Verification of Quantum Computation and the Price of Trust

Verification of quantum computation and the price of trust

Alexandru Gheorghiu*[1], Theodoros Kapourniotis†[1,2], and Elham Kashefi‡[1,3]

[1]*School of Informatics, University of Edinburgh, UK*
[2]*Department of Physics, University of Warwick, UK*
[3]*CNRS LIP6, Université Pierre et Marie Curie, Paris*

Quantum computers promise to efficiently solve not only problems believed to be intractable to classical computers [1], but also problems for which verifying the solution is considered intractable [2]. In particular, there are problems in the complexity class BQP, i.e. solvable in polynomial time by a quantum computer, that are believed to be outside of NP, the class of problems for which checking the solution can be performed in polynomial time by a classical computer. This raises the question of how one can verify whether quantum computers are indeed producing correct results. Answering this question leads to *quantum verification*, which has been highlighted as a significant challenge on the road to scalable quantum computing technology. Verification is pertinent to both medium-sized quantum computers, expected to be developed in under a decade, but also to future quantum cloud supercomputers used by remote users. It is also relevant for experiments of quantum mechanics, where the size of the system involved is beyond the regime of classical simulation. In this paper we attempt to categorize the different methods of quantum verification that have appeared in recent years. Since most of them are based on cryptographic primitives and treat quantum devices as untrusted entities, we highlight a general trade-off between trust assumptions and complexity.

The setting in which quantum verification has been studied extensively is that of *interactive proof systems*. This involves two distinct entities: a trusted party called the *verifier* (also known as *client*), tasked with verifying the correctness of a computation and an untrusted party called the *prover* (also known as *server*), who runs the computation and attempts to convince the verifier of the result. Formally, for some language $L \in \mathsf{BQP}$ the verifier wants to know, for an input x, whether $x \in L$ or $x \notin L$. The prover is trying to convince the verifier that one of these statements is true usually by demonstrating that it has performed the correct quantum computation. To ensure this, in a typical run of a verification protocol, the verifier asks the prover to not only perform the quantum computation, but also a series of trials that will be used to test his behaviour. Cryptographic methods are applied so that the prover cannot distinguish the tests from the computation and try to cheat selectively. This class of protocols constitutes the majority

*Email: a.gheorghiu@sms.ed.ac.uk
†Email: t.kapourniotis@warwick.ac.uk
‡Email: ekashefi@inf.ed.ac.uk

of verification protocols developed so far. For this reason, in our paper, we will primarily review these types of approaches. It is worth mentioning that all techniques reviewed in this paper assume that the prover can deviate in any possible way that follows the laws of quantum mechanics.

Essential to the effectiveness of a verification protocol is the ascription of trust to some of the used devices. Ideally, one wants to restrict the trust to the classical computer which the verifier controls. However, all existing approaches require some extra trust assumptions on the quantum devices or the channels involved in the protocol. For instance, protocols in which the verifier interacts with a single quantum prover require the verifier to posses a trusted quantum device. If there is more than one prover, the verifier can indeed be fully classical, but then the provers are forbidden from interacting with each other. Our goal in this paper is to highlight the trade-off between the trust assumptions of each verification technique and the required resources to achieve the same level of confidence in the verification.

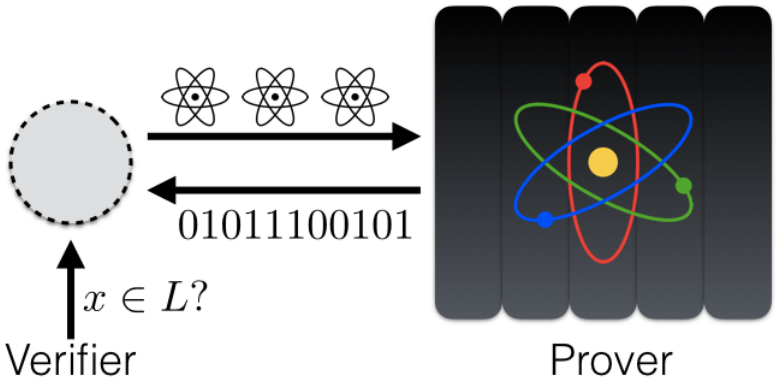

Figure 1: Prepare and send verification protocol

We proceed by first considering protocols which make use of cryptographic primitives and have *information-theoretic security*. These protocols are divided into two broad categories:

