entropy

MDPI

Article

Improved Resource State for Verifiable Blind Quantum Computation

Qingshan Xu , Xiaoqing Tan * and Rui Huang

College of Information Science and Technology, Jinan University, Guangzhou 510632, China; xuqingshan1008@stu2018.jnu.edu.cn (Q.X.); hrui01@stu2019.jnu.edu.cn (R.H.)
* Correspondence: ttanxq@jnu.edu.cn

Received: 26 August 2020; Accepted: 3 September 2020; Published: 7 September 2020

Abstract: Recent advances in theoretical and experimental quantum computing raise the problem of verifying the outcome of these quantum computations. The recent verification protocols using blind quantum computing are fruitful for addressing this problem. Unfortunately, all known schemes have relatively high overhead. Here we present a novel construction for the resource state of verifiable blind quantum computation. This approach achieves a better verifiability of 0.866 in the case of classical output. In addition, the number of required qubits is $2N + 4cN$, where N and c are the number of vertices and the maximal degree in the original computation graph, respectively. In other words, our overhead is less linear in the size of the computational scale. Finally, we utilize the method of repetition and fault-tolerant code to optimise the verifiability.

Keywords: blind quantum computation; quantum verification; delegated quantum computation

1. Introduction

Scalable quantum computing still has a long way to go, while quantum computing in cloud mode is relatively reasonable. The scenario is that a client who only has access to classical computation and a limited quantum device used for preparing or measuring single qubits delegates a computation task to an untrusted server with a full-fledged quantum computer. In addition, the client's input, output, and computation remain private to the server. Such secure quantum computing protocols are called blind quantum computing (BQC) [1–16]. However, how can a client verify the outcome of the computation sent by a server when a quantum experiment solves a problem which is proven to be intractable for classical computers? Fortunately, there has been a lot of progress in the development of verification protocols [17–29]. The goal of verifiable universal blind quantum computation (VUBQC) is to detect deviation with high probability when the server behave dishonestly and reject his output. Here, the VUBQC scheme we consider is based on constructing the delegated computation to include certain traps in such a way that the computation is not affected, while revealing no information to the device [17]. Then one can verify that the computation has been performed correctly, with exponentially small probability of error.

There are two important properties in the verification protocols [28]. The first one is verifiability, which means the maximal probability for the output of the protocol to be incorrect and the client accepting. The other one is correctness, which means the minimal probability for the client obtaining the correct outcome when the server behaves honestly. Especially, we characterize the client as a verifier and the server as a prover. The term "verifiable" for VUBQC is related to notions of completeness and soundness in the context of interactive-proof system. Given a problem that is classically intractable, the verifier can accept a correct solution with high probability and reject a invalid solution with high probability at the end of the interaction with the prover. Note that even if the verifier accept the outcome sent by the prover, the outcome may be still incorrect. However, the probability that

the verifier accepting a wrong outcome can be reduced to a value approaching 0 through some improvements for VUBQC.

In reality, exploring a verification protocol with arbitrarily small verifiability while keeping the cost of resource optimal is still an opening problem. Some progress has been made in this regard. In [17], dotted-complete graph was used for resource construction in verification protocol. It can achieve verifiability $\epsilon = (5/6)^{\lceil 2d/5 \rceil}$, where d is the distance of error correcting code used in the protocol. However, the overhead of verification protocol is quadratic in the size of the computation. In [25], a optimised resource construction using dotted-triple graph was proposed, where the number of traps can be a constant fraction of the total number of qubits. It can obtain verifiability $\epsilon = (8/9)^{\lceil d/18 \rceil}$. More importantly, it only requires a linear overhead in the size of the computation.

The verification scheme we present here makes use of similar elements as suggested in [17], trap computations are used to detect errors and a fault-tolerant encoding of the computation is used to amplify the detection rate. Compared with [17], we construct a sandglass-like resource state such that the overhead is linearly related to the size of the computation. In addition, compared with [25], not only do we just need fewer qubits, but also achieve a better verifiability.

The remainder of the paper is organized as follows. In Section 2 we give some basic notions about verifiable universal blind quantum computation. Next, in Section 3 we give the process of our sandglass-like resource state construction. Then in Section 4 we propose a verifiable blind quantum computation protocol with sandglass-like resource state and analyse the correctness and verifiability of the protocol. For classical output case and quantum output case, in Section 5 we propose an improved scheme to improve verifiability. We finally conclude, in Section 6, with some discussions and open problems.

2. Preliminaries

We briefly present the relevant concepts used in describing VUBQC protocols. The first one is the model of measurement-based quantum computation (MBQC) [30–32]. Different from the traditional quantum circuit model, in MBQC a given computation is performed by measuring qubits from a large entangled state. This special entangled state consists of qubits prepared in the state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, entangled using $\mathrm{CZ} = I \otimes |0\rangle\langle 0| + Z \otimes |1\rangle\langle 1|$ operations. The entangled state is also known as graph state, which can be determined by a given graph. In other words, given an undirected graph G with n vertices $i \in V$ and edges $(i,j) \in E$, the graph state $|G\rangle$ that corresponds to G is defined by $|G\rangle = \left(\prod_{(i,j)\in E} CZ_{ij}\right)|+\rangle^{\otimes n}$, where CZ_{ij} is the CZ operation acting on vertices sharing the edge (i,j). Then they are measured in the basis $\left\{|+_\phi\rangle = \left(|0\rangle + e^{i\phi}|1\rangle\right)/\sqrt{2},\ |-_\phi\rangle = \left(|0\rangle - e^{i\phi}|1\rangle\right)/\sqrt{2}\right\}$, where the measurement angle is $\phi \in \{0, \pi/4, \cdots, 7\pi/4\}$ depending on outcomes of previous measurements.

The second part is blind quantum computing [2], which is based on the MBQC model. The protocol runs as follows: (1) Randomly rotated single-qubit states $\left\{\left|+_{\theta_j}\right\rangle = \left(|0\rangle + e^{i\theta_j}|1\rangle\right)/\sqrt{2}\right\}_{j=1}^{N}$ are prepared by Alice, where $\theta_j \in \{0, \pi/4, \cdots, 7\pi/4\}$ is a random angle, and then Alice sends them to Bob. (2) Bob creates a certain graph state called the brickwork state [2] by entangling obtained states with CZ operations. (3) Alice calculates the measurement angle depending on outcomes of previous measurements and sends it to Bob. (4) Bob performs the measurement in the angle sent by Alice, and returns the measurement result to Alice. (5) Alice and Bob repeat (3) and (4) until all qubits of the brickwork state are measured. If Bob behaves honestly, Alice obtains the correct outcome of desired quantum computation. Furthermore, whatever malicious Bob does, Bob learns nothing about computation's input, output, and algorithm.

