Information Theoretically Secure Hypothesis Test for Temporally Unstructured Quantum Computation
(Extended Abstract)

Daniel Mills

School of Informatics, University of Edinburgh, UK

daniel.mills@ed.ac.uk

Anna Pappa

School of Informatics, University of Edinburgh,UK

Department of Physics, University College London, UK

annapappa@gmail.com

Theodoros Kapourniotis

School of Informatics, University of Edinburgh, UK

Department of Physics, University of Warwick, UK

T.Kapourniotis@warwick.ac.uk

Elham Kashefi

School of Informatics, University of Edinburgh, UK

LIP6, CNRS, Pierre et Marie Curie University, France

ekashefi@inf.ed.ac.uk

The efficient certification of classically intractable quantum devices has been a central research question for some time. However, to observe a "quantum advantage", it is believed that one does not need to build a large scale universal quantum computer, a task which has proven extremely challenging. Intermediate quantum models that are easier to implement, but which also exhibit this quantum advantage over classical computers, have been proposed. In this work, we present a certification technique for such a sub-universal quantum server which only performs commuting gates and requires very limited quantum memory. By allowing a verifying client to manipulate single qubits, we exploit properties of measurement based blind quantum computing to give them the tools to test the "quantum superiority" of the server.

1 Introduction

Quantum computers are believed to be able to efficiently simulate quantum systems [15, 19] and outperform classical systems for specific tasks. Examples include Shor's algorithm for prime factorisation [37], Grover's algorithm for unstructured search [20], and the BB84 [5] and Ekert91 [14] protocols for public key exchange. That said, it may be some time before a large scale universal quantum computer capable of demonstrating the computational power of these protocols is built. In the meantime, several intermediate, non-universal models of quantum computation, which are still believed to not be classically simulatable, may prove easier to implement. Examples include the one clean qubit model [25, 28], the boson sampling model [1, 18, 30] and the Ising model [17, 29]. The *Instantaneous Quantum Poly-time* (IQP) machine [36] is another such non-universal model with significant practical advantages [2, 6]. IQP uses only commuting gates but is believed to remain hard to classically simulate [6–8] even in a noisy environment [6, 9]. Providing evidence that a machine can perform hard IQP computations would be a proof of *quantum superiority* [33] before a universal quantum computer has been realised experimentally.

In [36], the authors present a *hypothesis test* which can be passed only by devices capable of performing hard IQP computations. In order to accommodate a purely classical client, computational assumptions (conjecturing the hardness of finding hidden sub-matroids) are required in order to prove quantum superiority. In the present work, by endowing the client with the ability to perform simple qubit manipulations similar to those used in Quantum Key Distribution schemes [5], we develop an information-theoretically secure hypothesis test for IQP.

The remainder of the paper proceeds as follows. In Section 2, we formally introduce the IQP machine and provide an implementation in Measurement Based Quantum Computing (MBQC) [34, 35] which is more suitable for proving security in our framework than previous ones [22,36]. In Section 3 we use tools from blind quantum computing [10,16] to derive a delegated protocol for IQP computations which keeps the details of the computation hidden from the device performing it. We prove information-theoretical security of that scheme in a composable framework. Finally, in Section 4 we develop a hypothesis test that a limited quantum client can run to certify the quantum superiority of an untrusted Server.

2   Preliminaries

2.1   X-programs

The IQP machine introduced in [36] is defined by its capacity to implement X-programs.

Definition 2.1. *An X-program consists of a Hamiltonian comprised of a sum of products of X operators on different qubits, and $\theta \in [0, 2\pi]$ describing the time for which it is applied. The hth term of the sum has a corresponding vector $\mathbf{q}_h \in \{0,1\}^{n_p}$, called a* program element, *which defines on which of the n_p input qubits, the product of X operators which constitute that term, acts. The vector $\mathbf{q}_h$ has 1 in the j-th position when X is applied on the j-th qubit.*

As such, we can describe the X-program using θ and the matrix $\mathbf{Q} = (\mathbf{Q}_{hj}) \in \{0,1\}^{n_a \times n_p}$ which has as rows the program elements $\mathbf{q}_h$, $h = 1, \ldots, n_a$.

Applying the X-program discussed above to the computational basis state $|0^{n_p}\rangle$ and measuring the result in the computational basis can also be viewed as a quantum circuit with input $|0^{n_p}\rangle$, comprised of gates diagonal in the Pauli-X basis, and classical output. Using the random variable X to represent the distribution of output samples, the probability distribution of outcomes $\widetilde{x} \in \{0,1\}^{n_p}$ is:

$$\mathbb{P}\left(\mathtt{X} = \widetilde{x}\right) = \left| \langle \widetilde{x} | \exp\left(\sum_{h=1}^{n_a} i\theta \bigotimes_{j:\mathbf{Q}_{hj}=1} X_j \right) |0^{n_p}\rangle \right|^2 \tag{1}$$

Definition 2.2. *Given some X-program, an* IQP machine *is any computational method capable of efficiently returning a sample $\widetilde{x} \in \{0,1\}^{n_p}$ from the probability distribution* (1).

2.2   IQP In MBQC

A common framework for studying quantum computation is the MBQC model [34,35], where a quantum operation is expressed by a set of measurement angles on an entangled state described by a graph. This entangled state is built by applying a controlled-Z operation between qubits when there is an edge in the corresponding graph. The probabilistic nature of the measurements of the qubits in this state introduces some randomness which may be corrected for by adjusting the angle of measurement of subsequent qubits depending on the outcomes of the already performed measurements. The entangling, measuring and correcting operations on a set of qubits are usually referred to as a *measurement pattern* [11, 12].