1. *Prepare and send/receive and measure protocols.* These are protocols in which the verifier and the prover exchange qubits through some quantum channel. As the name suggests, the verifier either prepares and sends qubits to the prover [3, 4, 5, 6, 7, 8] or, alternatively, receives qubits from the prover and measures them [9, 10, 11]. In the first case, the verifier relies on the uncertainty principle and the no-cloning theorem to ensure that the prover cannot distinguish tests from computations. In the second case, the verifier uses a type of cut-and-choose technique to decide whether to test the prover or perform a computation using the received states. In both cases, the essential element is the fact that the prover is oblivious to some part of the delegated computation. This property is commonly referred to as *blindness* [12, 13, 14, 15, 16, 17, 18, 19] and is a shared feature of most verification protocols. A schematic illustration of a prepare and send protocol is shown in Figure 1.

2. *Entanglement-based protocols.* These are protocols in which entangled states are shared either between the verifier and the prover [20, 21] or between multiple provers [22, 23]. One of the main reasons for considering the entanglement-based setting

is because it can lead to *device-independent* verification. In other words, because of the remarkable properties of non-local correlations, it is possible to verify a quantum computation in a situation in which all quantum devices are untrusted. It is, however, necessary to assume that the quantum devices sharing entanglement are not communicating throughout the protocol. In this case, the verifier needs to test not only the prover performing the computation, but also any other quantum device that is sharing entanglement. Depending on the trust assumptions about the shared entangled states as well as the measurement devices we notice different scalings for the communication complexity of the protocols as we show in the table below.

Entanglement / **Measurements**	*Trusted*	*Semi-trusted*	*Untrusted*
Trusted	$O(N)$	$O(N^2)$	$O(N^{13}log(N))$
Untrusted	$O(N^2)$	$O(N^2)$	$O(N^{64})$

We then also consider protocols which are not based in cryptography, but are more akin to quantum state certification. These are known as *post-hoc* verification protocols [24, 25, 26] and can also be categorized as either receive and measure or entanglement-based. While the cryptographic protocols aim to test the operations performed by the prover(s) towards achieving universal quantum computation, post-hoc protocols simply check quantum witnesses for decision problems. In other words, deciding whether some input x belongs or not to a language $L \in \mathsf{BQP}$ reduces to performing a two-outcome measurement of a quantum witness state $|\psi\rangle$. The protocols either have the prover send this state to the verifier to be measured, or the verifier coordinates a set of entangled provers to prepare and measure $|\psi\rangle$.

In both the previously mentioned cryptographic and post-hoc protocols, there are no limiting assumptions about the computational powers of the provers. In other words, even though we regard them as BQP machines, verification is possible even if the provers are computationally unbounded. Recently, however, verification protocols have been proposed for settings in which the provers are limited to a sub-universal model of quantum computations. The two that we review are for the *one-pure-qubit model* [27] and the *instantaneous quantum polynomial-time model* (or IQP) [28, 29].

Lastly, we address the issue of *fault-tolerance* [30]. This entails the ability to perform verification in a setting where quantum devices and quantum states are subject to noise that scales with the size of the system. Achieving fault-tolerant verification is crucial for the practical applicability of these protocols and their use in near-future experiments of *quantum supremacy* (attempting to demonstrate the "supraclassical power" of quantum computing).

By categorizing and analysing the resources required in each protocol, while at the same time making the trust assumptions explicit, we illustrate the bigger picture of quantum verification in the delegated setting. This highlights the significant overlap between quantum computation, cryptography and complexity theory and can serve as a guide for the development and improvement of future protocols.

References

[1] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[2] Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the jones polynomial. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 427–436. ACM, 2006.

[3] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In *Proceedings of Innovations in Computer Science 2010*, ICS2010, pages 453–, 2010.

[4] Joseph F Fitzsimons and Elham Kashefi. Unconditionally verifiable blind computation. *arXiv preprint arXiv:1203.5217*, 2012.