The last one is the VUBQC protocol [17], which augments BQC with the ability to detect malicious behaviour of server (Bob). Because no entanglement is created when CZ operation acts on state $|0\rangle$ or $|1\rangle$. One can randomly choose a $|+_\theta\rangle$ qubit (trap qubit) whose neighbours are computational basis states (dummy qubits) such that this qubit is disentangled from the rest qubits in the graph state. Then measuring this trap qubit in θ angle will obtain the deterministic outcome. In [17], the cylinder

brickwork state was used such that there are only disentangled trap qubits in a product state with a brickwork state left after entanglement operations are applied by the Bob. Due to the the positions of the traps and dummies are unknown to Bob, the blindness of the protocol is guaranteed. The verifier (Alice) uses trap qubits as traps to test that the prover (Bob) performs desired quantum operations.

3. Sandglass-Like Resource State Construction

We now proceed to construct sandglass-like resource state in a manner similar to the construction of the dotted-triple graph state of [25]. As mentioned in MBQC, given a graph G we can obtain a corresponding graph state $|G\rangle$, which is used to perform a universal quantum computation. We call G a base graph. Then we use the base graph G to construct a sandglass-like graph $S(G)$ whose corresponding graph state $|S(G)\rangle$ achieves verifiable quantum computation. Furthermore, some operations will be performed on a coloured version of the sandglass-like graph $S(G)$ in order to obtain a subgraph used for computation and a subgraph used for traps. Because the selection of computation subgraph and trap subgraph is unknown to the prover, security of the scheme is protected. The same as [25], our construction of subgraphs is local. However, our method requires less qubits and obtains a better verifiability.

We then give specific definitions and related properties of the sandglass-like resource state. Since the construction of resource state depends completely on the sandglass-like graph (each vertex represents a qubit and each edge represents a CZ entanglement operation), we only need to consider the construction of the sandglass-like graph.

According to reference [17], the dotting operator on graph G is defined to be the operator that transforms a graph G to a new graph denoted by $D(G)$ by replacing every edge in G with a new vertex connected to the two vertices originally joined by that edge. Given an arbitrary base graph G, the construction procedure of the sandglass-like graph $S(G)$ is described as follows.

(1) A base graph G consists of vertices $v \in V(G)$ and edges $e \in E(G)$.
(2) For each vertex $v_i \in V(G)$, we define a set of two new vertices $P_{v_i} = \{p_1^{v_i}, p_2^{v_i}\}$, where $p_k^{v_i}$ represents the kth vertex of the set P_{v_i}.
(3) For each edge $e_{ij} \in E(G)$, which connects the vertices v_i and v_j, we define a set of four edges E_{ij} that connect vertices in the set P_{v_i} with the vertices in the set P_{v_j}. More concretely, the set E_{ij} consists of four edges e_{ij}^{11}, e_{ij}^{12}, e_{ij}^{21}, and e_{ij}^{22}, where e_{ij}^{mn} represents an edge connecting the vertex $p_m^{v_i}$ and the vertex $p_n^{v_j}$.
(4) We define an intermediate graph $I(G)$ to be a graph consisting of vertices $\bigcup_{v_i \in V(G)} P_{v_i}$ and edges $\bigcup_{e_{ij} \in E(G)} E_{ij}$ described in steps (2) and (3). We perform the dotting operator on the intermediate graph $I(G)$ resulting in a sandglass-like graph denoted by $S(G)$.

Note that the sandglass-like graph $S(G)$ is actually equal to $D(I(G))$. An example of the construction of sandglass-like graph $S(G)$ is illustrated in Figure 1. The base graph G considered in this example consists of four vertices and three edges, as shown in the Figure 1a. The corresponding intermediate graph $I(G)$ is shown in the Figure 1b. Furthermore, Figure 1c gives the sandglass-like graph $S(G)$ corresponding to the base graph G. According to the construction method, total number of vertices in the sandglass-like graph is $|V(S(G))| = 2|V(G)| + 4|E(G)|$. We therefore only need $(2N + 4cN)$ qubits for our verifiable quantum computation, where N is the number of qubits for universal quantum computation and c is the maximum degree of the base graph. Our construction can apply to other graph states. Since the basic unit of any graph state is two qubits entangled by a CZ gate, the construction procedure of our sandglass-like graph is exactly aimed at a transformation to each basic unit.

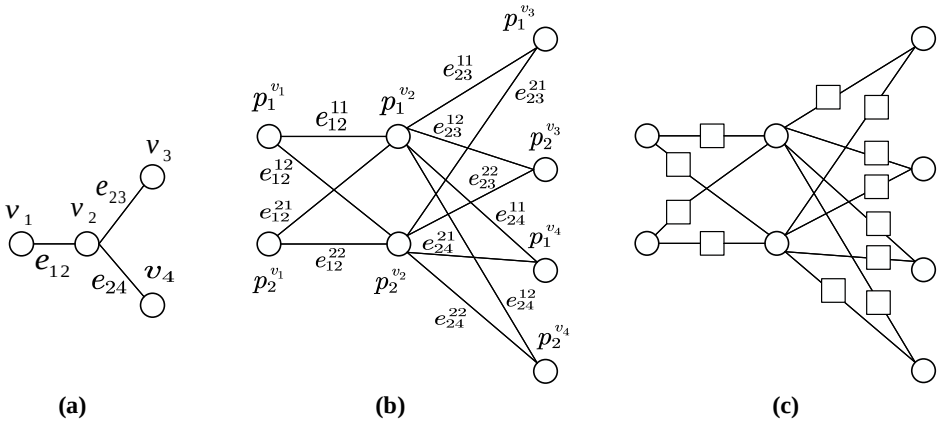

Figure 1. (**a**) A base graph G consisting of four vertices and three edges. (**b**) The intermediate graph I(G) corresponding to the base graph G. (**c**) The sandglass-like graph $S(G)$ corresponding to the base graph G. The circle represents a primary vertex and the square represents an added vertex.

Once we have the sandglass-like graph $S(G)$ we can color it for subsequent break operations and bridge operations. We call the set of vertices P_{v_i} a primary set. In addition, we say that the vertices in each primary set are primary vertices. Similarly, we denote the set of four vertices related to each edge e_{ij} as an added set $A_{e_{v_i,v_j}}$, and say that the vertices in each added set are added vertices. Similar to the trap-coloring in [25], our definition about trap-coloring of the sandglass-like graph $S(G)$ satisfies the following conditions.

(1) Primary vertices are coloured in one of the three colours of white, red or green.
(2) Added vertices are coloured in one of the three colours of white, red or green.
(3) One of vertices in each primary set P_{v_i} is uniformly at random chosen to be colored in green. The remaining one vertex of P_{v_i} has probability α to be colored in red and probability $1-\alpha$ to be colored in white, where α is an appropriate constant and $0 < \alpha < 1$.
(4) The colours of the primary vertices determine the colours of the added vertices. These added vertices connecting primary vertices of different colours are white. These added vertices connecting both green primary vertices are green. Moreover, these added vertices connecting both white primary vertices are red.