In this work, we will deal with a specific type of graphs, given in the following definition:

Definition 2.3. *We define the* IQP graph *of an X-program $(\mathbf{Q}, \theta)$, as the graph with biadjacency matrix $\mathbf{Q} = (\mathbf{Q}_{hj}) \in \{0,1\}^{n_a \times n_p}$. This means that there is a bipartition of vertices into two sets P and A of cardinality n_p and n_a and that an edge exists in the graph between vertex a_h of set A and vertex p_j of set P when $\mathbf{Q}_{hj} = 1$.*
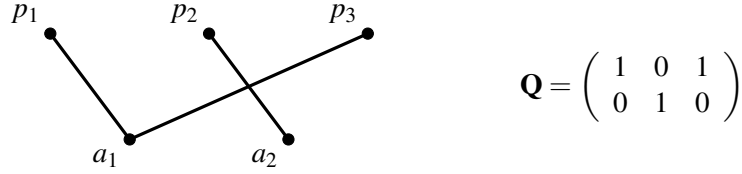
Figure 1: An example of an IQP graph described by matrix $\mathbf{Q}$. Here, $n_p = 3$ and $n_a = 2$ while the partition used is $P = [p_1, p_2, p_3]$ and $A = [a_1, a_2]$.

In what follows, we will denote with $\mathbf{Q}$ both the matrix of the X-program and its corresponding IQP graph (see Figure 1 for an example). The sets of vertices $A = \{a_1, ..., a_{n_a}\}$ and $P = \{p_1, ..., p_{n_p}\}$ will be called *primary* and *ancillary vertices* respectively. A result which is vital to the remainder of this paper, is the following:

Lemma 2.1. *A measurement pattern can always be designed to simulate an X-program efficiently.*

One can prove [27] that the distribution of (1) may be achieved by initialising n_p *primary qubits* in the states $\left|p_j\right\rangle = \left|+\right\rangle$, n_a *ancillary qubits* in the states $\left|a_h\right\rangle = \left|+\right\rangle$, applying Controlled-$Z$ operations between qubits when there is an edge in the IQP graph described by the X-program matrix $\mathbf{Q}$ and measuring the resulting state. We form this proof by demonstrating that sampling from the distribution in equation (1) is equivalent to inputting the primary qubits into a circuit made from composing circuits like the one in Figure 2 (one for each term of the sum in equation (1)) and measuring the result in the Hadamard basis. We then argue that all measurements may be delayed to the end of the composed circuit while the ancillary measurement basis is:

$$\{\left|0_\theta\right\rangle, \left|1_\theta\right\rangle\} = \left\{ \frac{1}{\sqrt{2}}\left(e^{-i\theta}\left|+\right\rangle + e^{i\theta}\left|-\right\rangle\right), \frac{1}{\sqrt{2}}\left(e^{-i\theta}\left|+\right\rangle - e^{i\theta}\left|-\right\rangle\right) \right\} \tag{2}$$
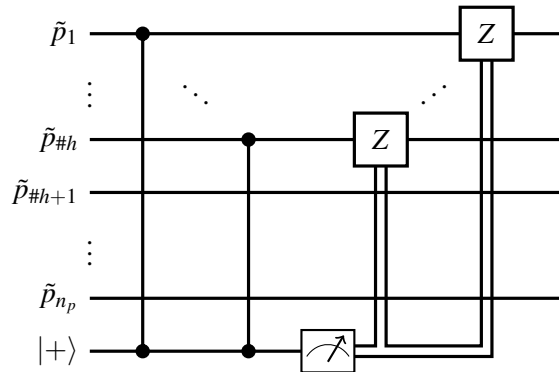

Figure 2: The circuit implementing one term in the sum of equation (1). The input qubits $\{p_j\}_{j=1}^{n_p}$ are rearranged so that if #h is the Hamming weight of row h of matrix $\mathbf{Q}$, then for $k = 1, ..., $#$h$ each $\tilde{p}_k$ corresponds to one p_j such that $\mathbf{Q}_{hj} = 1$ and for $k = $#$h+1, ..., n_p$ they correspond to the ones such that $\mathbf{Q}_{hj} = 0$. The ancillary qubit measurement is in the basis $\{\left|0_\theta\right\rangle, \left|1_\theta\right\rangle\}$ defined in expression (2).

3   Blind Delegated IQP Computation

We now move to build a method for blindly performing an IQP computation in a delegated setting. We consider a client with limited quantum power delegating an IQP computation to a powerful Server. The novel method that we use in this work is to keep the *X*-program secret by concealing the quantum state used. Intuitively, this is done by the client asking the Server to produce a quite general quantum state and then 'move' from that one to the one that is required for the computation. If this is done in a blind way then the Server only has knowledge of the general starting state from which any number of other quantum states may have been built. Hence, there are three key problems to be addressed:

1. How to transform one state to another.

2. Which general quantum state to build and transform into the one required for the IQP computation.

3. How to do this transformation secretly in a delegated setting.

3.1   Break and Bridge

The break and bridge operations [16,21] on a graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$, with vertex set $\widetilde{V}$ and edge set $\widetilde{E}$ describe the operations necessary to solve problem 1.

Definition 3.1. *The* break *operator acts on a vertex* $v \in \widetilde{V}$ *of degree 2 in a graph* $\widetilde{G}$. *It removes v from* $\widetilde{V}$ *and also removes any edges connected to v from* $\widetilde{E}$.

The bridge *operator also acts on* $v \in \widetilde{V}$ *of degree 2 in a graph* $\widetilde{G}$. *It removes v from* $\widetilde{V}$, *removes edges connected to v from* $\widetilde{E}$ *and adds an edge between the neighbours of v.*
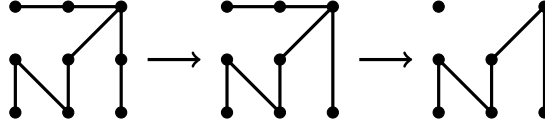

Figure 3: An example of a sequence of one bridge and one break operation.

