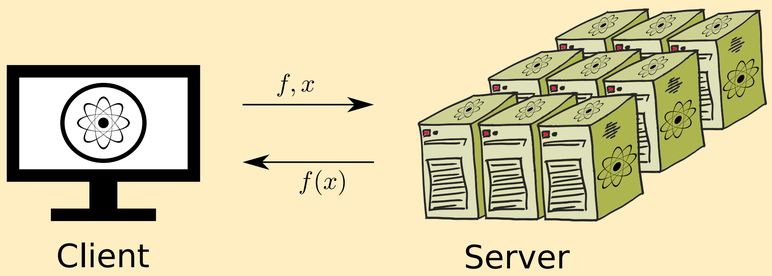
Verifying BQP Computations on Noisy Devices with Minimal Overhead

full paper to be published in PRX Quantum

Dominik Leichtle, Luka Music, Elham Kashefi, Harold Ollivier

Delegated Quantum Computing



The future availability of quantum computing through remotely accessible servers pose peculiar challenges: Clients with quantum-limited capabilities want their data and algorithms to remain hidden (*blindness*), while being able to verify that their computations are performed correctly (*verifiability*).

However, available techniques [FK12] suffer not only from high overheads but also from oversensitivity: When running on noisy devices, imperfections trigger the same detection mechanisms as malicious attacks, resulting in perpetually aborted computations. Hence, while malicious quantum computers are rendered harmless by blind and verifiable protocols, inherent noise severely limits their usability.

Our proposal

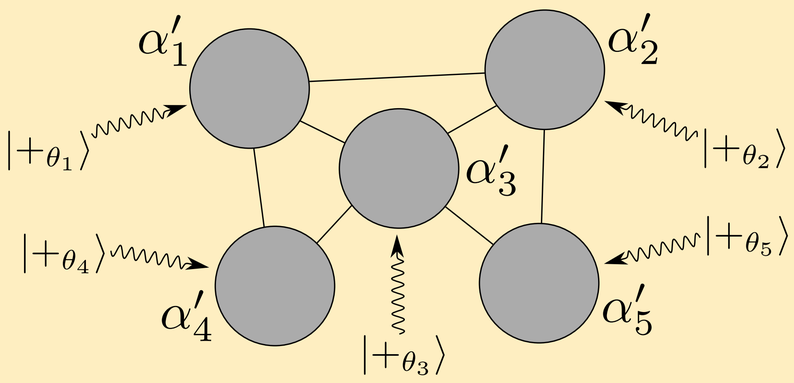
We present an **efficient, robust, blind, verifiable** scheme to delegate bounded-error quantum computations with classical inputs and outputs.

While our protocol requires quantum communication and the client's ability to generate single-qubit states, there is **no overhead on the physical resources** in terms of memory and gates.

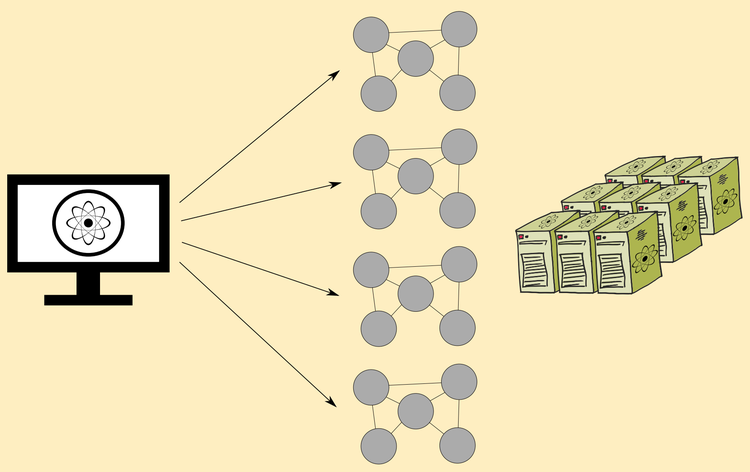
Our protocol does neither attempt nor rely on scalable quantum fault-tolerance.

How to get blindness

Our protocol is set in the Measurement-Based Quantum Computing (MBQC) model and uses the blindness idea of the Universal Blind Quantum Computing (UBQC) protocol [BFK09].