Since the color of the added vertices depends on the color of the primary vertices, one may have no red vertex in each primary set P_{v_i} or added set $A_{e_{v_i,v_j}}$. A specific example of trap-coloring is given in Figure 2a.

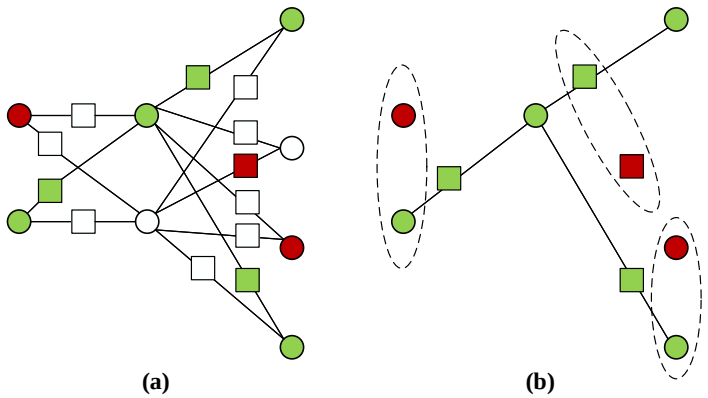

Figure 2. (**a**) The trap-colouring of the sandglass-like graph $S(G)$. (**b**) A computation subgraph and a trap subgraph obtained by performing break operations on the white vertices of the coloured $S(G)$. For each green computation vertex, there may be a corresponding red trap vertex.

While the construction and the coloring principle of the sandglass-like graph is public, the specific coloring scheme is completely decided by Alice (the client) so that Bob (the server) can not know which vertex is green or red or white. Every vertex has the possibility to be coloured in red (trap qubit). In addition, the coloring of every primary set is independent from the coloring of other primary sets, and the coloring of every added set depends on the coloring of two adjacent primary sets. These features make the security proof of [25] still applicable for our analysis.

Our inspiration comes from that we keep the computation qubits (green vertices) hidden to an untrusted client while increasing the probability that the qubit (vertex) that any attack acts on is a trap qubit (red vertex) such that any attack has a higher probability to be detected. Specifically, compared with [25] whose such a detection probability is $1/3$ for each primary set and $1/9$ for each added set, our detection probability is $\alpha/4$ for each primary set and $(1-\alpha)^2/4$ for each added set. The lower detection probability obtained when α is $2-\sqrt{3}$ crucially leads to our better verifiability for the case of classical output (see Theorem 2 in Section 4). Note that we actually trade certain symmetry (an arbitrary qubit is uniformly and randomly coloured in any one of the three colours of white, red or green) to obtain less resource overhead. However, as we will see later, this asymmetry just cause slightly inferior verifiability for the case of quantum output.

In what follows we show how to get computation subgraph and trap subgraph from the colored sandglass-like graph. To do this, we need to introduce the break and bridge operations in [17].

The bridge operator on a vertex v of degree 2 on graph G is defined to be the operator which connects the two neighbors of v and then removes vertex v and both adjacent edges from G. The break operator on a vertex v of graph G is defined to be the operator that removes vertex v and all adjacent edges from G.

As shown in Figure 3, the break and bridge operators are demonstrated, respectively. In Figure 3a the break operator acting on vertex v_2 removes vertex v_2 and edges e_{12},e_{23} resulting in isolated two vertices v_1,v_3. In Figure 3b the bridge operator acting on vertex v_2 connects vertices v_1,v_3 with a new edge e_{13} and removes vertex v_2 and edges e_{12},e_{23} resulting in two direct connected vertices.

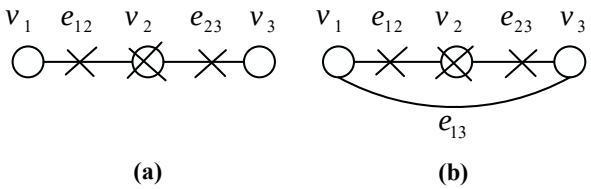

Figure 3. (**a**) A break operator on the vertex v_2. (**b**) A bridge operator on the vertex v_2.

Note that both break and bridge operations on a graph have corresponding implementations of quantum form [17]. To clarify this, if we measure any qubit in a graph state in Pauli Z basis, we will get a state obtained from the graph, in which the measured vertex and its adjacent edges are removed, up to local Pauli Z corrections. It is equivalent to the break operation. However, what we use more frequently is another equivalent method. In other words, we set the qubit that the break operator acts on to be a dummy qubit, where the dummy qubit is in the state $|0\rangle$ or $|1\rangle$. Depending on the specific value of the dummy qubit, a Pauli Z rotation on all the neighboring qubits in the graph will be introduced after the entanglement operation is performed. As for the bridge operation, if we measure any qubit in Pauli Y basis, we will obtain the graph state corresponding to the graph, in which the measured vertex and its adjacent edges are removed and a new edge connecting the adjacent vertices is created, up to local Z rotations by $\pi/2$ or $-\pi/2$.

Now we move on the generation process of the computation subgraph and trap subgraph. Given a colored sandglass-like graph $S(G)$, we perform break operations on the white vertices and bridge operations on the green added vertices (green square vertices) such that we can obtain a computation subgraph and a trap subgraph, as illustrated by Figure 2b. Further more, the red vertices and green vertices are actually trap qubits and computation qubits, respectively. Note that

in Figure 2b we preserve the green square vertices for matching computation qubits with trap qubits (dashed circle).

It is noteworthy that our sandglass-like graph draws the same conclusion as Theorem 1 in [25], which will be used in Section 5. To interpret this, we introduce relevant concepts in [25]. We define the base-location of a vertex f of the sandglass-like graph $S(G)$ to be the set P_v or A_e that contains f in $S(G)$. Given a sandglass-like graph $S(G)$ and a collection of n base-locations $\mathcal{E}$, we call the set $\mathcal{E}$ independently colourable locations (ICL) if the choice of colours within any set corresponding to a base-location in $\mathcal{E}$ is independent from the choice of colours in sets corresponding to other base-locations in $\mathcal{E}$.

Lemma 1. *Given a set S consisting of n base-locations in the sandglass-like graph $S(G)$ and assume that the base graph G has maximum degree c. Then there exist a subset $S' \subseteq S$ such that S' is independently colourable locations and contains at least $|S'| = \frac{n}{2c+1}$ base-locations.*

Proof. From the graph S with n locations, an ICL subset S' can be found as follows. From our construction of the sandglass-like graph, a local-colouring of an added base location corresponds to a local-colouring of both adjacent primary base-locations. Then the necessary and sufficient condition of ICL is obtained. In other words, a set of n base-locations $\mathcal{E}$ is ICL if and only if for all pairs $i, j \in \mathcal{E}$ the sets $\epsilon_i \cap \epsilon_j = \varnothing$ (Lemma 3 of [25]), where $\epsilon_i = \{i\}$ if the base-location i is primary and $\epsilon_i = N_{S(G)}(i)$ if the base-location i is added.