Figure 3 gives an example of multiple applications of the bridge and break operators. We will now define the general states required to solve problem 2.

Definition 3.2. *An* extended IQP graph *is represented by* $\widetilde{\mathbf{Q}} \in \{-1, 0, 1\}^{n_a \times n_p}$. *The vertex set contains* $A = \{a_1, ..., a_{n_a}\}$ *and* $P = \{p_1, ..., p_{n_p}\}$ *while* $\widetilde{\mathbf{Q}}_{hj} = 0$ *and* $\widetilde{\mathbf{Q}}_{hj} = 1$ *denote if there is a connection or not between vertices* a_h *and* p_j *in the same way as for the IQP graph of Definition 2.3.*

We interpret $\widetilde{\mathbf{Q}}_{hj} = -1$ *as the existence of an intermediary vertex* b_k *between vertices* p_j *and* a_h, *and denote with* n_b *the number of -1s in* $\widetilde{\mathbf{Q}}$. *The vertex set includes the* bridge and break vertices $B = \{b_1, ..., b_{n_b}\}$ *and the edge set includes edges between* b_k *and* a_h *as well as between* b_k *and* p_j *when* $\widetilde{\mathbf{Q}}_{hj} = -1$. *We define a surjective function g for which* $g(h, j) = k$ *when* b_k *is the intermediate vertex connected to* a_h *and* p_j.

An extended IQP graph $\widetilde{\mathbf{Q}}$ can be built from an IQP graph $\mathbf{Q}$ by replacing any number of the entries of $\mathbf{Q}$ with -1. Throughout the remainder of this work we will use tilde notation to represent an extended IQP graph $\widetilde{\mathbf{Q}}$ build from an IQP graph $\mathbf{Q}$ in this way. Figure 4 displays an example of an extended IQP graph. By applying a bridge operator to b_1 and a break operation to b_2 in $\widetilde{\mathbf{Q}}$ of Figure 4 we arrive at $\mathbf{Q}$ of Figure 1. It is in this sense that an extended IQP graph is 'more general' that an IQP graph.

To solve our three problems we must translate these graph theoretic ideas into operations on quantum states. The following definition allows us to use graphs defined above to describe entanglement patterns.
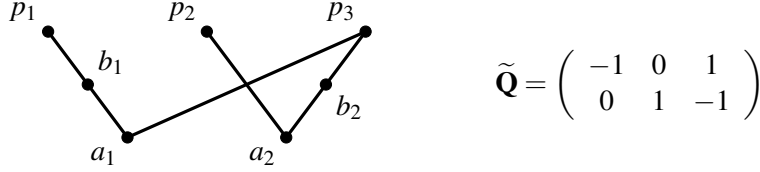
Figure 4: An extended IQP graph described by $\widetilde{\mathbf{Q}}$ with $(n_a, n_p, n_b) = (2, 3, 2)$, $P = [p_1, p_2, p_3]$, $B = [b_1, b_2]$ and $A = [a_1, a_2]$. Two vertices b_1 and b_2 are introduced and the function $g : \mathbb{Z}_{n_a \times n_p} \to \mathbb{Z}_{n_b}$ is defined as $g(1, 1) = 1$ and $g(2, 3) = 2$.

Definition 3.3. *Consider the extended IQP graph $\widetilde{\mathbf{Q}} \in \{-1, 0, 1\}^{n_a \times n_p}$ and use function $g(h, j) = k$ to define indices $k = 1, \ldots, n_b$ for the elements $\widetilde{\mathbf{Q}}_{hj} = -1$. The circuit $E_{\widetilde{\mathbf{Q}}}$ on $(n_a + n_p + n_b)$ qubits applies controlled-Z operations between qubits p_j and a_h if $\widetilde{\mathbf{Q}}_{hj} = 1$ and, between qubits $b_{g(h,j)}$ and a_h, and, $b_{g(h,j)}$ and p_j, when $\widetilde{\mathbf{Q}}_{hj} = -1$.*

Now we reformulate a lemma from [16] in order to translate bridge and break operations from graph theoretical ideas into operations on quantum states.

Lemma 3.1. *Consider a quantum state $E_{\mathbf{Q}} |\phi\rangle$ where $|\phi\rangle$ is arbitrary. If $\widetilde{\mathbf{Q}}$ is an extended IQP graph built from $\mathbf{Q}$ then there exists a state $E_{\widetilde{\mathbf{Q}}} |\psi\rangle$, which can be transformed into the state $E_{\mathbf{Q}} |\phi\rangle$ through a sequence of Pauli-Y basis measurements on qubits and local rotations around the Z axis on the unmeasured qubits through angles $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$.*

The detailed proof of Lemma 3.1 [27] shows us that we can create the following state where p_j and a_h are the primary and ancillary qubits connected to b_k (i.e. $g(h, j) = k$), r^b and d^b are strings described later in Algorithm 1 and s_k^b is the outcome of a measurement on qubit b_k.

$$\prod_{k=1}^{n_b} \left(S_{p_j}^{(-1)^{s_k^b + r_k^b}} \otimes S_{a_h}^{(-1)^{s_k^b + r_k^b}} \right)^{d_k^b} \left(Z_{p_j}^{r_k^b} \otimes Z_{a_j}^{r_k^b} \right)^{1 - d_k^b} E_{\mathbf{Q}} |\phi\rangle \tag{3}$$

To achieve this we make measurements of the qubits corresponding to bridge and break vertices (which we call *bridge and break qubits* and which are initialised as $|b_k\rangle = Y^{r_k^b} \sqrt{Y}^{d_k^b} |0\rangle$) in $E_{\widetilde{\mathbf{Q}}} |\psi\rangle$ in the Pauli Y basis. Using this method, we generate the state in Lemma 2.1 (i.e. $E_{\mathbf{Q}} \bigotimes_1^{n_a + n_p} |+\rangle$) up to S corrections, which can be accounted for by correcting the primary and ancillary measurement bases.

