Quantum Science and Technology

Blind oracular quantum computation

To cite this article: Cica Gustiani and David P DiVincenzo 2021 *Quantum Sci. Technol.* **6** 045022

Quantum Science and Technology

OPEN ACCESS

RECEIVED
12 November 2020

REVISED
19 June 2021

ACCEPTED FOR PUBLICATION
13 July 2021

PUBLISHED
22 September 2021

PAPER

Blind oracular quantum computation

Cica Gustiani[1,2,*] and David P DiVincenzo[1,3,4]

[1] Institute for Quantum Information, RWTH Aachen University, D-52056 Aachen, Germany
[2] Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, United Kingdom
[3] Peter Grünberg Institute, Theoretical Nanoelectronics, Forschungszentrum Jülich, D-52425 Jülich, Germany
[4] Jülich-Aachen Research Alliance (JARA), Fundamentals of Future Information Technologies, D-52425 Jülich, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: cica.gustiani@materials.ox.ac.uk and d.divincenzo@fz-juelich.de

Keywords: oracle separation, blind quantum computing, composable security, solid-state quantum server, lazy computation, measurement-based quantum computation, abstract cryptography

Abstract

In the standard oracle model, an oracle efficiently evaluates an unknown classical function independent of the quantum algorithm itself. Quantum algorithms have a complex interrelationship to their oracles; for example the possibility of quantum speedup is affected by the manner by which oracles are implemented. Therefore, it is physically meaningful to separate oracles from their quantum algorithms, and we introduce one such separation here. We define the *blind oracular quantum computation* (BOQC) scheme, in which the oracle is a distinct node in a quantum network. Our work augments the client–server setting of quantum computing, in which a powerful quantum computer server is available on the network for discreet use by clients on the network with low quantum power. In BOQC, an oracle is another client that cooperates with the main client so that an oracular quantum algorithm is run on the server. The cooperation between the main client and the oracle takes place (almost) without communication. We prove BOQC to be *blind*: the server cannot learn anything about the clients' computation. This proof is performed within the composable security definitions provided by the formalism of abstract cryptography. We enhance the BOQC scheme to be runnable with minimal physical qubits when run on a solid-state quantum network; we prove that this scheme, which we refer to as BOQCo (BOQC-optimized), possesses the same security as BOQC.

1. Introduction

Oracle constructions in quantum algorithms provide an essential conceptual framework for understanding quantum speedups. The detailed interrelationship between oracle properties and algorithmic efficiency is complex: an interesting example arises in the Grover algorithm, where the quantum speedup becomes impossible if the oracle has a small probability of failing on every call [1]. Moreover, for some quantum algorithms, adding internal dice to an oracle introduces a strong separation. For example, Simon's algorithm, when the oracle has internal dice, is unsolvable on classical computers while it is solvable in a linear time on quantum computers [2]. In the standard model, a quantum oracle is specified as a unitary map $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$, where $f\colon \{0, 1\}^n \rightarrow \{0, 1\}$ indicates a subroutine whose code we cannot usefully examine or a 'black box' whose properties we would like to estimate.[5] While oracle constructions have been considered artificial, we aim to introduce and analyze a multiparty setting for which the oracle paradigm is physically meaningful.

[5] An equivalent oracle model, reducible to the standard oracle model, is the *phase oracle*, represented as a map $|x, y\rangle \mapsto (-1)^{y\cdot f(x)}\,|x, y\rangle$ [3]. In the standard oracle model, f is a deterministic classical function; however, some generalizations are introduced in [2, 4], where f can be a probabilistic classical function.