One the one hand, if S' contains a primary base-location v_i, then all its adjacent added base-locations (the maximal number is c) a_{ij},a_{ik} will be excluded, as shown in Figure 4a. On the other hand, if S' contains an added base-location a_{ij}, then all its adjacent primary base-locations v_i,v_j and the adjacent added base-locations a_{ik},a_{im},a_{jn} of the primary base-locations v_i,v_j will be excluded, as shown in Figure 4b. The number of excluded base-locations is at most $2c$.

As a result, in the worst case there exists an ICL subset S' with at least $\frac{n}{2c+1}$ base-locations. □

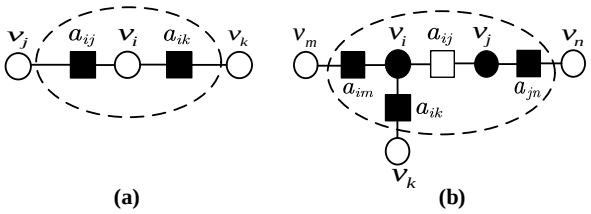

Figure 4. (**a**) The introduction of a primary base-location v_i. (**b**) The introduction of an added base-location a_{ij}. Black vertices are excluded for satisfying independently colourable locations (ICL). The dash line circles all the influenced vertices.

4. Verifiable Blind Quantum Computation with the Sandglass-Like Resource State

In this section, similar to [17,25], we present our verifiable blind quantum computation protocol. However, we use our sandglass-like graph state as the resource state of verifiable blind quantum computation. In addition, compared with [17,25], the verifiability and overhead of our protocol are optimised.

The essential idea of verification protocol is that the trap-colouring chosen by Alice (verifier) is unknown to Bob (prover) so that malicious Bob is difficult to deviate the computation while keeping the trap qubit untouched.

Recall the main procedures for VUBQC [17]. Alice converts a computation task to a graph G', where corresponding graph state $|G'\rangle$ consists of computation qubits, dummy qubits and trap qubits. In addition, each qubit of $|G'\rangle$ has a measurement angle ϕ_i called computation angle, where $\phi_i \in A = \{0, \pi/4, \cdots, 7\pi/4\}$ for all computation qubits and dummy qubits and $\phi_i = 0$ for all trap qubits. Alice then prepares states $|+_{\theta_i}\rangle$ for all computation qubits and trap qubits and computational

basis states $|0\rangle$ and $|1\rangle$ for all dummy qubits. Alice sends all these qubits to Bob, who then entangles them to obtain the graph state $|G'\rangle$. Alice sends the practical measurement angle $\delta_i = \phi'_i + \theta_i + r_i$ in the measurement order to Bob, where ϕ'_i is the updated computation angle depending on ϕ_i and outcomes of Bob's previous measurements $\boldsymbol{s}$, $\theta_i \in A$ is used to encrypt measurement angle ϕ_i and $r_i \in \{0,1\}$ is used to encrypt the outcome of measurement. Especially, $\delta_i = \theta_i + r_i$ for all trap qubits and $\delta_i \in A$ for all dummy qubits. When Bob's each trap measurement outcome b_t is equal to expected value r_t, the outcomes of measurements are accepted and corrected by Alice to obtain real results of the computation task.

Here, in our scheme the graph G' is replaced by sandglass-like graph $S(G)$, ϕ_i is equal to $\pi/2$ for all the added green vertices required to be measured in Pauli Y basis for performing the bridge operators and the function of calculation δ_i is set as $C(i, \phi_i, \theta_i, r_i, x_i, \boldsymbol{s})$. We therefore give our verification protocol as shown in Protocol 1.

Protocol 1 Verifiable blind quantum computation with sandglass-like resource state.

Alice's resources:
(1) A graph G with N vertices for performing the desired computation task in MBQC mode. The coloured sandglass-like graph $S(G)$ with at most $2N + 4cN$ vertices, where c is the maximal degree of the base graph G and labeling of vertices is known to Alice and Bob.
(2) The positions of dummy qubits for the break operations, set D, chosen to be positions of all white vertices. The positions of trap qubits chosen to be all red vertices. The positions of computation qubits chosen to be all green vertices, where green square vertices are used to perform bridge operations.
(3) An l-qubit input state $|I\rangle$.
(4) A sequence of measurement angles ϕ=$(\phi_i)_{1\leq i\leq(4N+6cN)}$ with $\phi_i \in A = \{0, \pi/4, \cdots, 7\pi/4\}$. $2N + 4cN$ random variables θ_i with values taken uniformly at random from A. l random variables x_i, $2N + 4cN$ random variables r_i, and $|D|$ random variables d_i with values taken uniformly at random from $\{0,1\}$. A binary string $\boldsymbol{s}$ of length at most $2N + 4cN$ for recording true measurement outcomes related to Bob's measurement outcomes, where $\boldsymbol{s}$ is initially set to be vector $\mathbf{0}$.
(5) A fixed function $C(i, \phi_i, \theta_i, r_i, x_i, \boldsymbol{s})$ that for each non-output qubit i computes the angle of the measurement of qubit i to be sent to the Bob.
Initial step:
(1) Alice's move: Alice sets all the values in $\boldsymbol{s}$ to be 0 and encodes the l-qubit input state as $|e\rangle = X^{x_1} Z(\theta_1) \otimes \cdots \otimes X^{x_l} Z(\theta_l) |I\rangle$. She then prepares the remaining qubits in the following form: If $i \in D$, then qubit i is set to be $|d_i\rangle$; otherwise qubit i is set to be $\prod_{j\in N_{S(G)}(i)\cap D} Z^{d_j} |+_{\theta_i}\rangle = \left|+_{\theta_i+\sum_{j\in N_{S(G)}(i)\cap D} d_j\pi}\right\rangle$, where $N_{S(G)}(i)$ represents the neighborhood of vertex i in $S(G)$. Then Alice sends Bob all qubits in the order of the labeling of the vertices of the graph $S(G)$.
(2) Bob's move: Bob receives $2N + 4cN$ single qubits and entangles them according to $S(G)$.
Step i**:** $1 \leq i \leq (2N + 4cN)$
(1) Alice's move: Alice computes the angle δ_i equal to $C(i, \phi_i, \theta_i, r_i, x_i, \boldsymbol{s})$ and sends it to Bob. If qubit i is the trap qubit, then the angle δ_i is set to be $\theta_i + r_i\pi$.
(2) Bob's move: Bob measures qubit i with angle δ_i and sends Alice result b_i.
(3) Alice's move: Alice sets the value of s_i in $\boldsymbol{s}$ to be $b_i \oplus r_i$.
Verification:
(1) After obtaining all the output qubits from Bob, if the trap qubit t is an output qubit, Alice measures it with angle $\delta_t = \theta_t + r_t\pi$ to obtain b_t.
(2) Alice accepts if $b_i = r_i$ for all the trap qubits i.
(3) Alice applies corrections according to measurement outcomes b_i and secret parameters θ_i, r_i at the output layer green qubits in order to obtain the final output.

