
	even T gate	$z_i = 0$	H (observable X)
		$z_i = 1$	HP (observable Y)

Table 3: The choice of U_{W_i} in the T gadget. We also indicate the observable W_i associated with the final measurement $W_i = \mathsf{U}^\dagger_{W_i} Z \mathsf{U}_{W_i}$.

The EPR Protocol. We show how the gadgets just described are used in the complete protocol. We first describe the protocol for V_{EPR} below. For later convenience we have divided the action of V_{EPR} into classical actions and a single quantum subroutine V^r_{EPR} depending on the round type.

The procedure V^r_{EPR} measures each of the $n+t$ EPR halves according to some observable that depends on r, $\boldsymbol{c}$, and $\boldsymbol{z}$. In the case of a computation round, V^0_{EPR} mea-

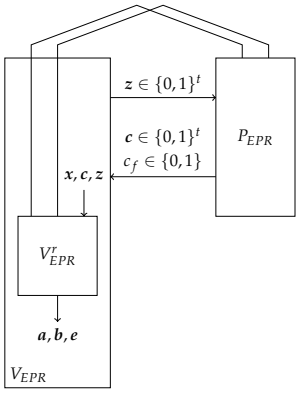


Fig. 2: This figure describes how different pieces of the protocol fit together. V_{EPR} and P_{EPR} share $n+t$ EPR pairs. The honest prover P_{EPR} can be seen as a procedure that acts on $n+t$ qubits — the EPR pair halves — depending on a t-bit string z. We have separated the quantum part of V_{EPR} into its own procedure, called V^r_{EPR}, where $r \in \{0,1,2\}$ indicates the *round type*, which V_{EPR} runs on her $n+t$ EPR halves, and the $2t$ bits c and z. Aside from running V^r_{EPR}, V_{EPR} is classical.

sures the qubits in T adaptively. We describe the steps of V_{EPR}, V^r_{EPR} and the honest behaviour of P_{EPR} in Fig. 3.

Completeness and Soundness. We summarize the relevant part of the analysis of the EPR protocol from [Bro18]. First suppose P_{EPR} behaves honestly. If $\|\Pi_0 Q|0^n\rangle\|^2 = p$, then in a computation round, V_{EPR} outputs accept with probability p, whereas in a test round, V_{EPR} outputs accept with probability 1. This establishes completeness of the protocol:

Theorem 2 (Completeness). *Suppose the verifier executes the EPR Protocol, choosing $r = 0$ with probability p, on an input $(Q, |x\rangle)$ such that $\|\Pi_0 Q|x\rangle\|^2 \geq 1-\delta$. Then the probability that V_{EPR} accepts when interacting with the honest prover P_{EPR} is at least $(1-p)+p(1-\delta)$.*

The following theorem is implicit in [Bro18, Section 7.6], but we include a brief proof sketch:

Theorem 3 (Soundness). *Suppose the verifier executes the EPR Protocol, choosing $r = 0$ with probability p, on an input $(Q, |x\rangle)$ such that $\|\Pi_0 Q|x\rangle\|^2 \leq \delta$. Let P^*_{EPR} be an arbitrary prover such that P^*_{EPR} is accepted by V_{EPR} with probability q_t conditioned on $r \neq 0$, and q_c conditioned on $r = 0$. Then the prover's overall acceptance probability is $pq_c + (1-p)q_t$, and*

$$q_c \leq 2\left(q_t\,\delta + (1-q_t)\right) - \delta.$$

Proof (Proof sketch). Using the notation of [Bro18], let $A = \sum_k \sum_{Q \in B'_{t,n}} |\alpha_{k,Q}|^2$.[4] For intuition, A should be thought of as the total weight on attacks that could change the outcome of the computation, called non-benign attacks in [Bro18]. By [Bro18], the probability of rejecting in a computation round is $1 - q_c \geq (1-\delta)(1-A)$, whereas the probability of rejecting in a test round is $1 - q_t \geq \frac{1}{2}A$. Combining these gives $q_c \leq 2(q_t\delta + (1-q_t)) - \delta$.

[4] Here, we consider the decomposition of the attack as a sum of tensors of Pauli $A = \sum_k \sum_{Q \in \{I,X,Z,Y\}} \alpha_{k,Q} Q$

1. V_{EPR} sends $\boldsymbol{z} \in_R \{0,1\}^t$ to P_{EPR}, and receives back $\boldsymbol{c} \in \{0,1\}^t$ and $c_f \in \{0,1\}$.
2. V_{EPR} chooses a random round type $r \in \{0,1,2\}$ and runs V^r_{EPR} (see Figure 3b) on her EPR halves, $\boldsymbol{x}$, $\boldsymbol{c}$ and $\boldsymbol{z}$, to obtain bits $\boldsymbol{a}, \boldsymbol{b} \in \{0,1\}^n$ and $\boldsymbol{e} \in \{0,1\}^t$.
3. V_{EPR} applies the update rules from Table 2 on the initial keys $(\boldsymbol{a}, \boldsymbol{b})$, gate-by-gate, to obtain, for every $i \in [t]$, the X-key before the i-th T gate is applied, a'_i, and the final X key for the output wire, a'_f. If $r = 1$ (X-test round) and there exists an i such that the i-th T gate is even and $c_i \neq a'_i \oplus e_i$, output reject. If $r = 2$ (Z-test round) and there exists an i such that the i-th T gate is odd and $c_i \neq a'_i \oplus e_i$, output reject. If $r \in \{0,1\}$ (computation or X-test round) and $c_f \oplus a'_f \neq 0$, output reject. Otherwise, output accept.

(a) V_{EPR}'s point of view.

Input: A circuit Q with t T gates, $\boldsymbol{x} \in \{0,1\}^n$, $\boldsymbol{c}, \boldsymbol{z} \in \{0,1\}^t$, an n-qubit system indexed by N, and a t-qubit system indexed by T.

1. If $r \in \{0,1\}$, measure each qubit in N in the Z basis, and otherwise measure in the X basis, to get results $\boldsymbol{d} \in \{0,1\}^n$. If $r = 0$, set $(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{d} \oplus \boldsymbol{x}, 0^n)$; if $r = 1$, set $(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{d}, 0^n)$; and if $r = 2$ set $(\boldsymbol{a}, \boldsymbol{b}) = (0^n, \boldsymbol{d})$.
2. Going through Q gate-by-gate, use the update rules in Table 2 to update the one-time-pad keys. For every $i \in [t]$, when the i-th T gate is reached, let a'_i be the X key before the i-th T gate is applied. Choose an observable W_i according to Table 3 in which to measure the i-th qubit in T, corresponding to the i-th T gate, obtaining result e_i.

(b) The procedure V^r_{EPR}, employed by V_{EPR}.

1. Receive $\boldsymbol{z} \in \{0,1\}^t$ from V_{EPR}.
2. Evaluate Q gate-by-gate using the appropriate gadget for each gate. In particular, use z_i to implement the i-th T gadget, and obtain measurement result c_i.
3. Measure the output qubit to obtain c_f, and return $\boldsymbol{c}$ and c_f to V_{EPR}.

(c) Honest prover strategy P_{EPR}

Fig. 3: The EPR Protocol.

3 Rigidity

Each of our delegation protocols includes a *rigidity test* that is meant to verify that one of the provers measures his half of shared EPR pairs in a basis specified by the verifier, thereby preparing one of a specific family of post-measurement states on the other prover's space; the post-measurement states will form the basis for the delegated computation. This will be used to certify that one of the provers in our two-prover schemes essentially behaves as the quantum part of V_{EPR} would in the EPR protocol.

In this section we outline the structure of the test, giving the important elements for its use in our delegation protocols. We refer the reader to the full version of the paper for a detailed presentation, including the soundness analysis. The test is parametrized by the number m of EPR pairs to be used. The test consists of a single round of clas-

sical interaction between the verifier and the two provers. With constant probability the verifier sends one of the provers a string W chosen uniformly at random from Σ^m where the set $\Sigma = \{X, Y, Z, F, G\}$ contains a label for each single-qubit observable to be tested. With the remaining probability, other queries, requiring the measurement of observables not in Σ^m (such as the measurement of pairs of qubits in the Bell basis), are sent.