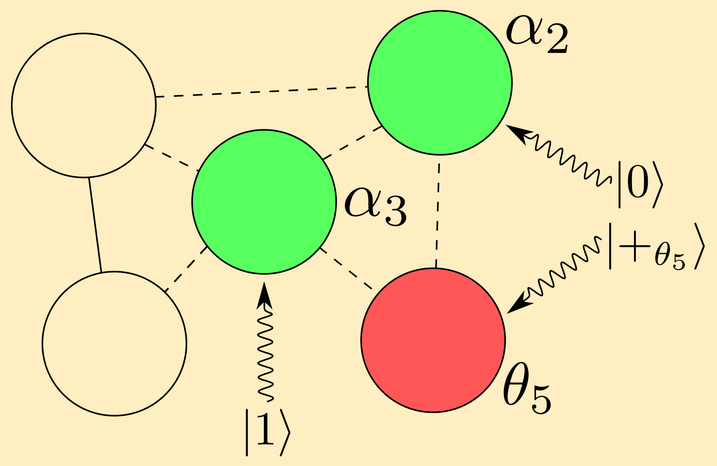
How to get robustness



It turns out that a *classical repetition code* is sufficient to protect against small noise.

How to get verifiability

The computation graph is replaced by a test graph of the same topology containing *traps*.

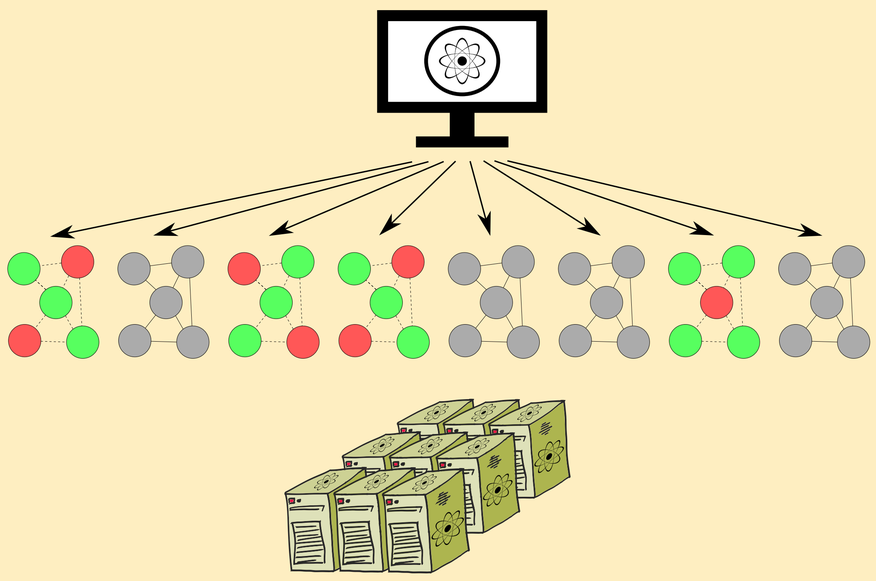


The protocol

Parameters:

- Number of test runs t
- Number of computation runs d
- Threshold of tolerated failed test runs w

There are k different types of test runs whose traps cover the whole graph. Generally, k equals the chromatic number of the graph; for the universal brickwork state $k = 2$.



Instructions:

- The client samples a random permutation of the d computation runs and t test runs.
- For every computation run, the client instructs the server to perform the original computation, using fresh randomness each time for blindness.
- For every test run, the client chooses a test run type at random, instructs the server to perform it, and checks the reported measurement outcome for consistency.
- If more than w test runs failed, the client returns REJECT, otherwise the client determines the accepted output by majority vote over the results of the computation runs.

Security and robustness of the protocol

Let p be the inherent error probability of the delegated BQP computation.

Theorem 1 (Security)**.** If $w/t < (2p-1)/k(2p-2)$, the protocol is secure in the *Constructive Cryptography* framework with **perfect blindness** and **exponential verification** against unbounded and fully malicious servers.

We quantify the global noise with the parameter p_{max} equal to the maximum probability that a randomly chosen test run fails.

Theorem 2 (Noise robustness)**.** On a setup with $p_{max} < w/t$, the protocol rejects at most with negligible probability.

References

- Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi, *Universal blind quantum computation*, Proceedings of the 50th Annual Symposium on Foundations of Computer Science, FOCS '09, IEEE Computer Society, 2009, p. 517.
- Joseph F. Fitzsimons and Elham Kashefi, *Unconditionally verifiable blind computation*, 2012, Eprint:http://arxiv.org/abs/1203.5217.