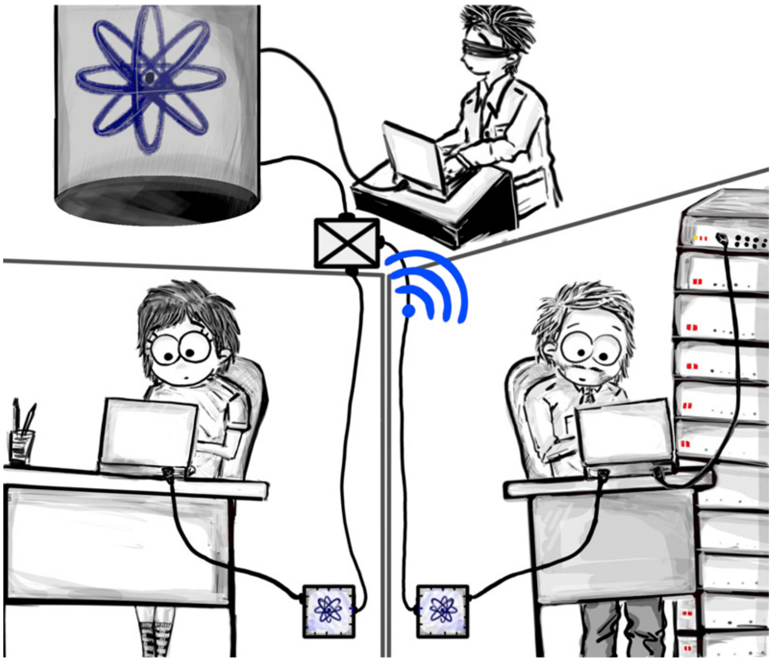


Figure 1. [5] Alice (left) wants to run a private oracular quantum algorithm, but she has neither a powerful quantum computer nor the knowledge needed to construct her oracle function. Oscar (right) is equipped with a device that evaluates the oracle functions, e.g. it computes stocks' prediction, gives access to a database, or performs massive classical computations. Bob (top), who owns a powerful quantum computer, offers a quantum computing service. However, Alice and Oscar do not trust Bob's quantum computer, and they do not risk revealing information about their computation while running it. Therefore, they execute the computation within the BOQC scheme in which Bob is blinded to the computation. All players are connected to an insecure classical channel (the wireless). Alice and Oscar still need a small quantum power—the small boxes in the bottom, making their computation power 'almost classical'. Such power is necessary because obtaining perfect blindness with purely classical clients is still an open problem [6].

Our view of the near-term situation in the development of quantum processing is that there will be quantum computers of some moderate power (the *servers* of the discussion below) owned by particular organizations which offer their service on a quantum internet to clients with small quantum power.[6] We will further assume that there will be nodes on the quantum internet, also with modest quantum processing power, but in possession of some special information or data—the *oracles*. Consider figure 1 for an illustration. We introduce a protocol where a client (Alice), who is aware of these oracle resources on the network, needs to delegate her oracular quantum algorithm to an untrustworthy server. We refer to our scheme as blind oracular quantum computation (BOQC). The BOQC protocol views the oracle as a trustworthy third party.[7] We will consider the server quantum computer (Bob) to be an untrustworthy party, therefore we adopt the concept of *blindness* [9–12], where the server can learn nothing about the algorithm that is running and nothing about the measurement outcomes. Our protocol is set to run any of the family of quantum oracular algorithms, an extensive catalog of which can be found in [13].

BOQC is an extension of *universal blind quantum computation* (UBQC) [10], enabling computation controlled by two cooperating clients (the main client and the oracle) that are not interacting during the execution of the computation. We will show that BOQC is *composably blind* using the *abstract cryptography* (AC) framework [14]; references closely related to our work are [15, 16]. UBQC allows the desired delegation of computation by a client to an untrustworthy server, where the algorithms are implemented within a measurement-based computation model, that is the *one-way quantum computer* (1WQC) [17, 18]. The essential resource of 1WQC is a highly entangled qubit state, i.e. a graph state or a cluster state.

The idea of our choice to use the 1WQC model for our protocol stems from an important property of quantum maps over a graph state: combining two quantum operators (graphs) is performed by simply connecting the graphs. Thus, outsourcing a quantum operation to another party means outsourcing the corresponding graph. This outsourcing is not so straightforward in the conventional gate-based quantum computation model.

[6] Small quantum power on clients is needed for a single quantum server. If two non-interacting quantum servers share EPR pairs, an entirely classical client can perform blind quantum computations [7]; an experiment has been conducted in such a setting [8].
[7] We also consider a party with oracle access as a party with more classical power.