In general, an arbitrary strategy for the provers consists of an arbitrary entangled state $|\psi\rangle \in \mathcal{H}_\mathsf{A} \otimes \mathcal{H}_\mathsf{B}$ (which we take to be pure), and measurements (which we take to be projective) for each possible question.[5] This includes an m-bit outcome projective measurement $\{W^u\}_{u\in\{0,1\}^m}$ for each of the queries $W \in \Sigma^m$. Our rigidity result states that any strategy that succeeds with probability $1-\varepsilon$ in the test is within $\mathrm{poly}(\varepsilon)$ of the honest strategy, up to local isometries (see Theorem 4 for a precise statement). This is almost true, but for an irreconcilable ambiguity in the definition of the complex phase $\sqrt{-1}$. The fact that complex conjugation of observables leaves correlations invariant implies that no classical test can distinguish between the two nontrivial inequivalent irreducible representations of the Pauli group, which are given by the Pauli matrices $\sigma_X, \sigma_Y, \sigma_Z$ and their complex conjugates $\overline{\sigma_X} = \sigma_X$, $\overline{\sigma_Z} = \sigma_Z$, $\overline{\sigma_Y} = -\sigma_Y$ respectively. In particular, the provers may use a strategy that uses a combination of both representations; as long as they do so consistently, no test will be able to detect this behavior.[6]. The formulation of our result accommodates this irreducible degree of freedom by forcing the provers to use a single qubit, the $(m+1)$-st, to make their choice of representation (so honest provers require the use of $(m+1)$ EPR pairs to test the operation of m-fold tensor products of observables from Σs).

Theorem 4 below summarizes the guarantees of our main test, which is denoted as $\textsc{rigid}(\Sigma, m)$. Informally, Theorem 4 establishes that a strategy that succeeds in $\textsc{rigid}(\Sigma, m)$ with probability at least $1-\epsilon$ must be such that (up to local isometries):

- The players' joint state is close to a tensor product of m EPR pairs, together with an arbitrary ancilla register;
- The projective measurements performed by either player upon receipt of a query of the form $W \in \Sigma^m$ are, on average over the uniformly random choice of $W \in \Sigma^m$, close to a measurement that consists in first, measuring the ancilla register to extract a single bit that specifies whether to perform the ideal measurements or their conjugated counterparts, and second, measuring the player's m half-EPR pairs in either the bases indicated by W, or their complex conjugate, depending on the bit obtained from the ancilla register.

For an observable $W \in \Sigma$, let $\sigma_W = \sigma_W^{+1} - \sigma_W^{-1}$ be its eigendecomposition, where σ_W are the "honest" Pauli matrices defined in (1) and (2). For $u \in \{\pm 1\}$ let $\sigma^u_{W,+} = \sigma^u_W$

[5] We make the assumption that the players employ a pure-state strategy for convenience, but it is easy to check that all proofs extend to the case of a mixed strategy. Moreover, it is always possible to consider (as we do) projective strategies only by applying Naimark's dilation theorem, and adding an auxiliary local system to each player as necessary, since no bound is assumed on the dimension of their systems.

[6] See [RUV12, Appendix A] for an extended discussion of this issue, with a similar resolution to ours.

for $W \in \Sigma$, and

$$\sigma^u_{X,-} = \sigma^u_X, \quad \sigma^u_{Z,-} = \sigma^u_Z, \quad \sigma^u_{Y,-} = \sigma^{-u}_Y, \quad \sigma^u_{F,-} = \sigma^{-u}_G, \quad \sigma^u_{G,-} = \sigma^{-u}_F .$$

(In words, $\sigma^u_{W,-}$ is just the complex conjugate of σ^u_W.) We note that for the purpose of our delegation protocols, we made a particular choice of the set Σ. The result generalizes to any constant-sized set of single-qubit Clifford observables, yielding a test for m-fold tensor products of single-qubit Clifford observables from Σ.

Theorem 4. *Let $\varepsilon > 0$ and m an integer. Suppose a strategy for the players succeeds with probability $1-\varepsilon$ in test $\textsc{rigid}(\Sigma, m)$. For $W \in \Sigma^m$ and $D \in \{A, B\}$ let $\{W^u_D\}_u$ be the measurement performed by prover D on question W. Let also $|\psi\rangle$ be the state shared by the players. Then for $D \in \{A, B\}$ there exists an isometry*

$$V_D : \mathcal{H}_D \to (\mathbb{C}^2)^{\otimes m}_{D'} \otimes \mathcal{H}_{\widehat{D}}$$

such that

$$\left\| (V_A \otimes V_B)|\psi\rangle_{AB} - |\mathrm{EPR}\rangle^{\otimes m} \otimes |\mathrm{AUX}\rangle_{\widehat{A}\widehat{B}} \right\|^2 = O(\sqrt{\varepsilon}), \tag{3}$$

and positive semidefinite matrices τ_λ on $\widehat{A}$ with orthogonal support, for $\lambda \in \{+,-\}$, such that $\mathrm{Tr}(\tau_+) + \mathrm{Tr}(\tau_-) = 1$ and

$$\begin{aligned} \mathop{\mathrm{E}}_{W\in\Sigma^m} \sum_{u\in\{\pm 1\}^m} \Big\| & V_A \mathrm{Tr}_B\big((\mathrm{Id}_A \otimes W^u_B)|\psi\rangle\langle\psi|_{AB}(\mathrm{Id}_A \otimes W^u_B)^\dagger\big) V_A^\dagger \\ & - \sum_{\lambda\in\{\pm\}} \Big(\bigotimes_{i=1}^m \frac{\sigma^{u_i}_{W_i,\lambda}}{2} \Big) \otimes \tau_\lambda \Big\|_1 \\ = O(\mathrm{poly}(\varepsilon)). & \end{aligned}$$

Moreover, players employing the honest strategy succeed with probability $1 - e^{-\Omega(m)}$ in the test.

The proof of the theorem is based on standard techniques developed in the literature on "rigidity theorems" for nonlocal games. We highlight two components. The first is a "conjugation test" that allows us to extend the guarantees of elementary tests based on the CHSH game or the Magic Square game, which test for Pauli σ_X and σ_Z observables, to a test for single-qubit Clifford observables — since the latter are characterized by their action on the Pauli group (see full version of the paper for details). The second is an extension of the "Pauli braiding test" from [NV17] to handle tensor products of not only σ_X and σ_Z, but also σ_Y Pauli observables (see full version of the paper for details). As already emphasized in the introduction, the improvements in efficiency of our scheme are partly enabled by the strong guarantees of Theorem 4, and specifically the independence of the final error dependence from the parameter m.

4 The Verifier-on-a-Leash Protocol

4.1 Protocol and statement of results

The Verifier-on-a-Leash Protocol (or "Leash Protocol" for short) involves a classical verifier and two quantum provers. The idea behind the Leash Protocol is to have a

first prover, nicknamed PV for Prover V, carry out the quantum part of V_{EPR} from Broadbent's EPR Protocol by implementing the procedure V_{EPR}^r. (See Section 2.3 for a summary of the protocol and a description of V_{EPR}. Throughout this section we assume that the circuit Q provided as input is compiled in the format described in Section 2.3.). A second prover, nicknamed PP for Prover P, will play the part of the prover P_{EPR}. Unlike in the EPR Protocol, the interaction with PV (i.e. running V_{EPR}^r) will take place first, and PV will be asked to perform random measurements from the set $\Sigma = \{X, Y, Z, F, G\}$. The values $\boldsymbol{z}$, rather than being chosen at random, will be chosen based on the corresponding choice of observable. We let n be the number of input bits and t number of T gates in Q.

The protocol is divided into two sub-games; which game is played is chosen by the verifier by flipping a biased coin with probability $(p_r, p_d = 1 - p_r)$.

- The first game is a sequential version of the rigidity game $\text{RIGID}(\Sigma, m)$ described in Figure 9. This aims to enforce that PV performs precisely the right measurements;
- The second game is the delegation game, described in Figures 6, 7, and 8, and whose structure is summarized in Figure 4. Here the verifier guides PP through the computation in a similar way as in the EPR Protocol.