Theorem 1 (Correctness)**.** *If Alice and Bob follow the steps of Protocol 1 honestly, then Alice accepts the correct outcome.*

Proof. Proof follows along similar lines of Theorem 2 in [25]. In Protocol 1 the dummy qubits are placed at white vertices of the coloured sandglass-like graph $S(G)$. Note that the effect of dummy qubits is the break operation on the graph $S(G)$. As a result, a green computation subgraph and

a red trap subgraph are obtained. Since both subgraphs have no effect to each other, we consider the measurements on the computation subgraph and the trap subgraph separately.

The correctness on the computation subgraph stems from the correctness of universal blind quantum computation [2]. To clarify, if each qubit in computation subgraph is rotated qubit $|+_{\theta_i}\rangle$ and measured in angle $\delta_i = C(i, \phi_i, \theta_i, r_i, x_i, \mathbf{s})$, then all the deviations from the actual implementation of the measurement pattern are corrected. Therefore Alice will get desired computation output.

As for the trap subgraph, trap qubits are isolated. Every trap qubit $|+_{\theta_i}\rangle$ will obtain deterministic measurement outcome $b_i = r_i$ after it is measured in angle δ_i equal to $\theta_i + r_i\pi$. Alice will accepts the output, as honest Bob will always return $b_i = r_i$ for all trap qubits. □

Theorem 2 (Verifiability)**.** *Protocol 1 is 0.905 verifiable in the case of quantum output and 0.866 verifiable in the case of classical output.*

The proof of above Theorem 2 can be found in Appendix A. According to the process of proof, the verifiability of Protocol 1 is equivalent to solving the following optimization problems.

$$\min_{\alpha\in(0,1)} \max\left\{1-\tfrac{1}{4}\alpha, 1-\tfrac{1}{2}\alpha, -\tfrac{1}{4}\alpha^2+\tfrac{1}{2}\alpha+\tfrac{3}{4}\right\}, \quad (1)$$

$$\min_{\alpha\in(0,1)} \max\left\{1-\tfrac{1}{2}\alpha, -\tfrac{1}{4}\alpha^2+\tfrac{1}{2}\alpha+\tfrac{3}{4}\right\}, \quad (2)$$

where Equations (1) and (2) respectively correspond to the case of quantum output and the case of classical output. Theorem 2 shows that the probability of accepting an incorrect outcome is constant.

5. Optimization of Verifiability

While we achieve the verification of blind quantum computing with the sandglass-like resource state, our protocol's verifiability is too high to be applied in practice. In this section, similar to [25], we will utilize one method respectively to reduce verifiability ϵ to arbitrarily small number in both cases of classical output and quantum output.

In the case of classical output we repeat Protocol 1 a certain number of times. Since all repetitions obtain the same correct output when Bob is honest, the verifiability can be decreased by adding an additional verification condition that Alice accepts final output if all of repetitions get the same output. From this result we can construct a new verification protocol based on repetitions, as given in Protocol 2. In the case of quantum output we use the technology of fault-tolerant code [33,34], which is often used in topological fault-tolerant blind quantum computation [4]. The main idea is that malicious Bob needs to make more attacks on computation qubits because of the existence of fault-tolerant code , which will increase the possibility of being caught by Alice. We therefore have Protocol 3.

Protocol 2 Optimised VUBQC with sandglass-like resource state for classical output.

Alice's resources:
(1) The number of repetitions $R = \frac{\log\epsilon}{\log 0.866}$, where ϵ is the desired security level.
(2) The rest of the resources are the same as Protocol 1.
Step i**:** $1 \le i \le R$
(1) Follow the steps of Protocol 1, where each repetition of Protocol 1 corresponds to identical computation task.
(2) If Alice accepts the output, she records the classical output as O_i.
Verification:
(1) If any single repetition of Protocol 1 is rejected, the overall computation will be rejected. Otherwise, Alice compares all O_i. If all O_i are identical, Alice accepts this output as the output of computation.

Theorem 3. *Protocol 2 is* 0.866^R *verifiable for classical output, where R is the number of repetitions.*

Proof. Recall that the verifiability represents maximal probability that Alice accepts an incorrect outcome. Because the condition that Alice accepts final output is that all repetitions of Protocol 1 are accepted and all of them return the same output. The event that Alice accepts an incorrect output means that all repetitions of Protocol 1 are accepted and return the same incorrect output. Since the verifiability of Protocol 1 is 0.866, the verifiability of Protocol 2 is 0.866^R. □

Protocol 3 Optimised VUBQC with sandglass-like resource state for quantum output.

Alice's resources:
(1) A base graph G encoded in a fault-tolerant way for correcting errors less than δ.
(2) The rest of the resources are the same as Protocol 1.
Same steps as in Protocol 1.

Theorem 4. *Protocol 3 is* $0.905^{\left\lceil \frac{\delta}{2(2c+1)} \right\rceil}$ *verifiable for quantum output, where* δ *is the number of tolerated errors on the base graph* G *and* c *is the maximal degree of* G.

The proof of the above Theorem 4 can be found in Appendix B. From Theorem 3 and Theorem 4 we can see that the verifiability of verification protocol is exponentially small.

6. Conclusions

Inspired by the dotted triple-graph by Kashefi and Wallden [25], we have introduced the concept of sandglass-like graph whose corresponding graph state can be used to be the resource state of verifiable blind quantum computing. We then proposed a verifiable blind quantum computation protocol with sandglass-like resource state. Based on this protocol, we proposed one new scheme in the case of classical and the case of quantum output to improve the verifiability of the original protocol.

Our main contribution can be described as follows. We have broken the symmetry of the trap-coloring in [25]. In other words, the possibility to be colored in green, the possibility to be colored in white, and the possibility to be colored in red are set to be not equal for each primary vertex. This essential point allows us to design a better resource state, which only requires a less linear overhead in the size of the computation. In addition, we achieves a better verifiability for the case of classical output, i.e., a lower probability that the client accepts a wrong outcome from the server, by optimizing the setting of the probability in the trap-coloring.