Initially, the 1WQC paradigm was tailored to perform quantum computations on ultracold atoms in optical lattice systems on which the resource state (cluster states) can be generated efficiently [19]. In the 1WQC paradigm, the computation is done by systematically 'consuming' the resource via measurements—thus, it is called a 'one-way computer'. It turns out that the 1WQC paradigm can also be efficiently performed on linear optics systems, whose qubits are memoryless. Several small-scale 1WQC computations have been experimentally conducted on linear optical systems [20–25].

However, performing adaptive measurement using memoryless qubits remains challenging. In our companion paper [26], we have sought possibilities beyond memoryless qubits by considering solid-state qubits, since here performing adaptive measurements can be more feasible. However, producing graph states is very demanding, e.g. running an exact quantum search algorithm within a BOQC scheme needs more than 90 qubits for searching one item within five [26]. Those motivate us to push forward the implementations of 1WQC using solid-state qubits and to diminish the massive requirement for physical qubits: we propose to prepare a graph state by parts, in a 'just-in-time' fashion in which qubits are prepared only when needed.[8] We call our scheme *lazy 1WQC*, for which we require qubits that possess permanence and can be rapidly re-initialized.

Note that, the reusing-qubits scheme has been analyzed in in [27], which obtains bounds on the number of physical qubits needed. However, here, we provide lazy 1WQC as a clear-cut and explicit scheme that is provably correct. Then, we integrate lazy 1WQC into the BOQC scheme, producing BOQCo (BOQC-optimized). BOQCo is BOQC in which the number of physical qubits is minimal, given that Bob's quantum computer is a solid-state system whose qubits possess permanence and can be rapidly re-initialized. In the BOQCo scheme, the exact search of one item within five exactly requires 4 physical qubits [26].

2. Preliminaries

2.1. Measurement-based computation and formalism

Here we review *one-way quantum computing* (1WQC) [17, 18], the measurement calculus [28], and deterministic computations made possible by the existence of *flow* [29]. A computation within 1WQC is executed by consecutively measuring qubits in a *cluster state*: a highly entangled quantum state, which can be efficiently parameterized by mathematical graphs [30]. A cluster state corresponds to the space-time layout of the quantum computer, and consecutive measurements define quantum operations. In this study, we use measurement calculus to describe processes within a 1WQC computation, and we use flow to specify measurement-dependency structure.

A graph is used to represent the resource (cluster state) of a 1WQC computation. The graph's vertices represent qubits whose states are initially in the xy-plane of Bloch sphere, and its edges represent CPHASE gates applied to the corresponding nodes. We will interchangeably use graph and graph state. In particular, *open graph* is used as a 1WQC resource, that is a triplet $(\mathcal{G}, I, O)$, with a set of quantum input nodes I and quantum output nodes O that may intersect, where $\mathcal{G} = (V, E)$ is a simple graph[9] with a set of vertices V and edges E, $I \subset V$ and $O \subset V$, where $I \neq \varnothing$, and $O \neq \varnothing$. For a subset $K \subset V$, $\mathcal{G}[K]$ denotes the induced subgraph whose vertex set is K and whose edge set is those from E whose endpoints are both in K—denoted as $E(K)$. Given a node k, we define the following notations:

$$\begin{aligned} &N_{\mathcal{G}}(k) \text{ as nodes adjacent to } k \text{ — called open neighborhood,} \\ &N_{\mathcal{G}}[k] \text{ as nodes adjacent to } k, \text{including } k \text{ — called closed neighborhood.} \end{aligned} \tag{1}$$

Another formalism that we use here is measurement calculus on *measurement patterns* (or patterns) [28] in order to describe processes within the 1WQC scheme. A pattern comprises commands: (i) $N_j^{\theta} :=$ prepare qubit j in state $|+_{\theta}\rangle$,

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

(ii) $E_{ij} :=$ apply CPHASE between qubits i, j. (iii) $M_j^{\theta} :=$ measure j in the basis $|\pm_{\theta}\rangle$. Finally (iv) X_i^s, Z_i^s are Pauli corrections on qubit i for signal s; that is, if $s = 1$, the corrections $X_i^1 = X_i, Z_i^1 = Z_i$ are done, and if $s = 0$ no corrections are done since $X_i^0 = Z_i^0 = \mathbb{1}_i$. In addition, we denote $N_Q^{\theta} := \bigotimes_{i \in Q} N_i^{\theta_i}$ as preparing a set of qubits Q accordingly.