We call the resulting protocol the Leash Protocol with parameters (p_r, p_d). In both sub-games the parameter $m = \Theta(n + t)$ is chosen large enough so that with probability close to 1 each symbol in Σ appears in a random $W \in \Sigma^m$ at least $n + t$ times. It is important that PV is not able to tell which kind of game is being played. Notice also that in order to ensure blindness, we will require that the interaction with PV in the delegation game is sequential (more details on this are found in Section 4.4). In order for the two sub-games to be indistinguishable, we also require that the rigidity game $\text{RIGID}(\Sigma, m)$ be played sequentially (i.e. certain subsets of questions and answers are exchanged sequentially, but the acceptance condition in the test is the same). Note, importantly, that the rigidity guarantees of $\text{RIGID}(\Sigma, m)$ hold verbatim when the game is played sequentially, since this only reduces the number of ways that the provers can cheat. The following theorem states the guarantees of the Leash Protocol.

Theorem 5. *There are constants $p_r, p_d = 1 - p_r$, and $\Delta > 0$ such that the following hold of the Verifier-on-a-Leash Protocol with parameters (p_r, p_d), when executed on an input $(Q, |\boldsymbol{x}\rangle)$.*

- (Completeness:) *Suppose that $\|\Pi_0 Q|\boldsymbol{x}\rangle\|^2 \geq 2/3$. Then there is a strategy for* PV *and* PP *that is accepted with probability at least $p_{\text{compl}} = p_r(1 - e^{-\Omega(n+t)}) + 8p_d/9$.*
- (Soundness:) *Suppose that $\|\Pi_0 Q|\boldsymbol{x}\rangle\|^2 \leq 1/3$. Then any strategy for* PV *and* PP *is accepted with probability at most $p_{\text{sound}} = p_{\text{compl}} - \Delta$.*

Further, the protocol leaks no information about $\boldsymbol{x}$ to either prover individually, aside from an upper bound on the length of $\boldsymbol{x}$.

The proof of the completeness property is given in Lemma 1. The soundness property is shown in Lemma 4. Blindness is established in Section 4.4. We first give a detailed description of the protocol. We start by describing the delegation game, specified in Figures 6, 7 and 8, which describe the protocol from the verifier's view, an honest

PV's view, and an honest PP's view respectively. This will motivate the need for a sequential version of the game $\text{RIGID}(\Sigma, m)$, described in Figure 9. As we will show, the rigidity game forces PV to behave honestly. Thus, for the purpose of exposition, we assume for now that PV behaves honestly, which results in the joint behavior of PV and V being similar to that of the verifier V_{EPR} in the EPR Protocol.

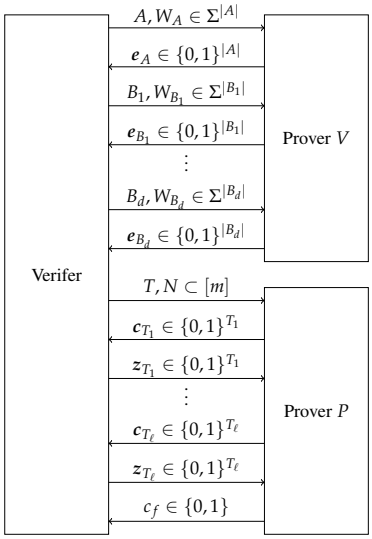


Fig. 4: Structure of the delegation game.

From the rigidity game we may also assume that PV and PP share m EPR pairs, labeled $\{1,\dots,m\}$, for $m = \Theta(n+t)$. We will assume that the circuit Q is broken into d layers, $Q = Q_1 \dots Q_d$, such that in every Q_ℓ, each wire has at most one T gate applied to it, after which no other gates are applied to that wire. We will suppose the T gates are indexed from 1 to t, in order of layer.

The protocol begins with an interaction between the verifier and PV. The verifier selects a uniformly random partition $A, B_1, \dots, B_d$ of $\{1,\dots,m\}$, with $|A| = \Theta(n)$, and for every $\ell \in \{1,\dots,d\}$, $|B_\ell| = \Theta(t_\ell)$, where t_ℓ is the number of T gates in Q_ℓ. The verifier also selects a uniformly random $W \in \Sigma^m$, and partitions it into substrings W_A and $W_{B_1}, \dots, W_{B_d}$, meant to contain observables to initialize the computation qubits and auxiliary qubits for each layer of T gates respectively. The verifier instructs PV to measure his halves of the EPR pairs using the observables W_A first, and then $W_{B_1}, \dots, W_{B_d}$, sequentially. Upon being instructed to measure a set of observables, PV measures the corresponding half-EPR pairs and returns the results $\boldsymbol{e}$ to the verifier. Breaking this interaction into multiple rounds is meant to enforce that, for example, the results output by PV upon receiving W_{B_ℓ}, which we call $\boldsymbol{e}_{B_\ell}$, cannot depend on the choice of observables $W_{B_{\ell+1}}$. This is required for blindness.

Once the interaction with PV has been completed, as in the EPR Protocol, V selects one of three round types: computation ($r = 0$), X-test ($r = 1$), and Z-test ($r = 2$). The verifier selects a subset $N \subset A$ of size n of qubits to play the role of inputs to the computation. These are chosen from the subset of A corresponding to wires that PV has measured in the appropriate observable for the round type (see Table 4). For example, in an X-test round, PV's EPR halves corresponding to input wires should be measured in the Z basis so that PP is left with a one-time pad of the state $|0\rangle^{\otimes n}$, so in an X-test round, the computation wires are chosen from the set $\{i \in A : W_i = Z\}$. The input wires N are labeled by $\mathcal{X}_1, \dots, \mathcal{X}_n$.

The verifier also chooses subsets $T_\ell = T_\ell^0 \cup T_\ell^1 \subset B_\ell$ of sizes $t_{\ell,0}$ and $t_{\ell,1} = t_\ell - t_{\ell,0}$ respectively, where $t_{\ell,0}$ is the number of odd T gates in the ℓ-th layer of Q (recall the definition of even and odd T gates from Section 2.3). The wires T_ℓ^0 and

T^1_ℓ will play the role of auxiliary states used to perform T gates from the ℓ-th layer. They are chosen from those wires from B_ℓ whose corresponding EPR halves have been measured in a correct basis, depending on the round type. For example, in an X-test round, the auxiliaries corresponding to odd T gates should be prepared by measuring the corresponding EPR half in either the X or Y basis (see Table 3), so in an X-test round, T^1_ℓ is chosen from $\{i \in B_\ell : W_i \in \{X, Y\}\}$ (see Table 4). We will let $\mathcal{T}_1, \ldots, \mathcal{T}_t$ label those EPR pairs that will be used as auxiliary states. In particular, the system $\mathcal{T}_i$ will be used for the i-th T gate in the circuit, so if the i-th T gate is even, $\mathcal{T}_i$ should be chosen from $T^0 = \cup_\ell T^0_\ell$, and otherwise it should be chosen from $T_1 = \cup_\ell T^1_\ell$. The verifier sends labels $\mathcal{T}_1, \ldots, \mathcal{T}_t$ and $\mathcal{X}_1, \ldots, \mathcal{X}_n$ to PP, who will act as P_{EPR} on the $n+t$ qubits specified by these labels.

Just as in the EPR Protocol, the input on PP's system specified by $\mathcal{X}_1, \ldots, \mathcal{X}_n$ is a quantum one-time pad of either $|\boldsymbol{x}\rangle$, $|0\rangle^{\otimes n}$, or $|+\rangle^{\otimes n}$, depending on the round type, with V holding the keys (determined by $\boldsymbol{e}$). Throughout the interaction, PP always maintains a one-time pad of the current state of the computation, with the verifier in possession of the one-time-pad keys. The verifier updates her keys as the computation is carried out, using the rules in Table 2.

From PP's perspective, the protocol works just as the EPR Protocol, except that he does not receive the bit z_i needed to implement the T gadget until *during* the T gadget, after he has sent V his measurement result c_i (see Figure 5).