3.2 Blindness

To address problem 3, we wish to construct the *Ideal Resource* of Figure 5 which takes as input from the client an IQP computation $(\mathbf{Q}, \theta)$, and in return gives a classical output $\widetilde{x}$. If the Server is honest, $\widetilde{x}$ comes from the distribution corresponding to $(\mathbf{Q}, \theta)$. If the Server is dishonest, they can input some quantum operation $\mathscr{E}$ and some quantum state ρ_B and force the output to the client into the classical state $\mathscr{E}(\mathbf{Q}, \theta, \rho_B)$. We assume that the following are public knowledge: an extended IQP graph $\widetilde{\mathbf{Q}}$, the distribution $\mathscr{Q}$ over the possible $\mathbf{Q}$s from which $\widetilde{\mathbf{Q}}$ could be built, and θ. It is important to note that neither $\mathbf{Q}$, nor anything besides that which we have specified as public, is revealed to the Server.

Blindness is added to the work of Section 3.1 by performing random rotations when initialising the primary and ancillary qubits. These are corrected by rotating the measurement bases of those qubits, therefore ensuring the original IQP computation is performed. Intuitively, this randomness provides
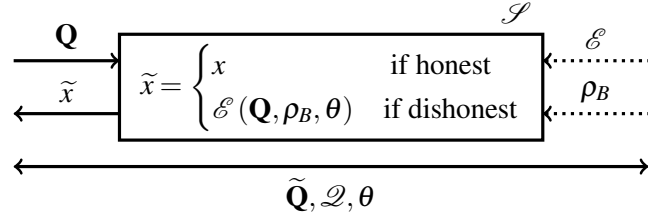
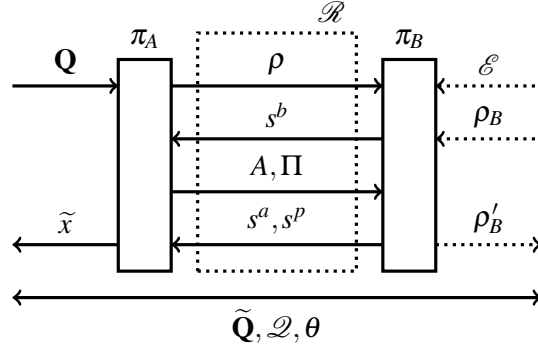Figure 5: The ideal blind delegated IQP computation resource.

Figure 6: The real communication protocol. π_A is the set of operation performed by the client, π_B are those of the Server and $\mathscr{R}$ is the communication channel (quantum and classical) used by the client and the Server in the protocol. Details of the communication protocol $\mathscr{R}$ can be seen in Algorithm 1.

blindness as it hides the previous corrections shown in equation (3), which would otherwise give away if a bridge or break operation was applied to a neighbour. Our communication protocol can be seen in Algorithm 1 and is summarised in Figure 6.

To prove composable security of the proposed protocol we drop the notion of a malicious Server for that of a global distinguisher which has a view of all inputs and outputs of the relevant resources. To be able to compare the ideal resource with the real communication protocol, we develop a simulator σ interfacing between the ideal resource $\mathscr{S}$ of Figure 5 and the distinguisher such that the latter cannot tell the difference between an interaction with the ideal resource and the real protocol. We employ the Abstract Cryptography framework introduced in [26, 32] and teleportation techniques inspired by [13] to prove security in the case of a malicious Server. If π_A is protocol run by the Client and $\mathscr{R}$ is the communication channel between them and the Server, then we can prove [27] that:

$$\pi_A \mathscr{R} \equiv \mathscr{S} \sigma \tag{10}$$

where equality here means the two protocols are indistinguishable.

Theorem 3.1. *The communication protocol described by Algorithm 1 is information theoretically secure against a dishonest Server.*

We can now be sure that our communication protocol is indistinguishable from an ideal resource which performs an IQP computation without communicating any information to the Server which is not already public. This is proven in a composable framework [13, 32] and so can be used as part of future protocols, as we will do in section 4.

Algorithm 1 Blind distributed IQP computation

Public: $\widetilde{\mathbf{Q}}, \mathscr{Q}, \theta$
client input: Q
client output: $\widetilde{x}$
Protocol:

1: The client randomly generates $r^p, d^p \in \{0,1\}^{n_p}$ and $r^a, d^a \in \{0,1\}^{n_a}$ where n_p and n_a are the numbers of primary and ancillary qubits respectively.
2: The client generates the states $\left|p_j\right\rangle = Z^{r_j^p} S^{d_j^p} \left|+\right\rangle$ and $\left|a_h\right\rangle = Z^{r_h^a} S^{d_h^a} \left|+\right\rangle$ for $j \in \{1, \ldots, n_p\}$ and $h \in \{1, \ldots, n_a\}$.
3: The client creates $d^b \in \{0,1\}^{n_b}$ in the following way: For $h = 1, \ldots, n_a$ and $j = 1, \ldots, n_p$, if $\widetilde{\mathbf{Q}}_{hj} = -1$ and $\mathbf{Q}_{hj} = 0$, then $d_k^b = 0$ else if $\widetilde{\mathbf{Q}}_{hj} = -1$ and $\mathbf{Q}_{hj} = 1$ then $d_k^b = 1$. He keeps track of the relation between k and (h, j) via the surjective function $g : \mathbb{Z}_{n_a \times n_p} \to \mathbb{Z}_{n_b}$.
4: The client generates $r^b \in \{0,1\}^{n_b}$ at random and produces the states $\left|b_k\right\rangle = Y^{r_k^b} \left(\sqrt{Y}\right)^{d_k^b} \left|0\right\rangle$ for $k \in \{1, \ldots, n_b\}$.
5: The state ρ, comprising of all of the client's produced states, is sent to the Server.
6: The Server implements $E_{\widetilde{\mathbf{Q}}}$.
7: The Server measures qubits $b_1, \ldots, b_{n_b}$ in the Y-basis $\left\{\left|+^Y\right\rangle, \left|-^Y\right\rangle\right\}$ and sends the outcome $s^b \in \{0,1\}^{n_b}$ to the client.
8: The client calculates $\Pi^z, \Pi^s \in \{0,1\}^{n_p}$ and $A^z, A^s \in \{0,1\}^{n_a}$ using equations (4), (5), (6) and (7).