In [17], Joseph F. Fitzsimons et al proposed a VUBQC protocol using a dotted-complete graph state. Their verifiability is $(5/6)^{\lceil 2d/5 \rceil}$, where d is the defect thickness under RHG fault-tolerance scheme [35–37]. Here, RHG fault-tolerance scheme is a fault-tolerant version of the one-way quantum computer using a cluster state in three spatial dimensions, which was proposed by Raussendorf, Harrington and Goyal [36]. However, the overhead of their protocol is quadratic. In other words, the number of qubits required for the protocol is $O(N^2)$, where N is the number of qubits used to implement desired computation. The protocol of Elham Kashefi et al [25] considered a dotted triple-graph state. Their verifiability is $(8/9)^R$ in the case of classical output and $\left(\frac{8}{9}\right)^{\left\lceil \frac{\delta}{2(2c+1)} \right\rceil}$ in the case of quantum output, where R denotes the number of repetitions, δ is the number of errors that can be detected or corrected, and c is the maximal degree of the base graph G implementing desired computation. In addition, their overall cost is $3N + 9cN$. In contrast to these schemes, our protocols' verifiability is 0.866^R in the case of classical output and $0.905^{\left\lceil \frac{\delta}{2(2c+1)} \right\rceil}$ in the case of quantum output. It means that our verifiability is better in the former case and slightly worse in the later case. More importantly, our overhead is $2N + 4cN$.

For future studies, our construction can be applied to device-independent VUBQC [20,21,24] and other specific fault-tolerance codes. It is still an open problem to further reduce overhead of VUBQC.

Author Contributions: Conceptualization, Q.X.; methodology, Q.X.; formal analysis, Q.X.; investigation, Q.X., X.T. and R.H.; writing—original draft preparation, Q.X.; writing—review and editing, Q.X., X.T. and R.H.; supervision, X.T. and R.H.; project administration, X.T.; funding acquisition, X.T. All authors have read and agreed to the published version of the manuscript.

Funding: The research is partly supported by National Natural Science Foundation of China (Grant No. 61672014), National Cryptography Development Fund of China (Grant No. MMJJ20180109), Natural Science Foundation of Guangdong Province of China (Grant No. 2019A1515011069).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Theorem 2

Our proof makes full use of results from the proof of Theorem 3 of [25]. We represent the probability for Alice accepting an incorrect outcome in Protocol 1 by p_{error}. According to (C.12) of [25], we have the following bound for p_{error}.

$$p_{error} \leq \max_{i \in E_i} \sum_{T} p\left(T\right) \prod_{t \in T} \left(\sum_{\theta_t, r_t} p\left(\theta_t\right) p\left(r_t\right) \left(\left\langle \eta_t^{v_T} \right| \sigma_{i|t} \left| \eta_t^{v_T} \right\rangle \right)^2 \right), \tag{A1}$$

where σ_i is a tensor product of Pauli operators related to Bob's deviation and $\sigma_{i|t} \in \{I, X, Y, Z\}$ represents the action of σ_i on the qubit t. Here, $i \in E_i$ means all i satisfying condition $|B_i| + |C_i| + \left|D_i^O\right| \geq 1$, where the sets are defined as follows.

$$\begin{aligned} A_i &= \left\{ \gamma : \sigma_{i|\gamma} = I, 1 \leq \gamma \leq 4N + 6cN \right\}, \\ B_i &= \left\{ \gamma : \sigma_{i|\gamma} = X, 1 \leq \gamma \leq 4N + 6cN \right\}, \\ C_i &= \left\{ \gamma : \sigma_{i|\gamma} = Y, 1 \leq \gamma \leq 4N + 6cN \right\}, \\ D_i &= \left\{ \gamma : \sigma_{i|\gamma} = Z, 1 \leq \gamma \leq 4N + 6cN \right\}. \end{aligned} \tag{A2}$$

Moreover, D_i^O denotes subset of D_i subject to the constraint that γ is an output qubit. Then we explain the meaning of $\left|\eta_t^{v_T}\right\rangle$. $v_T = \{t, r_t, \theta_t\}$ represents the fixed choice of Alice's random variables about trap qubits. In addition, $\left|\eta_t^{v_T}\right\rangle = \left|+_{\theta_t}\right\rangle$ when trap qubit t belongs to output qubits and $\left|\eta_t^{v_T}\right\rangle = \left|+_{r_t}\right\rangle$ otherwise. Note that $\left|\eta_t^{v_T}\right\rangle$ is actually the ideal state of trap qubit after all the entanglements done by Bob. In addition, T denotes positions of trap qubits, whose corresponding probability is $p\left(T\right)$. θ_t or r_t represents the value of trap qubit t corresponding to whether the trap qubit belongs to the output qubit or not. In the same way, $p\left(\theta_t\right)$ and $p\left(r_t\right)$ denote probability of choosing θ_t and r_t, respectively.

To further bound p_{error}, we use the conclusion [25] that an attack σ_i having the fewest non-trivial terms (i.e., $\sigma_{i|\gamma} \in \{X, Y, Z\}$ if γ is an output qubit, or $\sigma_{i|\gamma} \in \{X, Y\}$ if γ is not an output qubit) corresponds to the maximal p_{error}. Moreover, there is at least one non-trivial Pauli attack in the set E_i. Combining both points, we can achieve the maximal p_{error} when there is exactly one non-trivial Pauli attack. Assume that the position of the single non-trivial attack Bob does is β. β belongs to either P_{v_β} or A_{e_β}. For convenience, we use F_β to denote P_{v_β} or A_{e_β} uniformly, where $F_\beta = P_{v_\beta}$ if β belongs to a primary location P_{v_β} and $F_\beta = A_{e_\beta}$ if β belongs to a primary location A_{e_β}. Then according to (C.14) of [25], the maximal p_{error} becomes

$$p_{error} \leq \max_{i \in E_i} \sum_{t_\beta \in F_\beta} \sum_{\theta_{t_\beta}, r_{t_\beta}} p\left(t_\beta\right) p\left(\theta_{t_\beta}\right) p\left(r_{t_\beta}\right) \left(\left\langle \eta_{t_\beta}^{v_T} \right| \sigma_{i|t_\beta} \left| \eta_{t_\beta}^{v_T} \right\rangle \right)^2. \tag{A3}$$

Now we divide F_β into three cases to analyse the upper bound of p_{error}. The first case is that the non-trivial attack acts on the output primary location $P^O_{v_\beta}$, i.e., $F_\beta = P^O_{v_\beta}$.

$$\begin{aligned} p_{error} &\leq \max_{i\in E_i} \sum_{t_\beta \in P^O_{v_\beta}} \sum_{\theta_{t_\beta}, r_{t_\beta}} p\left(t_\beta\right) p\left(\theta_{t_\beta}\right) p\left(r_{t_\beta}\right) \left(\left\langle \eta^{v_T}_{t_\beta} \middle| \sigma_{i|t_\beta} \middle| \eta^{v_T}_{t_\beta} \right\rangle\right)^2 \\ &= \max_{i\in E_i} \frac{1}{16} \sum_{t_\beta \in P^O_{v_\beta}} \sum_{\theta_{t_\beta}, r_{t_\beta}} p\left(t_\beta\right) \left(\left\langle +\theta_{t_\beta} \middle| \sigma_{i|t_\beta} \middle| +\theta_{t_\beta} \right\rangle\right)^2 \\ &= \max_{i\in E_i} \frac{1}{16} \sum_{\theta_{t_\beta}, r_{t_\beta}} \left(\left(1-\frac{\alpha}{2}\right)\cdot 1 + \frac{\alpha}{2}\cdot \left(\left\langle \eta^{v_T}_{t_\beta} \middle| \sigma_{i|t_\beta} \middle| \eta^{v_T}_{t_\beta} \right\rangle\right)^2\right) \\ &\leq \max_{i\in E_i} \frac{1}{16} (16\cdot \left(1-\frac{\alpha}{2}\right) + 4\alpha) \\ &= 1-\frac{\alpha}{4}. \end{aligned} \tag{A4}$$