To perform the i-th T gate on the j-th wire, PP performs the circuit shown in Figure 5. As Figure 5 shows, PV has already applied the observable specified by V to his half of the EPR pair. The T gadget requires interaction with the verifier, to compute the bit z_i, which depends on the measured c_i, the value W_i, and one-time-pad key a_j, however, this interaction can be done in parallel for T gates in the same layer.

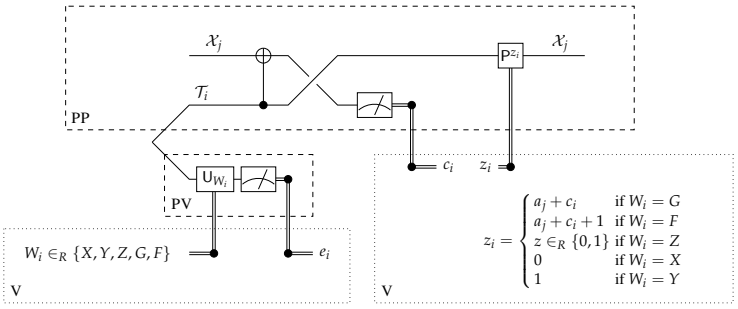


Fig. 5: The gadget for implementing the i-th T gate, on the j-th wire.

It is simple to check that the T gadget in Figure 5 is the same as the T gadget for the EPR Protocol shown in Figure 1. In the case of the leash protocol, W is chosen at random, and then $\boldsymbol{z}$ is chosen accordingly, whereas in the case of the EPR Protocol, $\boldsymbol{z}$ is chosen at random and then W is chosen accordingly.

We now give the precise protocols for V (Figure 6) and honest provers PV (Figure 7) and PP (Figure 8).

	Computation Round	X-test Round	Z-test Round
N	$\{i \in A : W_i = Z\}$	$\{i \in A : W_i = Z\}$	$\{i \in A : W_i = X\}$
T^0_ℓ	$\{i \in B_\ell : W_i \in \{G,F\}\}$	$\{i \in B_\ell : W_i = Z\}$	$\{i \in B_\ell : W_i \in \{X,Y\}\}$
T^1_ℓ	$\{i \in B_\ell : W_i \in \{G,F\}\}$	$\{i \in B_\ell : W_i \in \{X,Y\}\}$	$\{i \in B_\ell : W_i = Z\}$

Table 4: How the verifier chooses index sets $T = T^0 \cup T^1$ and N for each type of round. These sets determine which systems are labeled by $\{\mathcal{T}_i\}_{i=1}^t$ and $\{\mathcal{X}_j\}_{j=1}^n$, respectively.

Let $(Q, \boldsymbol{x})$ be the input to the verifier, where Q is compiled in the form described in Section 2.3. Let n be the size of the input to Q. Let d be the T-depth, and for $\ell \in \{1,\dots,d\}$ let t_ℓ be the number of T gates in the ℓ-th layer.

1. The verifier selects $W \in_R \Sigma^m$. She partitions $\{1,\dots,m\}$ arbitrarily into non-overlapping subsets A and $B_1,\dots,B_d$ such that W_A contains at least n copies of each symbol in Σ, and for each $\ell \in \{1,\dots,d\}$, W_{B_ℓ} contains at least t_ℓ copies of each symbol in Σ.
2. The verifier sends A, W_A to PV, who returns $\boldsymbol{e}_A \in \{0,1\}^A$. The verifier sequentially sends $(B_1, W_{B_1}),\dots,(B_d, W_{B_d})$ to PV, each time receiving $\boldsymbol{e}_{B_\ell} \in \{0,1\}^{B_\ell}$ as answer.
3. The verifier selects a round type uniformly at random. She selects sets $N \subseteq A$ and $T_\ell \subseteq B_\ell$, for $\ell \in \{1,\dots,d\}$, of sizes $|N| = n$ and $|T_\ell| = t_\ell$, as follows:
   Computation Round: N is chosen at random from $\{i \in A : W_i = Z\}$. T_ℓ is chosen at random from $\{i \in B_\ell : W_i \in \{G,F\}\}$. She sets $\boldsymbol{a} = \boldsymbol{e}_N + \boldsymbol{x}$ and $\boldsymbol{b} = 0^n$.
   X-test Round: N is chosen at random from $\{i \in A : W_i = Z\}$. $T_\ell = T^0_\ell \cup T^1_\ell$, where T^0_ℓ is of size $t_{\ell,0}$ chosen at random from $\{i \in B_\ell : W_i = Z\}$ and T^1_ℓ is of size $t_{\ell,1}$ chosen at random from $\{i \in B_\ell : W_i \in \{X,Y\}\}$. She sets $\boldsymbol{a} = \boldsymbol{e}_N$ and $\boldsymbol{b} = 0^n$.
   Z-test Round: N is chosen at random from $\{i \in A : W_i = X\}$. $T_\ell = T^0_\ell \cup T^1_\ell$, where T^0_ℓ is of size $t_{\ell,0}$ chosen at random from $\{i \in B_\ell : W_i \in \{X,Y\}\}$ and T^1_ℓ is of size $t_{\ell,1}$ chosen at random from $\{i \in B_\ell : W_i = Z\}$. She sets $\boldsymbol{a} = 0^n$ and $\boldsymbol{b} = \boldsymbol{e}_N$.
   The verifier sends the sets N and T^0_ℓ, T^1_ℓ, for all $\ell \in \{1,\dots,d\}$, to PP.
4. For $\ell = 1,\dots,d$,
   (a) For each Clifford gate in the ℓ-th layer, perform the appropriate key update (Table 2).
   (b) The verifier receives $\boldsymbol{c} = \{c_i\}_{i \in T_\ell}$ from PP. If it's an X-test round and $i \in T^0_\ell$, or it's a Z-test round and $i \in T^1_\ell$, reject if $c_i \neq a_j + e_i$, where j is the wire to which the i-th T gate is applied.
   (c) For each $i \in T_\ell$, the verifier computes $\boldsymbol{z} = \{z_i\}_{i \in T_\ell}$ as follows:
   Computation Round $z_i = a_j + 1_{W_i = F} + c_i$;
   X-test Round if $i \in T^0_\ell$, $z_i \in_R \{0,1\}$; else if $i \in T^1_\ell$, $z_i = 1_{W_i = Y}$;
   Z-test Round if $i \in T^0_\ell$, $z_i = 1_{W_i = Y}$; else if $i \in T^1_\ell$, $z_i \in_R \{0,1\}$.
   (d) The verifier sends $\boldsymbol{z}$ to PP and updates keys for each T gate applied (Table 2).
5. The verifier receives a bit c_f from PP. She outputs reject if it's a computation or X-test round and $c_f + a_f \neq 0$, where a_f is the final X-key on the output wire; and accept otherwise.

Fig. 6: The Delegation Game: Verifier's point of view.

Finally, we describe the sequential version of the game $\textsc{rigid}(\Sigma, m)$ in Figure 9. It is no different than $\textsc{rigid}(\Sigma, m)$, except for the fact that certain subsets of questions

1. For $\ell = 0, 1, \ldots, d$,
   (a) PV receives a string $W_S \in \Sigma^S$, for some subset S of $\{1, \ldots, m\}$, from V.
   (b) For $i \in S$, PV measures his half of the i-th EPR pair using the observable indicated by W_i, obtaining an outcome $e_i \in \{0, 1\}$.
   (c) PV returns $\boldsymbol{e}_S$ to V.

Fig. 7: Honest strategy for PV

1. PP receives subsets N and T^0_ℓ, T^1_ℓ of $\{1, \ldots, m\}$, for $\ell \in \{1, \ldots, d\}$, from the verifier.
2. For $\ell = 1, \ldots, d$,
   (a) PP does the Clifford computations in the ℓ-th layer.
   (b) For each $i \in T_\ell = T^0_\ell \cup T^1_\ell$, PP applies a CNOT from $\mathcal{T}_i$ into the input register corresponding to the wire on which this T gate should be performed, $\mathcal{X}_j$, and measures this wire to get a value c_i. The register $\mathcal{T}_i$ is relabeled $\mathcal{X}_j$. He sends $\boldsymbol{c}_{T_\ell} = \{c_i\}_{i \in T_\ell}$ to V.
   (c) PP receives $\boldsymbol{z}_{T_\ell} = \{z_i\}_{i \in T_\ell}$ from V. For each $i \in T_\ell$, he applies P^{z_i} to the corresponding $\mathcal{X}_j$.
3. PP performs the final computations that occur after the d-th layer of T gates, measures the output qubit, $\mathcal{X}_1$, and sends the resulting bit, c_f, to V.