[8] The just-in-time fashion has the same principle as the *lazy computation* scheme that is common programming practice.

[9] Simple graphs are a class of graphs without direction, without self loops, and without multiple edges.

Apart from those commands, we introduce the following extra notations. First, $Z_i(\theta)$ signifies rotation about z-axis on qubit i with angle θ; in particular

$$Z(\theta) := \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\theta} \end{pmatrix}. \tag{3}$$

Second, given a graph $\mathcal{G} = (V, E)$ with an ordering $>$ on the nodes, for any subgraph $\mathcal{G}[K]$ that $K \subset V$,

$$E_K := \prod_{(i,j)\in E(K)} E_{ij}, \qquad E_{iK} := \prod_{k\in K} E_{ik} \quad \text{where} \quad i \notin K, \quad E^{>}_{iK} := \prod_{k\in K, k>i} E_{ik}, \tag{4}$$

where E_{ij} is the entangling command (ii) in measurement calculus; E_{ij} are all mutually commuting.

Note that Pauli corrections can be absorbed into measurement angles [28]:

$$M^{\theta}_j X_j = M^{(-\theta)}_j \qquad M^{\theta}_j Z_j = M^{\theta+\pi}_j, \tag{5}$$

where $-\theta$ and $\theta + \pi$ are understood to be evaluated modulo 2π.

A set of angles $\theta = \{\theta_j\}_{j\in\mathbb{N}}$ specifies quantum operations; angle θ_j can denote the parameter of a projective measurement performed on node j with projectors

$$\left\{ \left(\left|+_{\theta_j}\right\rangle \left\langle +_{\theta_j}\right|, \mathtt{0} \right), \left(\left|+_{\theta_j+\pi}\right\rangle \left\langle +_{\theta_j+\pi}\right|, \mathtt{1} \right) \right\}, \tag{6}$$

where $\theta_j \in [0, 2\pi)$. The two projectors in (equation (6)) will be reported as outcomes $\mathtt{0}$ and $\mathtt{1}$ respectively. We also refer to such a measurement as 'measure in basis $\left|\pm_{\theta_j}\right\rangle$'. An angle θ_j may be dependent on signals $s_{<j}$, that is, measurement outcomes obtained previous to measuring qubit j. It is inevitable that measurements introduce indeterminacies. Thus, adaptive measurements—measurements which are depending on some prior signals—are performed. The adaptive measurement in θ_j may be X_i- or Z_i-dependent on a signal i. If it is X_i-dependent, θ_j is replaced by $(-1)^{s_i}\theta_j$; if it is Z_i-dependent, θ_j is replaced by $\theta_j + s_i\pi$; these replacements are indicated in equation (5). This dependency structure is captured within the notion of *flow*. It is worth mentioning that it is possible in 1WQC to perform measurements other than those in the *xy*-plane, such as those in the *xz*- or *yz*-plane [31]. However, we consider only *xy*-plane measurements here.

A *flow f* is a map from the measured qubit to the prepared qubit. $f : O^{\mathrm{c}} \to I^{\mathrm{c}}$, where A^{c} denotes the complement of set A. More specifically for cluster states, $f(j)$ indicates the X correction and $N_{\mathcal{G}}(f(j))$ indicate Z corrections for the measured node j. By its definition, we will see that a flow induces a partial order that covers V. The state of an open graph $(\mathcal{G}, I, O)$ has flow if there exists a map $f : O^{\mathrm{c}} \to I^{\mathrm{c}}$ together with a partial order $\succ$ over nodes V such that for all $j \in O^{\mathrm{c}}$ [29, 31]:

$$(\mathrm{F0})\ (j, f(j)) \in E, \quad (\mathrm{F1})\ f(j) \succ j, \quad (\mathrm{F2})\ \forall\ k \in N_{\mathcal{G}}(f(j)) \setminus \{j\}, k \succ j. \tag{7}$$

Hence, a 1WQC computation can be described with a set

$$\{(\mathcal{G}, I, O), f, \phi, \rho^{\mathrm{in}}\}, \tag{8}$$

where $(\mathcal{G}, I, O)$ denotes an open graph with flow f, ϕ signifies a set of measurement angles, and ρ^{in} is a quantum state assigned to nodes I.