In the equality of the second line, we used $\theta_{t_\beta} \in \{0, \pi/4, \cdots, 7\pi/4\}$ and $r_{t_\beta} \in \{0,1\}$. In the equality of the third line, we used that the probability that any qubit of primary set is a trap qubit is $\alpha/2$, and $\sigma_{i|t_\beta}$ is non-trivial if and only if β=t_β. In the last inequality, $\sum_{\theta_{t_\beta}} \left(\left\langle +_{\theta_{t_\beta}} \middle| \sigma_{i|t_\beta} \middle| +_{\theta_{t_\beta}} \right\rangle\right)^2 \leq 4$ for any non-trivial $\sigma_{i|t_\beta} \in \{X, Y, Z\}$ is used.

The second case is that the non-trivial attack acts on the added location A_{e_β}. In the similar way, we can obtain

$$\begin{aligned} p_{error} &\leq \max_{i\in E_i} \sum_{t_\beta \in A_{e_\beta}} \sum_{\theta_{t_\beta}, r_{t_\beta}} p\left(t_\beta\right) p\left(\theta_{t_\beta}\right) p\left(r_{t_\beta}\right) \left(\left\langle \eta^{v_T}_{t_\beta} \middle| \sigma_{i|t_\beta} \middle| \eta^{v_T}_{t_\beta} \right\rangle\right)^2 \\ &= \max_{i\in E_i} \frac{1}{16} \sum_{t_\beta \in A_{e_\beta}} \sum_{\theta_{t_\beta}, r_{t_\beta}} p\left(t_\beta\right) \left(\left\langle +_{r_{t_\beta}} \middle| \sigma_{i|t_\beta} \middle| +_{r_{t_\beta}} \right\rangle\right)^2 \\ &= \max_{i\in E_i} \frac{1}{16} \sum_{\theta_{t_\beta}, r_{t_\beta}} \left(\left(1-\frac{(1-\alpha)^2}{4}\right)\cdot 1 + \frac{(1-\alpha)^2}{4}\cdot \left(\left\langle +_{r_{t_\beta}} \middle| \sigma_{i|t_\beta} \middle| +_{r_{t_\beta}} \right\rangle\right)^2\right) \\ &= -\frac{1}{4}\alpha^2 + \frac{1}{2}\alpha + \frac{3}{4}. \end{aligned} \tag{A5}$$

In the equality of the third line, we considered that the probability that any qubit of added set is a trap qubit is $(1-\alpha)^2/4$. In the equation of the fourth line, $\sum_{r_{t_\beta}} \left(\left\langle +_{r_{t_\beta}} \middle| \sigma_{i|t_\beta} \middle| +_{r_{t_\beta}} \right\rangle\right)^2 = 0$ for any non-trivial $\sigma_{i|t_\beta} \in \{X, Y\}$.

The last case is that the non-trivial attack acts on the non-output primary location $P^{NO}_{v_\beta}$, i.e., $F_\beta = P^{NO}_{v_\beta}$. Similarly, we can get the upper bound of p_{error}, which is $1-\alpha/2$.

The overall bound of p_{error} for the case of quantum output is the maximal value of all above situations. As for the case of classical output, the overall bound of p_{error} is the maximal value of above situations excluding the output primary location. We aim to determine the value of α to minimize the overall bound of p_{error}. The concrete optimal functions corresponding to quantum output and classical output are described in the Equations (1) and (2), respectively. From solving problems, we obtain the minimum 0.905 for the case of quantum output iff α is $(3-\sqrt{5})/2$, and we get the minimum 0.866 for the case of quantum output iff α is $2-\sqrt{3}$.

Appendix B. Proof of Theorem 4

Our proof follows results from the proof of Theorem 4 of [25]. In our sandglass-like graph $S(G)$, any non-trivial error (X or Y) on an added qubit is equivalent to a local error on each of two adjacent primary qubits. So one needs at least $\delta/2$ errors on the qubits of $S(G)$ to corrupt a computation. Then the

set E_i of attacks in the proof of Theorem 2 becomes $E_i' = \{i : |B_i| + |C_i| + |D_i^O| \geq \delta/2\}$. Since every location ($P_v$ or A_e) contains exactly a qubit for computation subgraph, at least $\delta/2$ non-trivial attacks should be performed on different locations to disturb computation. We denote the set of locations having at least one non-trivial attack by S_i, where $|S_i| \geq \delta/2$. According to the expression of p_{error}, the fewer non-trivial attacks result in the greater value of bound. An upper bound of p_{error} is obtained when there are exactly $\delta/2$ different locations with exactly a single non-trivial attack in each location, i.e., $|S_i| = \delta/2$. Utilizing Lemma 1, there is a subset $S_i' \subseteq S_i$ that is independently colourable locations and it contains at least $|S_i'| = \lceil \delta/2\,(2c+1) \rceil$ locations. Here we set attacks in locations of $S_i \backslash S_i'$ to be trivial ($\sigma_{i|\gamma} = I$ or Z) and attacks in locations of S_i' to be non-trivial for gaining an upper bound of p_{error}. According to the inequality (A1), we have the same expression as (E.1) of [25].

$$p_{error} \leq \max_{i \in E_i'} \prod_{\beta=1}^{|S_i'|} \sum_{t_\beta \in F_\beta} p\left(t_\beta\right) \sum_{\theta_{t_\beta}, r_{t_\beta}} p\left(\theta_{t_\beta}\right) p\left(r_{t_\beta}\right) \left(\left\langle \eta_{t_\beta}^{v_T} \right| \sigma_{i|t_\beta} \left| \eta_{t_\beta}^{v_T} \right\rangle\right)^2 \tag{A6}$$

The right side of above expression consists of the product of $|S_i'|$ items, while the upper bound of each item is 0.905. So we have $p_{error} \leq 0.905^{\left\lceil \frac{\delta}{2(2c+1)} \right\rceil}$.