Fig. 8: Honest strategy for PP

and answers are exchanged sequentially, but the acceptance condition is the same. As mentioned earlier, running the game sequentially only reduces the provers' ability to cheat. Hence the guarantees from $\text{RIGID}(\Sigma, m)$ hold verbatim for the sequential version.

Let m, n, and $t_1, \ldots, t_d$ be parameters provided as input, such that $m = \Theta(n + t_1 + \cdots + t_d)$.

1. The verifier selects questions $W, W' \in \Sigma^m$, for the first and second player respectively, according to the distribution of questions in the game $\text{RIGID}(\Sigma, m)$. She partitions $\{1, \ldots, m\}$ at random into subsets A and B_ℓ, for $\ell \in \{1, \ldots, d\}$, of size $|A| = \Theta(n)$ and $|B_\ell| = \Theta(t_\ell)$, exactly as in Step 1 of the Delegation Game.
2. The verifier sends $(A, W_A), (B_1, W_{B_1}), .., (B_d, W_{B_d})$ and $(A, W'_A), (B_1, W'_{B_1}), .., (B_d, W'_{B_d})$ in sequence to the first and second prover respectively. They sequentially return respectively $\boldsymbol{e}_A \in \{0,1\}^{|A|}$, $\boldsymbol{e}_{B_1} \in \{0,1\}^{|B_1|}, .., \boldsymbol{e}_{B_d} \in \{0,1\}^{|B_d|}$ and $\boldsymbol{e}'_A \in \{0,1\}^{|A|}$, $\boldsymbol{e}'_{B_1} \in \{0,1\}^{|B_1|}, .., \boldsymbol{e}'_{B_d} \in \{0,1\}^{|B_d|}$.
3. The verifier accepts if and only if $\boldsymbol{e}, \boldsymbol{e}'$ and W, W' satisfy the winning condition of $\text{RIGID}(\Sigma, m)$.

Fig. 9: Sequential version of $\text{RIGID}(\Sigma, m)$.

4.2 Completeness

Lemma 1. *Suppose the verifier executes the rigidity game with probability p_r and the delegation game with probability $p_d = 1 - p_r$, on an input $(Q, |x\rangle)$ such that $\|\Pi_0 Q|x\rangle\|^2 \geq 2/3$. Then there is a strategy for the provers which is accepted with probability at least $p_{\text{compl}} = p_r(1 - e^{-\Omega(n+t)}) + \frac{8}{9}p_d$.*

Proof. The provers PV and PP play the rigidity game according to the honest strategy, and the delegation game as described in Figures 7 and 8 respectively. Their success probability in the delegation game is the same as the honest strategy in the EPR Protocol, which is at least $\frac{2}{3} + \frac{2}{3}\frac{1}{3} = \frac{8}{9}$, by Theorem 2 and since in our protocol the verifier chooses each of the three types of rounds uniformly.

4.3 Soundness

We divide the soundness analysis into three parts. First we analyze the case of an honest PV, and a cheating PP (Lemma 2). Then we show that if PV and PP pass the rigidity game with almost optimal probability, then one can construct new provers PV′ and PP′, with PV′ honest, such that the probability that they are accepted in the delegation game is not changed by much (Lemma 3). In Lemma 4, we combine the previous to derive the desired constant soundness-completeness gap, where we exclude that the acceptance probability of the provers in the rigidity game is too low by picking a p_r large enough.

Lemma 2 (Soundness against PP). *Suppose the verifier executes the delegation game on input $(Q, |x\rangle)$ such that $\|\Pi_0 Q|x\rangle\|^2 \leq 1/3$ with provers (PV, PP^*) such that* PV *plays the honest strategy. Then the verifier accepts with probability at most $7/9$.*

Proof. Let PP^* be any prover. Assume that PV behaves honestly and applies the measurements specified by his query W on halves of EPR pairs shared with PP^*. As a result the corresponding half-EPR pair at PP^* is projected onto the post-measurement state associated with the outcome reported by PV to V.

From PP^*, we define another prover, P^*, such that if P^* interacts with V_{EPR}, the honest verifer for the EPR Protocol (Figure 3a), then V_{EPR} rejects with the same probability that V would reject on interaction with PP^*. The main idea of the proof can be seen by looking at Figure 5, and noticing that: (1) the combined action of V and PV is unchanged if instead of choosing the W_i-values at random and then choosing z_i as a function of these, the z_i are chosen uniformly at random, and then the W_i are chosen as a function of these; and (2) with this transformation, the combined action of V and PV is now the same as the action of V_{EPR} in the EPR Protocol.

We now define P^*. P^* acts on a system that includes $n + t$ qubits that, in an honest run of the EPR Protocol, are halves of EPR pairs shared with V_{EPR}. P^* receives $\{z_i\}_{i=1}^t$ from V_{EPR}. P^* creates $m - (n + t)$ half EPR pairs (i.e. single-qubit maximally mixed states) and randomly permutes these with his $n + t$ unmeasured qubits, n of which correspond to computation qubits on systems $\mathcal{X}_1, \ldots, \mathcal{X}_n$ — he sets N to be the indices of these qubits — and t of which correspond to T-auxiliary states — he sets T^0 and T^1 to be the indices of these qubits. P^* simulates PP^* on these m qubits in the following way. First, P^* gives PP^* the index sets N, T^0, and T^1. In the ℓ-th iteration of the loop

(Step 2. in Figure 8), PP* returns some bits $\{c_i\}_{i\in T_\ell}$, and then expects inputs $\{z_i\}_{i\in T_\ell}$, which P^* provides, using the bits he received from V_{EPR}. Finally, at the end of the computation, PP* returns a bit c_f, and P^* outputs $\{c_i\}_{i\in T}$ and c_f.

This completes the description of P^*. To show the lemma we argue that for any input $(Q,|x\rangle)$ the probability that V outputs accept on interaction with PV and PP* is the same as the probability that V_{EPR} outputs accept on interaction with P^*, which is at most $\frac{2}{3}q_t+\frac{1}{3}q_c$ whenever $\|\Pi_0 Q|x\rangle\|^2\leq 1/3$, by Theorem 3. Using $\delta=\frac{1}{3}$, Theorem 3 gives $q_c\leq\frac{5}{3}-\frac{4}{3}q_t$, which yields

$$\frac{2}{3}q_t+\frac{1}{3}q_c\leq\frac{5}{9}+\frac{2}{9}q_t\leq\frac{7}{9}.$$

There are two reasons that V_{EPR} might reject: (1) in a computation or X-test round, the output qubit decodes to 1; or (2) in an evaluation of the gadget in Figure 5 (either an X-test round for an even T gate, or a Z-test round for an odd T gate) the condition $c_i=a_j\oplus e_i$ fails.

We first consider case (1). This occurs exactly when $c_f\oplus a_f=1$, where a_f is the final X key of the output wire, held by V_{EPR}. We note that a_f is exactly the final X key that V would hold in the Verifier-on-a-Leash Protocol, which follows from the fact that the update rules in both the EPR Protocol and the leash protocol are the same. Thus, the probability that V_{EPR} finds $v_f\oplus a_f=1$ on interaction with P^* is exactly the probability that V finds $c_f\oplus a_f=1$ in Step 5 of Figure 6.

Next, consider case (2). The condition $c_i\neq a_j\oplus e_i$ is exactly the condition in which a verifier interacting with P^* as in Figure 6 would reject (see Step 4.(b)).

Thus, the probability that V_{EPR} outputs reject upon interaction with P^* is exactly the probability that V outputs reject on interaction with PP*, which, as discussed above, is at most $7/9$.

The following lemma shows soundness against cheating PV*.