[5] Anne Broadbent. How to verify a quantum computation. *arXiv preprint arXiv:1509.09180*, 2015.

[6] Stefanie Barz, Joseph F Fitzsimons, Elham Kashefi, and Philip Walther. Experimental verification of quantum computation. *Nature Physics*, 9(11):727–731, 2013.

[7] Theodoros Kapourniotis, Vedran Dunjko, and Elham Kashefi. On optimising quantum communication in verifiable quantum computing. *arXiv preprint arXiv:1506.06943*, 2015.

[8] Elham Kashefi and Petros Wallden. Optimised resource construction for verifiable quantum computation. *arXiv preprint arXiv:1510.07408*, 2015.

[9] Tomoyuki Morimae. Measurement-only verifiable blind quantum computing with quantum input verification. *arXiv preprint arXiv:1606.06467*, 2016.

[10] Masahito Hayashi and Tomoyuki Morimae. Verifiable measurement-only blind quantum computing with stabilizer testing. *arXiv preprint arXiv:1505.07535*, 2015.

[11] Masahito Hayashi and Michal Hajdusek. Self-guaranteed measurement-based quantum computation. *arXiv preprint arXiv:1603.02195*, 2016.

[12] Pablo Arrighi and Louis Salvail. Blind quantum computation. *International Journal of Quantum Information*, 4(05):883–898, 2006.

[13] A. Childs. Secure assisted quantum computation. *Quant. Inf. Compt.*, 5(6):456, 2005.

[14] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 517–526. IEEE, 2009.

[15] Atul Mantri, Carlos A Perez-Delgado, and Joseph F Fitzsimons. Optimal blind quantum computation. *Physical review letters*, 111(23):230502, 2013.

[16] Carlos A Perez-Delgado and Joseph F Fitzsimons. Overcoming efficiency constraints on blind quantum computation. *arXiv preprint arXiv:1411.4777*, 2014.

[17] Vittorio Giovannetti, Lorenzo Maccone, Tomoyuki Morimae, and Terry G Rudolph. Efficient universal blind quantum computation. *Physical review letters*, 111(23):230501, 2013.

[18] Tomoyuki Morimae and Keisuke Fujii. Blind quantum computation protocol in which alice only makes measurements. *Physical Review A*, 87(5):050301, 2013.

[19] Takahiro Sueki, Takeshi Koshiba, and Tomoyuki Morimae. Ancilla-driven universal blind quantum computation. *Physical Review A*, 87(6):060301, 2013.

[20] Alexandru Gheorghiu, Elham Kashefi, and Petros Wallden. Robustness and device independence of verifiable blind quantum computing. *arXiv preprint arXiv:1502.02571*, 2015.

[21] Michal Hajdusek, Carlos A Perez-Delgado, and Joseph F Fitzsimons. Device-independent verifiable blind quantum computation. *arXiv preprint arXiv:1502.02563*, 2015.

[22] Ben W Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496(7446):456–460, 2013.

[23] Matthew McKague. Interactive proofs for BQP via self-tested graph states, 2013. arXiv:1309.5675.

[24] Joseph F Fitzsimons and Michal Hajdušek. Post hoc verification of quantum computation. *arXiv preprint arXiv:1512.04375*, 2015.

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

[26] D Hangleiter, M Kliesch, M Schwarz, and J Eisert. Direct certification of a class of quantum simulations. *arXiv preprint arXiv:1602.00703*, 2016.

[27] Theodoros Kapourniotis, Elham Kashefi, and Animesh Datta. Blindness and verification of quantum computation with one pure qubit. In *9th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2014)*, volume 27, pages 176–204, 2014.

[28] Daniel Mills, Anna Pappa, Theodoros Kapourniotis, and Elham Kashefi. Information theoretically secure hypothesis test for temporally unstructured quantum computation. *Ongoing work*, 2017.

[29] Theodoros Kapourniotis and Animesh Datta. Nonadaptive fault-tolerant verification of quantum supremacy with noise. *Ongoing work*, 2017.

[30] Peter W Shor. Fault-tolerant quantum computation. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 56–65. IEEE, 1996.