$$\Pi_j^z = \sum_{h,k:g(h,j)=k} r_k^b \left(1 - d_k^b\right) - r_j^p \tag{4}$$

$$\Pi_j^s = \sum_{h,k:g(h,j)=k} (-1)^{s_k^b + r_k^b} d_k^b - d_j^p \tag{5}$$

$$A_h^z = \sum_{j,k:g(h,j)=k} r_k^b \left(1 - d_k^b\right) - r_h^a \tag{6}$$

$$A_h^s = \sum_{j,k:g(h,j)=k} (-1)^{s_k^b + r_k^b} d_k^b - d_h^a \tag{7}$$

9: The client sends, to the serve, $A \in \{0,1,2,3\}^{n_a}$ and $\Pi \in \{0,1,2,3\}^{n_p}$ for the ancillary and primary qubits respectively, where $A_h = A_h^s + 2A_h^z \pmod 4$ and $\Pi_j = \Pi_j^s + 2\Pi_j^z \pmod 4$.
10: The Server measures the respective qubits in the basis below for the ancillary and primary qubits respectively.

$$S^{-A_h} \left\{\left|0_\theta\right\rangle, \left|1_\theta\right\rangle\right\} \text{ and } S^{-\Pi_j} \left\{\left|+\right\rangle, \left|-\right\rangle\right\} \tag{8}$$

11: The measurement outcomes $s^p \in \{0,1\}^{n_p}$ and $s^a \in \{0,1\}^{n_a}$ are sent to the client.
12: The client generates and outputs $\widetilde{x} \in \{0,1\}^{n_p}$ as follows.

$$\widetilde{x}_j = s_j^p + \sum_{h:\mathbf{Q}_{hj}=1} s_h^a \pmod 2 \tag{9}$$

---

4   The Hypothesis Test

4.1   Previous work

The general idea of our *Hypothesis Test*, building on the work of [36], is that there is some hidden structure in the program elements, $\mathbf{q}_h$, of the X-program, which results in some structure in the distribution of the outputs, known only to the client. The client can use this structure to check the Server's reply. A Server possessing an IQP machine can reproduce this structure by implementing the X-program. A Server not in possession of an IQP machine, cannot generate outputs obeying the same rules. We summarise this discussion by three conditions that a hypothesis test using this method must meet.

    1.1 A client asks a Server to perform a hard-to-classically-simulate IQP computation.

    1.2 The client can check the solution to this computation because they know some secret structure that makes this checking process efficient.

    1.3 The Server must be unable to uncover this structure in polynomial time.

The particular 'known structure' of the output used in [36] to satisfy 1.2 is its *bias*.

Definition 4.1. *If* $\mathtt{X}$ *is a random variable taking values in* $\{0,1\}^{n_p}$ *and* $\mathbf{s} \in \{0,1\}^{n_p}$ *then the bias of* $\mathtt{X}$ *in the direction* $\mathbf{s}$ *is* $\mathbb{P}\left(\mathtt{X} \cdot \mathbf{s}^T = 0\right)$ *where the product is performed modulo 2. Hence, the bias of a distribution in the direction* $\mathbf{s}$ *is the probability of a sample from the distribution being orthogonal to* $\mathbf{s}$.

To calculate the bias of $\mathtt{X}$ in direction $\mathbf{s} \in \{0,1\}^n$, we form the linear code $\mathscr{C}_{\mathbf{s}}$ by selecting all rows, $\mathbf{q}_h$, of the X-program, $(\mathbf{Q}, \theta) \in \{0,1\}^{n_a \times n_p} \times [0,2\pi]$, such that $\mathbf{q}_h \cdot \mathbf{s}^T = \mathbf{1}$. We form, from them, the matrix, $\mathbf{Q}_{\mathbf{s}}$, which is set as the generator matrix of $\mathscr{C}_{\mathbf{s}}$. Defining $n_{\mathbf{s}}$ as the number of rows of $\mathbf{Q}_{\mathbf{s}}$ and #$\mathbf{c}$ as the hamming weight of $\mathbf{c}$ allows us to understand the following expression derived in [36].

$$\mathbb{P}\left(\mathtt{X} \cdot \mathbf{s}^T = 0\right) = \mathbb{E}_{\mathbf{c} \sim \mathscr{C}_{\mathbf{s}}}\left[\cos^2\left(\theta\left(n_s - 2 \cdot \#\mathbf{c}\right)\right)\right] \tag{11}$$

Hence, the bias of an X-program in the direction $\mathbf{s}$ depends only on θ and the linear code defined by the generator matrix $\mathbf{Q}_{\mathbf{s}}$. One can now imagine a hypothesis test derived from this. Although the X-program to be implemented needs to be made public, the direction $\mathbf{s}$ which will be used for checking, will be kept secret. This gives a client, with the computational power to calculate the quantity of expression (11), the necessary information to compute the bias, but does not afford the Server the same privilege.

What we want to show is that the only way for the Server to produce an output with the correct bias is to use an IQP machine. If the Server could uncover $\mathbf{s}$ then they could calculate the value of expression (11) and return vectors to the client which are orthogonal to $\mathbf{s}$ with the correct probability. We specialise the conditions mentioned at the beginning of this section to this particular method.