Lemma 3. *Suppose the verifier executes the leash protocol on input $(Q,|x\rangle)$ such that $\|\Pi_0 Q|x\rangle\|^2\leq 1/3$ with provers $(\mathrm{PV}^*,\mathrm{PP}^*)$, such that the provers are accepted with probability $1-\varepsilon$, for some $\varepsilon>0$, in the rigidity game, and with probability at least q in the delegation game. Then there exist provers PP' and PV' such that PV' applies the honest strategy and PP' and PV' are accepted with probability at least $q-\mathrm{poly}(\varepsilon)$ in the delegation game.*

Proof. By assumption, PP* and PV* are accepted in the rigidity game with probability at least $1-\varepsilon$. Let V_A, V_B be the local isometries guaranteed to exist by Theorem 4, and $\{\tau_\lambda\}$ the sub-normalized densities associated with PP*'s Hilbert space (recall that playing the rigidity game sequentially leaves the guarantees from Theorem 4 unchanged, since it only reduces the provers' ability to cheat).

First define provers PV$''$ and PP$''$ as follows. PP$''$ and PV$''$ initially share the state

$$|\psi'\rangle_{\mathsf{AB}}=\otimes_{i=1}^m|\mathrm{EPR}\rangle\langle\mathrm{EPR}|_{\mathsf{AB}}\otimes\sum_{\lambda\in\{\pm\}}|\lambda\rangle\langle\lambda|_{\mathsf{A}'}\otimes|\lambda\rangle\langle\lambda|_{\mathsf{B}'}\otimes(\tau_\lambda)_{\mathsf{A}''}\,,$$

with registers $\mathsf{AA}'\mathsf{A}''$ in the possession of PP$''$ and BB' in the possession of PV$''$. Upon receiving a query $W\in\Sigma^m$, PV$''$ measures B' to obtain a $\lambda\in\{\pm\}$. If $\lambda=+$

he proceeds honestly, measuring his half-EPR pairs exactly as instructed. If $\lambda = -$ he proceeds honestly except that for every honest single-qubit observable specified by W, he instead measures the complex conjugate observable. Note that this strategy can be implemented irrespective of whether W is given at once, as in the game RIGID, or sequentially, as in the Delegation Game. PP'' simply acts like PP^*, just with the isometry V_A applied.

First note that by Theorem 4, the distribution of answers of PV'' to the verifier, as well as the subsequent interaction between the verifier and PP, generate (classical) transcripts that are within statistical distance $\mathrm{poly}(\varepsilon)$ from those generated by PV^* and PP^* with the same verifier.

Next we observe that taking the complex conjugate of both provers' actions does not change their acceptance probability in the delegation game, since the interaction with the verifier is completely classical. Define PP' as follows: PP' measures A' to obtain the same λ as PV'', and then executes PP'' or its complex conjugate depending on the value of λ. Define PV' to execute the honest behavior (he measures to obtain λ, but then discards it and does not take any complex conjugates).

Then PV' applies the honest strategy, and $(\mathrm{PV}', \mathrm{PP}')$ applies either the same strategy as $(\mathrm{PV}'', \mathrm{PP}'')$ (if $\lambda = +$) or its complex conjugate (if $\lambda = -$). Therefore they are accepted in the delegation game with exactly the same probability.

Combining Lemma 2 and Lemma 3 gives us the final soundness guarantee.

Lemma 4. *(Constant soundness-completeness gap) There exist constants $p_r, p_d = 1 - p_r$ and $\Delta > 0$ such that if the verifier executes the leash protocol with parameters (p_r, p_d) on input $(Q, |\boldsymbol{x}\rangle)$ such that $\|\Pi_0 Q|\boldsymbol{x}\rangle\|^2 \leq 1/3$, any provers $(\mathrm{PV}^*, \mathrm{PP}^*)$ are accepted with probability at most $p_{\mathrm{sound}} = p_{\mathrm{compl}} - \Delta$.*

Proof. Suppose provers PP^* and PV^* succeed in the delegation game with probability $\frac{7}{9} + w$ for some $w > 0$, and the testing game with probability $1 - \varepsilon_*(w)$, where $\varepsilon_*(w)$ will be specified below. By Lemma 3, this implies that there exist provers PP' and PV' such that PV' is honest and the provers succeed in the delegation game with probability at least $\frac{7}{9} + w - g(\varepsilon_*(w))$, where $g(\varepsilon) = \mathrm{poly}(\varepsilon)$ is the function from the guarantee of Lemma 3. Let $\varepsilon_*(w)$ be such that $g(\varepsilon_*(w)) \leq \frac{w}{2}$. In particular, $\frac{7}{9} + w - g(\varepsilon_*(w)) \geq \frac{7}{9} + \frac{w}{2} > \frac{7}{9}$. This contradicts Lemma 2.

Thus if provers PP and PV succeed in the delegation game with probability $\frac{7}{9} + w$ they must succeed in the rigidity game with probability less than $1 - \varepsilon_*(w)$. This implies that for any strategy of the provers, on any *no* instance, the probability that they are accepted is at most

$$\max\Big\{p_r + (1-p_r)\Big(\frac{7}{9} + \frac{1}{18}\Big),\ p_r\Big(1 - \varepsilon_*\Big(\frac{1}{18}\Big)\Big) + (1-p_r)\cdot 1\Big\}. \tag{4}$$

Since $\varepsilon_*(\frac{1}{18})$ is a positive constant, it is clear that one can pick p_r large enough so that

$$p_r\Big(1 - \varepsilon_*\Big(\frac{1}{18}\Big)\Big) + (1-p_r)\cdot 1 < p_r + (1-p_r)\Big(\frac{7}{9} + \frac{1}{18}\Big). \tag{5}$$

Select the smallest such p_r. Then the probability that the two provers are accepted is at most

$$p_{\text{sound}} := p_r + (1-p_r)\Big(\frac{7}{9}+\frac{1}{18}\Big) < p_r\big(1-e^{-\Omega(n+t)}\big) + (1-p_r)\frac{8}{9} = p_{\text{compl}},$$

which gives the desired constant completeness-soundness gap Δ.

4.4 Blindness

We now establish blindness of the Leash Protocol. In Lemma 5, we will prove that the protocol has the property that neither prover can learn anything about the input to the circuit, x, aside from its length. Thus, the protocol can be turned into a blind protocol, where Q is also hidden, by modifying any input (Q,x) where Q has g gates and acts on n qubits, to an input $(U_{g,n},(Q,x))$, where $U_{g,n}$ is a universal circuit that takes as input a description of a g-gate circuit Q on n qubits, and a string x, and outputs $Q|x\rangle$. The universal circuit $U_{g,n}$ can be implemented in $O(g\log n)$ gates. By Lemma 5, running the Leash Protocol on $(U_{g,n},(Q,x))$ reveals nothing about Q or x aside from g and n.

In the form presented in Figure 6, the verifier V interacts first with PV, sending him random questions that are independent from the input x, aside from the input length n. It is thus clear that the protocol is blind with respect to PV.

In contrast, the questions to PP depend on PV's answers and on the input, so it may a priori seem like the questions can leak information to PP. To show that the protocol is also blind with respect to PP, we show that there is an alternative formulation, in which the verifier first interacts with PP, sending him random messages, and then only with PV, with whom the interaction is now adaptive. We argue that, for an arbitrary strategy of the provers, the reduced state of all registers available to either prover, PP or PV, is exactly the same in both formulations of the protocol — the *original* and the *alternative* one. This establishes blindness for both provers. This technique for proving blindness is already used in [RUV13] to establish blindness of a two-prover protocol based on computation by teleportation.