References

1. Childs, A. Secure assisted quantum computation. *Quantum Info. Comput.* **2005**, *5*, 456.
2. Broadbent, A.; Fitzsimons, J.F.; Kashefi, E. Universal blind quantum computation. In Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, Atlanta, GA, USA, 25–27 October 2009; p. 517.
3. Barz, S.; Kashefi, E.; Broadbent, A.; Fitzsimons, J.F.; Zeilinger, A.; Walther, P. Demonstration of blind quantum computing. *Science* **2012**, *335*, 303. [CrossRef] [PubMed]
4. Morimae, T.; Fujii, K. Blind topological measurement-based quantum computation. *Nat. Commun.* **2012**, *3*, 1036. [CrossRef] [PubMed]
5. Morimae, T. Continuous-variable blind quantum computation. *Phys. Rev. Lett.* **2012**, *109*, 230502. [CrossRef]
6. Morimae, T.; Fujii, K. Blind quantum computation protocol in which Alice only makesmeasurements. *Phys. Rev. A* **2013**, *87*, 050301. [CrossRef]
7. Morimae, T.; Fujii, K. Secure entanglement distillation for double-server blind quantum computation. *Phys. Rev. Lett.* **2013**, *111*, 020502. [CrossRef]
8. Mantri, A.; Pérez-Delgado, C.A.; Fitzsimons, J.F. Optimal blind quantum computation. *Phys. Rev. Lett.* **2013**, *111*, 230502. [CrossRef]
9. Giovannetti, V.; Maccone, L.; Morimae, T.; Rudolph, T.G. Efficient universal blind quantum computation. *Phys. Rev. Lett.* **2013**, *111*, 230501. [CrossRef]
10. Sueki, T.; Koshiba, T.; Morimae, T. Ancilla-driven universal blind quantum computation. *Phys. Rev. A* **2013**, *87*, 060301. [CrossRef]
11. Morimae, T.; Dunjko, V.; Kashefi, E. Ground state blind quantum computation on AKLT state. *Quantum Inf. Comput.* **2015**, *15*, 200.
12. Pérez-Delgado, C.A.; Fitzsimons, J.F. Iterated gate teleportation and blind quantum computation. *Phys. Rev. Lett.* **2015**, *114*, 220502.
13. Takeuchi, Y.; Fujii, K.; Ikuta, R.; Yamamoto, T.; Imoto, N. Blind quantum computation over a collective-noise channel. *Phys. Rev. A* **2016**, *93*, 052307. [CrossRef]
14. Fitzsimons, J.F. Private quantum computation: An introduction to blind quantum computing and related protocols. *NPJ Quantum Inf.* **2017**, *3*, 23. [CrossRef]
15. Zhang, X.; Weng, J.; Tan, X.; Song, T.; Luo, W. Measurement-based universal blind quantum computation with minor resources. *arXiv* **2018**, arXiv:1801.03090.
16. Zhang, X.; Weng, J.; Li, X.; Luo, W.; Tan, X.; Song, T. Single-server blind quantum computation with quantum circuit model. *Quantum Inf. Process.* **2018**, *17*, 134. [CrossRef]
17. Fitzsimons, J.F.; Kashefi, E. Unconditionally verifiable blind quantum computation. *Phys. Rev. A* **2017**, *96*, 012303. [CrossRef]

18. Barz, S.; Fitzsimons, J.F.; Kashefi, E.; Walther, P. Experimental verification of quantum computation. *Nat. Phys.* **2013**, *9*, 727. [CrossRef]
19. Hayashi, M.; Morimae, T. Verifiable measurement-only blind quantum computing with stabilizer testing. *Phys. Rev. Lett.* **2015**, *115*, 220502. [CrossRef]
20. Gheorghiu, A.; Kashefi, E.; Wallden, P. Robustness and device independence of verifiable blind quantum computing. *New J. Phys.* **2015**, *17*, 083040. [CrossRef]
21. Hajdušek, M.; Pérez-Delgado, C.A.; Fitzsimons, J.F. Device-independent verifiable blind quantum computation. *arXiv* **2015**, arXiv:1502.02563.
22. McKague, M. Interactive proofs for BQP via self-tested graph states. *Theor. Comput.* **2016**, *12*, 1. [CrossRef]
23. Greganti, C.; Roehsner, M.-C.; Barz, S.; Morimae, T.; Walther, P. Demonstration of measurement-only blind quantum computing. *New J. Phys.* **2016**, *18*, 013020. [CrossRef]
24. Gheorghiu, A.; Wallden, P.; Kashefi, E. Rigidity of quantum steering and one-sided device-independent verifiable quantum computation. *New J. Phys.* **2017**, *19*, 023043. [CrossRef]
25. Kashefi, E.; Wallden, P. Optimised resource construction for verifiable quantum computation. *J. Phys. A Math. Theor.* **2017**, *50*, 145306. [CrossRef]
26. Hayashi, M.; Hajdušek, M. Self-guaranteed measurement-based quantum computation. *Phys. Rev. A* **2018**, *97*, 052308. [CrossRef]
27. Fitzsimons, J.F.; Hajdušek, M.; Morimae, T. Post hoc Verification of Quantum Computation. *Phys. Rev. Lett.* **2018**, *120*, 040501. [CrossRef]
28. Gheorghiu, A.; Kapourniotis, T.; Kashefi, E. Verification of quantum computation: An overview of existing approaches. *Theory Comput. Syst.* **2019**, *63*, 715. [CrossRef]
29. Takeuchi, Y.; Morimae, T.; Mizutani, A.; Fitzsimons, J.F. Resource-efficient verification of quantum computing using Serfling's bound. *NPJ Quantum Inf.* **2019**, *5*, 27. [CrossRef]
30. Raussendorf, R.; Browne, D.E.; Briegel, H.J. Measurement-based quantum computation on cluster states. *Phys. Rev. A* **2003**, *68*, 022312. [CrossRef]
31. Briegel, H.J.; Browne, D.E.; Dür, W.; Raussendorf, R.; Van den Nest, M. Measurement-based quantum computation. *Nat. Phys.* **2009**, *5*, 19. [CrossRef]
32. Raussendorf, R.; Briegel, H.J. A one-way quantum computer. *Phys. Rev. Lett.* **2001**, *86*, 5188. [CrossRef] [PubMed]
33. DiVincenzo, D.P.; Shor, P.W. Fault-tolerant error correction with efficient quantum codes. *Phys. Rev. Lett.* **1996**, *77*, 3260. [CrossRef] [PubMed]
34. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*; Cambridge University Press: Cambridge, UK 2010; p. 425–493.
35. Raussendorf, R.; Harrington, J.; Goyal, K. A fault-tolerant one-way quantum computer. *Ann. Phys.* **2006**, *321*, 2242. [CrossRef]
36. Raussendorf, R.; Harrington, J.; Goyal, K. Topological fault-tolerance in cluster state quantum computation. *New J. Phys.* **2007**, *9*, 199. [CrossRef]
37. Raussendorf, R.; Harrington, J. Fault-tolerant quantum computation with high threshold in two dimensions. *Phys. Rev. Lett.* **2007**, *98*, 190504. [CrossRef]