    2.1 The X-Program sent to a Server represents an IQP computation that is hard to classically simulate.

    2.2 It must be possible for a client, having knowledge of a secret $\mathbf{s}$ and the X-program, to calculate the quantity of expression (11).

    2.3 The knowledge of the Server must be insufficient to learn the value of $\mathbf{s}$.

In [36] the authors develop a protocol for building an X-program and a vector $\mathbf{s}$ performing this type of hypothesis test. The code $\mathscr{C}_{\mathbf{s}}$ used is a quadratic residue code with $\theta = \frac{\pi}{8}$ and condition 2.1 is conjectured to be satisfied by X-program matrices generating this code space. This conjecture is supported by giving a classical simulation that is believed to be optimal and achieves maximum bias

value 0.75, lower than that expected from an IQP machine. Condition 2.2 is satisfied in [36], by explicitly calculating the bias value to be $\cos^2\left(\frac{\pi}{8}\right)$ for their choice of X-program and $\mathbf{s}$.

The way in which condition 2.3 is addressed in [36] relies on the fact that the right-hand side of equation (11) is equal for all generator matrices in a *matroid* [31].

Definition 4.2. *A h-point binary* matroid *is an equivalence class of matrices with h rows, defined over* $\mathbb{F}_2$. *Two matrices,* $\mathbf{M}_1$ *and* $\mathbf{M}_2$, *are said to be equivalent if, for some permutation matrix* $\mathbf{R}$, *the column echelon reduced form of* $\mathbf{M}_1$ *is the same as the column echelon reduced form of* $\mathbf{R} \cdot \mathbf{M}_2$ *(In the case where the column dimensions do not match, we define equivalence by deleting columns containing only* 0*s after column echelon reduction).*

Consider right-multiplying the matrix $\mathbf{A}$ on $\mathbf{Q}$ where $\mathbf{A}$ is chosen so that $\mathbf{QA}$ belongs to the same matroid as $\mathbf{Q}$. Notice that $\mathbf{q}_h \mathbf{s}'^T = (\mathbf{q}_h \mathbf{A})\left(\mathbf{A}^{-1}\mathbf{s}'^T\right)$ and so rows originally non-orthogonal to $\mathbf{s}'^T$ are now non-orthogonal to $\mathbf{A}^{-1}\mathbf{s}'^T$. Hence, we can locate the matroid $\mathbf{Q}_{s'}$ in the matrix $\mathbf{QA}$ by using $\mathbf{A}^{-1}\mathbf{s}'^T$. We now understand what to do to the $\mathbf{Q}$ describing the X-program we are considering, so that we remain in the same matroid, and hence prevent the bias from changing, if we perform operations on $\mathbf{s}'$.

To hide the full description of $\mathbf{Q}$, use instead $\mathbf{Q}'$, built by adding additional random rows orthogonal to $\mathbf{s}'$, which do not affect the value of the bias. The combination of matrix randomisation, and the the fact that they do not know $\mathbf{Q}'$, makes it hard, as conjectured in [36], up to some computational complexity assumptions, for the Server to recover $\mathbf{A}^{-1}\mathbf{s}'^T$ if they are given $\mathbf{Q}'\mathbf{A}$, even when they know $\mathbf{s}'$ and $\mathbf{Q}$. It is now a matter for the Server to implement the X-program $\mathbf{Q}'\mathbf{A}$ and for the client to check the bias of the output in the hidden direction $\mathbf{s}^T = \mathbf{A}^{-1}\mathbf{s}'^T$. This is the idea behind the approach used by [36] to address condition 2.3.

4.2   Our Protocol

The main contribution of this work is to revisit condition 2.3.

Theorem 4.1. *Algorithm 2 presents an information-theoretically secure solution to condition 2.3.*

In Algorithm 2 we provide a hypothesis test that uses Algorithm 1 to verify quantum superiority. By using the blind IQP computation resource of Section 3.2 we have solved condition 2.3 but do so now with information theoretic security as opposed to the reliance on computational complexity assumptions used by [36]. This is true because the Server learns only the distribution $\mathscr{Q}$ over the possible set of graphs $\mathbf{Q}$. By setting $\mathbf{Q} = \mathbf{Q_s A}$, Algorithm 2 develops a bijection mapping $\widehat{\mathbf{s}} \in \{0,1\}^{n_p-1}$ to a unique matrix $\mathbf{Q} \in \{0,1\}^{n_a \times n_p}$. So $\mathscr{Q}$ is equivalent to the distribution from which $\widehat{\mathbf{s}}$ is drawn. In this case it is the uniform distribution over a set of size 2^{n_p-1}. In particular the server does not learn $\mathbf{s}$ as is required by condition 2.3. Notice that Algorithm 2 should be repeated to accurately check the bias is $\cos^2\left(\frac{\pi}{8}\right)$ as we would expect from the quantum case.

5   Discussion, Conclusion and Future Work

We have presented a new certification technique for IQP machines which can be run by a client able to prepare single-qubit Pauli eigenstates. By giving the client minimal quantum abilities we can remove computational restrictions placed on the Server in previous work [36] and, instead, prove information-theoretical security against an untrusted Server.