Lemma 5 (Blindness of the Leash Protocol). *For any strategy of* PV* *and* PP**, the reduced state of* PV* *(resp.* PP**) at the end of the leash protocol is independent of the input x, aside from its length.*

Proof. Let PV* and PP* denote two arbitrary strategies for the provers in the leash protocol. Each of these strategies can be modeled as a super-operator

$$\mathcal{T}_{\text{PV}} : \mathrm{L}(\mathcal{H}_{T_{\text{PV}}} \otimes \mathcal{H}_{\text{PV}}) \to \mathrm{L}(\mathcal{H}_{T'_{\text{PV}}} \otimes \mathcal{H}_{\text{PV}}),$$

$$\mathcal{T}_{\text{PP},ad} : \mathrm{L}(\mathcal{H}_{T_{\text{PP}}} \otimes \mathcal{H}_{\text{PP}}) \to \mathrm{L}(\mathcal{H}_{T'_{\text{PP}}} \otimes \mathcal{H}_{\text{PP}}).$$

Here $\mathcal{H}_{T_{\text{PV}}}$ and $\mathcal{H}_{T'_{\text{PV}}}$ (resp. $\mathcal{H}_{T_{\text{PP}}}$ and $\mathcal{H}_{T'_{\text{PP}}}$) are classical registers containing the inputs and outputs to and from PV* (resp. PP*), and $\mathcal{H}_{\text{PV}}$ (resp. $\mathcal{H}_{\text{PP}}$) is the private space of PV* (resp. PP*). Note that the interaction of each prover with the verifier is sequential, and we use $\mathcal{T}_{\text{PV}}$ and $\mathcal{T}_{\text{PP},ad}$ to denote the combined action of the prover and the verifier across all rounds of interaction (formally these are sequences of superoperators).

Consider an alternative protocol, which proceeds as follows. The verifier first interacts with PP. From Figure 8 we see that the inputs required for PP are subsets N and $T_1, \ldots, T_d$, and values $\{z_i\}_{i \in T_\ell}$ for each $\ell \in \{1, \ldots, d\}$. To select the former, the verifier proceeds as in the first step of the Delegation Game. She selects the latter uniformly at random. The verifier collects values $\{c_i\}_{i \in T_\ell}$ from PP exactly as in the original Delegation Game.

Once the interaction with PP has been completed, the verifier interacts with PV. First, she selects a random string $W_N \in \Sigma^N$, conditioned on the event that W_N contains at least n copies of each symbol in Σ, and sends it to PV, collecting answers $\boldsymbol{e}_N$. The verifier then follows the same update rules as in the delegation game. We describe this explicitly for computation rounds. First, the verifier sets $\boldsymbol{a} = \boldsymbol{e}_N$. Depending on the values $\{c_i\}_{i \in T_1}$ and $\{z_i\}_{i \in T_1}$ obtained in the interaction with PP, using the equation $z_i = a_j + 1_{W_i = F} + c_i$ she deduces a value for $1_{W_i=F}$ for each $i \in T_1 \subseteq B_1$. She then selects a uniformly random $W_{B_1} \in \Sigma^{B_1}$, conditioned on the event that W_{B_1} contains at least t_1 copies of each symbol from Σ, and for $i \in T_1$ it holds that $W_i = F$ if and only if $z_i = a_j + 1 + c_i$. The important observation is that, if T_1 is a uniformly random, unknown subset, the marginal distribution on W_{B_1} induced by the distribution described above is independent of whether $z_i = a_j + 1 + c_i$ or $z_i = a_j + 0 + c_i$: precisely, it is uniform conditioned on the event that W_{B_1} contains at least t_1 copies of each symbol from Σ. The verifier receives outcomes $\boldsymbol{e}_{B_1} \in \{0,1\}^{B_1}$ from PV, and using these outcomes performs the appropriate key update rules; she then proceeds to the second layer of the circuit, until the end of the computation. Finally, the verifier accepts using the same rule as in the last step of the original delegation game.

We claim that both the original and alternative protocols generate the same joint final state:

$$\mathcal{T}_{\mathrm{PP},ad} \circ \mathcal{T}_{\mathrm{PV}}(\rho_{orig}) = \mathcal{T}_{\mathrm{PV},ad} \circ \mathcal{T}_{\mathrm{PP}}(\rho_{alt}) \in \mathcal{H}_{\mathrm{PP}} \otimes \mathcal{H}_{T'_{\mathrm{PP}}} \otimes \mathcal{H}_{\mathrm{V}} \otimes \mathcal{H}_{T'_{\mathrm{PV}}} \otimes \mathcal{H}_{\mathrm{PV}}, \quad (6)$$

where we use ρ_{orig} and ρ_{alt} to denote the joint initial state of the provers, as well as the verifier's initialization of her workspace, in the original and alternative protocols respectively, and $\mathcal{T}_{\mathrm{PV},ad}$ and $\mathcal{T}_{\mathrm{PP}}$ are the equivalent of $\mathcal{T}_{\mathrm{PV}}$ and $\mathcal{T}_{\mathrm{PP},ad}$ for the reversed protocol (in particular they correspond to the same strategies PV^* and PP^* used to define $\mathcal{T}_{\mathrm{PV}}$ and $\mathcal{T}_{\mathrm{PP},ad}$). Notice that $\mathcal{T}_{\mathrm{PV},ad}$ and $\mathcal{T}_{\mathrm{PP}}$ are well-defined since neither prover can distinguish an execution of the original from the alternative protocol.[7] To see that equality holds in (6), it is possible to re-write the final state of the protocol as the result of the following sequence of operations. First, the verifier initializes the message registers with PP^* and PV^* using half-EPR pairs, keeping the other halves in her private workspace. This simulates the generation of uniform random messages to both provers. Then, the superoperator $\mathcal{T}_{\mathrm{PV}} \otimes \mathcal{T}_{\mathrm{PP}}$ is executed. Finally, the verifier post-selects by applying a projection operator on $\mathcal{H}_{T_{\mathrm{PV}}} \otimes \mathcal{H}_{T'_{\mathrm{PV}}} \otimes \mathcal{H}_{T_{\mathrm{PP}}} \otimes \mathcal{H}_{T'_{\mathrm{PP}}}$ that projects onto valid transcripts for the original protocol (i.e. transcripts in which the adaptive questions are chosen correctly). This projection can be implemented in two equivalent ways: either

[7] One must ensure that a prover does not realize if the alternative protocol is executed instead of the original; this is easily enforced by only interacting with any of the provers at specific, publicly decided times.

the verifier first measures $\mathcal{H}_{T_{\mathrm{PV}}} \otimes \mathcal{H}_{T'_{\mathrm{PV}}}$, and then $\mathcal{H}_{T_{\mathrm{PP}}} \otimes \mathcal{H}_{T'_{\mathrm{PP}}}$; based on the outcomes she accepts a valid transcript for the original protocol or she rejects. Or, she first measures $\mathcal{H}_{T_{\mathrm{PP}}} \otimes \mathcal{H}_{T'_{\mathrm{PP}}}$, and then $\mathcal{H}_{T_{\mathrm{PV}}} \otimes \mathcal{H}_{T'_{\mathrm{PV}}}$; based on the outcomes she accepts a valid transcript for the alternative protocol or she rejects. Using the commutation of the provers' actions, conditioned on the transcript being accepted, the first gives rise to the first final state in (6), and the second to the second final state. The two are equivalent because the acceptance condition for a valid transcript is identical in the two versions of the protocol.

Since in the first case the reduced state on $\mathcal{H}_{T'_{\mathrm{PV}}} \otimes \mathcal{H}_{\mathrm{PV}}$ is independent of the input to the computation, $\boldsymbol{x}$, and in the second the reduced state on $\mathcal{H}_{\mathrm{PP}} \otimes \mathcal{H}_{T'_{\mathrm{PP}}}$ is independent of $\boldsymbol{x}$, we deduce that the protocol hides the input from each of PV^* and PP^*.

Remark 1. In order to make a fair comparison between previous delegated computation protocols and ours (see Figure 1), one must analyze their resource requirements under the condition that they produce the correct outcome of the computation with a fixed, let us say 99%, probability. For most protocols, this is achieved by sequentially repeating the original version, in order to amplify the completeness-soundness gap. We refer to the full version of the paper for a sequencial procedure that allows the verifier to obtain the correct output with a fixed probability (or abort whenever the provers are malicious).

5 Dog-Walker protocol

The Dog-Walker Protocol again involves a classical verifier V and two provers PV and PP. As in the leash protocol presented in Section 4, PP and PV take the roles of P_{EPR} and V_{EPR} from [Bro18] respectively. The main difference is that the Dog-Walker Protocol gives up blindness in order to reduce the number of rounds to two (one round of interaction with each prover, played sequentially). After one round of communication with PP, who returns a sequence of measurement outcomes, V communicates all of PP's outcomes, except for the one corresponding to the output bit of the computation, as well as the input $\boldsymbol{x}$, to PV. With these, PV can perform the required adaptive measurements without the need to interact with V. It may seem risky to communicate bits sent by PP directly to PV — this seems to allow for communication between the two provers! Indeed, blindness is lost. However, if PP is honest, his outcomes $\{c_i\}_i$ in the computation round are the result of measurements he performs on half-EPR pairs, and are uniform random bits. If he is dishonest, and does not return the outcomes obtained by performing the right measurements, he will be caught in the test rounds. It is only in computation rounds that V sends the measurement results $\{c_i\}_i$ to PV.