There are several advantages of using this tailored verification protocol for IQP computations, rather than a straightforward verification in a universal quantum computing model [16, 24]. The latter requires

---

Algorithm 2 Our hypothesis test protocol

Input: n_a prime such that $n_a + 1$ is a multiple of 8.

client output: $o \in \{0, 1\}$

Protocol:

1: Set $n_p = \frac{n_a + 1}{2}$
2: Take the quadratic residue code generator matrix $\mathbf{Q_r} \in \{0, 1\}^{n_a \times (n_p - 1)}$
3: Let $\mathbf{Q_s} \in \{0, 1\}^{n_a \times n_p}$ be $\mathbf{Q_r}$ with a column of ones appended to the last column.
4: Pick $\widehat{\mathbf{s}} \in \{0, 1\}^{n_p - 1}$ chosen uniformly at random.
5: Define the matrix $\mathbf{A} \in \{0, 1\}^{n_p \times n_p}$ according to equation (12).

$$\mathbf{A}_{h,j} = \begin{cases} 1 & \text{if } h = j \\ 0 & \text{if } h \neq j \text{ and } j < n_p \\ \widehat{\mathbf{s}}_h & \text{if } j = n_p \text{ and } h < n_p \end{cases} \tag{12}$$

6: Set $\mathbf{Q} = \mathbf{Q_s A}$ and $\theta = \frac{\pi}{8}$.
7: Set $\widetilde{\mathbf{Q}}$ to be the matrix $\mathbf{Q_r}$ with a column of -1 appended.
8: Set $\mathscr{Q}$ to be the uniform distribution over all possible $\mathbf{Q}$ for different $\widehat{\mathbf{s}}$.
9: Perform IQP computation $\mathbf{Q}$ using Algorithm 1 with inputs $\mathbf{Q}$, $\widetilde{\mathbf{Q}}$, $\mathscr{Q}$ and θ and outputs $\widetilde{x}$ and ρ'_B.
10: Let $\mathbf{s}' \in \{0, 1\}^{n_p}$ be the vector with entries all equal to zero with the exception of the last which is set to one.
11: Test the orthogonality of the output $\widetilde{x}$ against $\mathbf{s}^T = A^{-1} \mathbf{s}'^T$ setting $o = 0$ if it is not orthogonal and $o = 1$ if it is orthogonal.

---

higher precision in the manipulation of single qubits from the client (use of states other than single-qubit Pauli eigenstates) and significantly more quantum communication and processing for the verification technique. Although the qubit consumption is still, as in this work, linear in the size of the computation, in early machines the constant factor will likely be important. Further, asking a Server to perform an IQP computation using a model that is universal for quantum computation would require the Server to create large cluster states and perform measurement that might lie far beyond its IQP capabilities.

IQP circuits are important as they may prove easier to implement experimentally compared to universal quantum computers. Due to the commutativity of the gates it is theoretically possible to perform an IQP computation in one round of measurements. Our protocol requires a two-round MBQC computation which we believe is not a significant additional requirement, given the implementation requirements of universal quantum computation. We therefore believe that our scheme will be implementable in the near future, for a small number of qubits, and so a future avenue of research would be to study this hypothesis test protocol under realistic experimental errors, following similar examples of work in this direction [9, 23].

The demand for the Server to have memory to support a two-round MBQC computation means machines capable of passing the original test of [36] might not be able to pass that of this work. These studies do not restrict the architectures that the Server can use, which comes at the high cost of placing computational restrictions on the Server [36]. Our requirement that the Server can perform two-round MBQC allows us to achieve information-theoretical security. Finally, given that Gaussian quantum subtheory can be seen as a continuous variable analogue of the stabiliser formalism [3, 4, 38], a natural extension would be a continuous variable analogue of our protocol where the client prepares only Gaussian states.

6   Acknowledgements

References

[1]   Scott Aaronson & Alex Arkhipov (2011): *The computational complexity of linear optics*. In: *Proceedings of the forty-third annual ACM symposium on Theory of computing*, ACM, pp. 333–342, doi:10.1145/1993636.1993682.

[2]   Panos Aliferis, Frederico Brito, David P DiVincenzo, John Preskill, Matthias Steffen & Barbara M Terhal (2009): *Fault-tolerant computing with biased-noise superconducting qubits: a case study*. New Journal of Physics 11(1), p. 013061, doi:10.1088/1367-2630/11/1/013061.

[3]   Stephen D Bartlett, Terry Rudolph & Robert W Spekkens (2012): *Reconstruction of Gaussian quantum mechanics from Liouville mechanics with an epistemic restriction*. Physical Review A 86(1), p. 012103, doi:10.1103/PhysRevA.86.012103.

[4]   Stephen D Bartlett, Barry C Sanders, Samuel L Braunstein & Kae Nemoto (2002): *Efficient classical simulation of continuous variable quantum information processes*. Physical Review Letters 88(9), p. 097904, doi:10.1103/PhysRevLett.88.097904.

[5]   Charles H Bennett & Gilles Brassard (1984): *Quantum cryptography: Public key distribution and coin tossing*. In: *Proceedings of the IEEE International Conference on Computers Systems and Signal Processing, 1984*, pp. 175–179.

[6]   Juan Bermejo-Vega, Dominik Hangleiter, Martin Schwarz, Robert Raussendorf & Jens Eisert (2017): *Architectures for quantum simulation showing quantum supremacy*. arXiv preprint arXiv:1703.00466.

[7]   Michael J Bremner, Richard Jozsa & Dan J Shepherd (2010): *Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy*. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, The Royal Society, p. rspa20100301, doi:10.1098/rspa.2010.0301.

[8]   Michael J Bremner, Ashley Montanaro & Dan J Shepherd (2016): *Average-case complexity versus approximate simulation of commuting quantum computations*. Physical review letters 117(8), p. 080501, doi:10.1103/PhysRevLett.117.080501.

[9]   MJ Bremner, A Montanaro & DJ Shepherd (2017): *Achieving quantum supremacy with sparse and noisy commuting quantum computations*. Quantum, doi:10.22331/q-2017-04-25-8.

[10] Anne Broadbent, Joseph Fitzsimons & Elham Kashefi (2009): *Universal blind quantum computation*. In: *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, IEEE, pp. 517–526, doi:10.1109/FOCS.2009.36.