We notice that PV has a much more important role in this protocol: he decides himself the measurements to perform according to previous measurements' outcomes as well as the input x. For this reason, we must augment the test discussed in Section 3 in order to test if PV remains honest with respect to these new tasks. For this reason, we introduce the Tomography test and prove a rigidity theorem that will allow us to prove the soundness of the Dog-walker protocol (see Figure 10 for a glimpse of the proof structure). Due to space limitations we refer to the full version of the paper for a

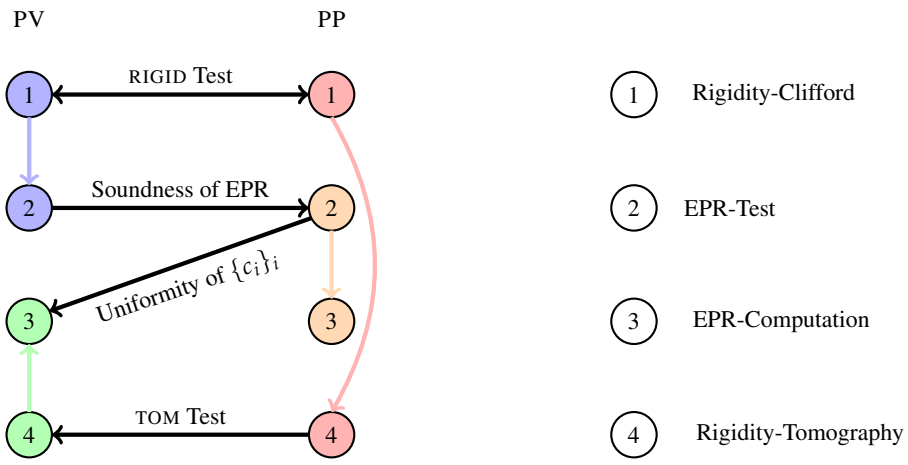


Fig. 10: Overview of the soundness of the Dog-Walker Protocol

presentation of the Tomography Test, a formal description of the Dog-walker protocol and the proof for their correctness.

Finally, the Dog-Walker Protocol can be easily extended to a classical-verifier two-prover protocol for all languages in QMA. Along the same lines of the proof that QMIP = MIP* from [RUV13], one of the provers plays the role of PP, running the QMA verification circuit, while the second prover creates and teleports the corresponding QMA witness. In our case, it is not hard to see that the second prover can be re-used as PV in the Dog-Walker Protocol, creating the necessary gadgets for the computation and allowing the Verifier to check the operations performed by the first prover. We describe this approach in more details in the full version of the paper.

References

ABE10. Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In *Proceedings of the first Symposium on Innovations in Computer Science (ICS 2010)*, pages 453–469, 2010.

ADSS17. Gorjan Alagic, Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum fully homomorphic encryption with verification. *arXiv preprint arXiv:1708.09156*, 2017.

Bel64. John S. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1:195–200, 1964.

BFGH10. Debajyoti Bera, Stephen A. Fenner, Frederic Green, and Steven Homer. Efficient universal quantum circuits. *Quantum Information & Computation*, 10(1&2):16–27, 2010.

Bro18. Anne Broadbent. How to verify a quantum computation. *Theory of Computing*, 14(11):1–37, 2018. arXiv preprint `arXiv:1509.09180`.

BvCA18. Joseph Bowles, Ivan Šupić, Daniel Cavalcanti, and Antonio Acín. Self-testing of Pauli observables for device-independent entanglement certification, 2018. arXiv:1801.10446.

Cas17. Davide Castelvecchi. IBM's quantum cloud computer goes commercial. *Nature News*, 543(7644), 6 March 2017.

CHSH69. John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23:880–884, 1969.

DSS16. Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *Advances in Cryptology – Proceedings of the 36th Annual International Cryptology Conference (CRYPTO 2016)*, pages 3–32, 2016. arXiv:1603.09717.

FH15. Joseph F. Fitzsimons and Michal Hajdušek. Post hoc verification of quantum computation, 2015. arXiv preprint `arXiv:1512.04375`.

FH17. Keisuke Fujii and Masahito Hayashi. Verifiable fault tolerance in measurement-based quantum computation. *Physical Review A*, 96:030301, Sep 2017.

Fit16. Joseph F. Fitzsimons. Private quantum computation: An introduction to blind quantum computing and related protocols, 2016. arXiv preprint `arXiv:1611.10107`.

FK17. Joseph F. Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Physical Review A*, 96(012303), 2017. arXiv preprint `arXiv:1203.5217`.

GKW15. Alexandru Gheorghiu, Elham Kashefi, and Petros Wallden. Robustness and device independence of verifiable blind quantum computing. *New Journal of Physics*, 17, 2015.

Gri17. Alex B. Grilo. Relativistic verifiable delegation of quantum computation, 2017. arXiv preprint arXiv:1711.09585.

HH16. Masahito Hayashi and Michal Hajdušek. Self-guaranteed measurement-based quantum computation, 2016. arXiv preprint `arXiv:1603.02195`.

HM15. Masahito Hayashi and Tomoyuki Morimae. Verifiable measurement-only blind quantum computing with stabilizer testing. *Physical Review Letters*, 115:220502, Nov 2015.

HPDF15. Michal Hajdušek, Carlos A. Pérez-Delgado, and Joseph F. Fitzsimons. Device-independent verifiable blind quantum computation, 2015. arXiv preprint `arXiv:1502.02563`.

HZM+17. He-Liang Huang, Qi Zhao, Xiongfeng Ma, Chang Liu, Zu-En Su, Xi-Lin Wang, Li Li, Nai-Le Liu, Barry C. Sanders, Chao-Yang Lu, and Jian-Wei Pan. Experimental Blind Quantum Computing for a Classical Client. *Physical Review Letters*, 119:050503, Aug 2017.

Ji16. Zhengfeng Ji. Classical verification of quantum proofs. In *Proceedings of the Forty-eighth Annual ACM SIGACT Symposium on Theory of Computing (STOC 2016)*, pages 885–898, 2016.

Mah17. Urmila Mahadev. Classical homomorphic encryption for quantum circuits. *arXiv preprint arXiv:1708.02130*, 2017.

Mah18. Urmila Mahadev. Classical verification of quantum computations. *arXiv preprint arXiv:1804.01082*, 2018.

McK16. Matthew McKague. Interactive proofs for BQP via self-tested graph states. *Theory of Computing*, 12(3):1–42, 2016. arXiv preprint `arXiv:1309.5675`.

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

Mon16. Ashely Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(15023), 2016.

Mor14. Tomoyuki Morimae. Verification for measurement-only blind quantum computing. *Physical Review A*, 89, 2014.

MTH17. Tomoyuki Morimae, Yuki Takeuchi, and Masahito Hayashi. Verified measurement-based quantum computing with hypergraph states, 2017. arXiv:1701.05688.

MY04. Dominic Mayers and Andrew Yao. Self testing quantum apparatus. *Quantum Information & Computation*, 4:273–286, 2004.

NV17. Anand Natarajan and Thomas Vidick. A quantum linearity test for robustly verifying entanglement. In *Proceedings of the Forty-ninth Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*, pages 1003–1015, 2017.

RUV12. Ben W. Reichardt, Falk Unger, and Umesh Vazirani. A classical leash for a quantum system: Command of quantum systems via rigidity of CHSH games, 2012. arXiv preprint `arXiv:1209.0448`.

RUV13. Ben W. Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496:456–460, 2013. Full version `arXiv:1209.0448`.

Slo16. William Slofstra. Tsirelson's problem and an embedding theorem for groups arising from non-local games, 2016. arXiv preprint `arXiv:1606.03140`.