[11] Vincent Danos & Elham Kashefi (2006): *Determinism in the one-way model*. Physical Review A 74(5), p. 052310, doi:10.1103/PhysRevA.74.052310.

[12] Vincent Danos, Elham Kashefi & Prakash Panangaden (2007): *The measurement calculus*. Journal of the ACM (JACM) 54(2), p. 8, doi:10.1145/1219092.1219096.

[13] Vedran Dunjko, Joseph F Fitzsimons, Christopher Portmann & Renato Renner (2014): *Composable security of delegated quantum computation*. In: *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, pp. 406–425, doi:10.1007/978-3-662-45608-8_22.

[14] Artur K Ekert (1991): *Quantum cryptography based on Bells theorem*. Physical review letters 67(6), p. 661, doi:10.1103/PhysRevLett.67.661.

[15] Richard P Feynman (1982): *Simulating physics with computers*. International journal of theoretical physics 21(6), pp. 467–488, doi:10.1007/BF02650179.

[16] Joseph F. Fitzsimons & Elham Kashefi (2017): *Unconditionally verifiable blind quantum computation*. Phys. Rev. A 96, p. 012303, doi:10.1103/PhysRevA.96.012303.

[17] Xun Gao, Sheng-Tao Wang & L-M Duan (2017): *Quantum supremacy for simulating a translation-invariant Ising spin model*. Physical Review Letters 118(4), p. 040502, doi:10.1103/PhysRevLett.118.040502.

[18] Bryan T Gard, Keith R Motes, Jonathan P Olson, Peter P Rohde & Jonathan P Dowling (2015): *An introduction to boson-sampling. From atomic to mesoscale: The role of quantum coherence in systems of various complexities*. World Scientific Publishing Co. Pte. Ltd, pp. 167–92, doi:10.1142/9789814678704_0008.

[19] IM Georgescu, S Ashhab & Franco Nori (2014): *Quantum simulation*. Reviews of Modern Physics 86(1), p. 153, doi:10.1103/RevModPhys.86.153.

[20] Lov K Grover (1996): *A fast quantum mechanical algorithm for database search*. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, ACM, pp. 212–219, doi:10.1145/237814.237866.

[21] Marc Hein, Jens Eisert & Hans J Briegel (2004): *Multiparty entanglement in graph states*. Physical Review A 69(6), p. 062311, doi:10.1103/PhysRevA.69.062311.

[22] Matty J Hoban, Joel J Wallman, Hussain Anwar, Naïri Usher, Robert Raussendorf & Dan E Browne (2014): *Measurement-based classical computation*. Physical review letters 112(14), p. 140505, doi:10.1103/PhysRevLett.112.140505.

[23] Theodoros Kapourniotis & Animesh Datta (2017): *Nonadaptive fault-tolerant verification of quantum supremacy with noise*. arXiv preprint arXiv:1703.09568.

[24] Elham Kashefi & Petros Wallden (2017): *Optimised resource construction for verifiable quantum computation*. Journal of Physics A: Mathematical and Theoretical 50(14), p. 145306, doi:10.1088/1751-8121/aa5dac.

[25] Emanuel Knill & Raymond Laflamme (1998): *Power of one bit of quantum information*. Physical Review Letters 81(25), p. 5672, doi:10.1103/PhysRevLett.81.5672.

[26] Ueli Maurer & Renato Renner (2011): *Abstract cryptography*. In: *In Innovations in Computer Science*, pp. 1–21.

[27] Daniel Mills, Anna Pappa, Theodoros Kapourniotis & Elham Kashefi (2017): *Information Theoretically Secure Hypothesis Test for Temporally Unstructured Quantum Computation*. arXiv preprint arXiv:1704.01998.

[28] Tomoyuki Morimae, Keisuke Fujii & Joseph F Fitzsimons (2014): *Hardness of classically simulating the one-clean-qubit model*. Physical review letters 112(13), p. 130502, doi:10.1103/PhysRevLett.112.130502.

[29] Maarten Van den Nest, Wolfgang Dür & Hans J Briegel (2008): *Completeness of the classical 2D Ising model and universal quantum computation*. Physical review letters 100(11), p. 110501, doi:10.1103/PhysRevLett.100.110501.

[30] Alex Neville, Chris Sparrow, Raphaël Clifford, Eric Johnston, Patrick M Birchall, Ashley Montanaro & Anthony Laing (2017): *No imminent quantum supremacy by boson sampling*. arXiv preprint arXiv:1705.00686.

[31] James G Oxley (2006): *Matroid theory*. 3, Oxford University Press, USA.

[32] Christopher Portmann & Renato Renner (2014): *Cryptographic security of quantum key distribution*. arXiv preprint arXiv:1409.3525.

[33] John Preskill (2012): *Quantum computing and the entanglement frontier*. arXiv preprint arXiv:1203.5813.

[34] Robert Raussendorf & Hans J Briegel (2001): *A one-way quantum computer*. Physical Review Letters 86(22), p. 5188, doi:10.1103/PhysRevLett.86.5188.

[35] Robert Raussendorf, Daniel E Browne & Hans J Briegel (2003): *Measurement-based quantum computation on cluster states*. Physical review A 68(2), p. 022312, doi:10.1103/PhysRevA.68.022312.

[36] Dan Shepherd & Michael J Bremner (2009): *Temporally unstructured quantum computation*. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 465, The Royal Society, pp. 1413–1439, doi:10.1098/rspa.2008.0443.

[37] Peter W Shor (1999): *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*. *SIAM review* 41(2), pp. 303–332, doi:10.1137/S0036144598347011.

[38] Robert W Spekkens (2016): *Quasi-quantization: classical statistical theories with an epistemic restriction*. In: *Quantum Theory: Informational Foundations and Foils*, Springer, pp. 83–135, doi:10.1007/978-94-017-7303-4_4.