Reference [29] also characterizes an interesting family of patterns as follows.

Theorem 1. *[29] Suppose the open graph state $(\mathcal{G}, I, O)$ has flow $(f, \succ)$, then the pattern:*

$$\mathfrak{P}_{f,\mathcal{G},\succ,\vec{\theta}} := \overset{\succ}{\prod_{i\in O^c}} \left(X^{s_i}_{f(i)} \left(\prod_{k\in N_{\mathcal{G}}(f(i))\setminus\{i\}} Z^{s_i}_k \right) M^{\theta_i}_i \right) E_G N^0_{I^c} \tag{9}$$

is runnable *and deterministic for all $\vec{\theta}$ and $\vec{s}$. It realizes the isometry $\bigotimes_{i\in O^{\mathrm{c}}} \langle +_{\theta_i}|_i E_{\mathcal{G}} N^0_{I^{\mathrm{c}}}$, where $E_{\mathcal{G}} := \prod_{(i,j)\in\mathcal{G}} E_{ij}$.*

A pattern is said to be runnable [29] if it satisfies the following conditions: (R0) no command depends on an outcome not yet measured, (R1) no command acts on a qubit already measured or not yet prepared (except preparation commands), and (R2) a qubit i is measured (prepared) if and only if i is not an output (input).

Theorem 1 provides a sufficient condition in which the existence of flow guarantees a deterministic computation. Notice that in equation (9) we introduce the notation $\overset{\succ}{\prod}$, signifying an ordered product with respect to any given ordering $\succ$.

2.2. Abstract cryptography

Abstract cryptography (AC) [14] is a mathematical framework that we use to model the security of our protocols. The AC framework ensures *composability*: a secure protocol in AC is guaranteed to be secure when composing a larger cryptographic system. AC is a constructive cryptography [32], i.e. we are constructing a resource from other weaker resources. To assess the security in such a construction, AC implements an 'ideal-world real-world' paradigm in which the distance of resources is used as the security metric. Simply put, we want to construct an ideal resource (resources with desired security in the ideal world) from a real resource (resources found in the real world), while minimizing the distance between the two worlds (or systems). Perfect security is achieved when both systems are completely indistinguishable.

AC uses a top-down approach: it starts from the highest-level possible of abstraction, then proceeds downwards, introducing in each level the minimum necessary specializations. The framework defines a *system* as an abstract object with interfaces. There are two types of systems: *resources* and *converters*. A *resource* is a system with a set of interfaces $\mathcal{I} = \{A, B, C, \ldots\}$; each element $i \in \mathcal{I}$ is associated with a player, which models how party i can access the resource. A *converter* is a system with two interfaces: an *inside* interface that is connected to a resource and the *outside* interface that receives inputs and gives outputs.

A *(cryptographic) protocol* is a set of converters possessed by honest parties $\{\pi_i\}$. Converters π_i, π_j can be appended to a resource $\mathcal{R}$ via their inside interfaces,[10] forming a new resource $\pi_i\pi_j\mathcal{R}$ with the same set of interfaces.[11] Resources and converters can be instantiated with any mathematical object that follows the AC composition defined in [14]; in our case, we can instantiate them with *quantum combs* [33]. The quantum comb generalizes quantum channels, mapping quantum circuits to quantum circuits rather than quantum states to quantum states. Thus, here the compositions may be defined as the operations of quantum combs as well.

A practical view of a protocol is seeing it as an effort to obtain the ideal-world functionalities from the real-world functionalities. Note that this framework does not capture the kind of failure, the severity of failure, nor the cheating strategy. Instead, we ask if the defined ideal resource captures all the security features that we need; otherwise, we define it differently.

Consider a three-party protocol π with players Alice($\mathcal{A}$), Oscar($\mathcal{O}$), and Bob($\mathcal{B}$), where Bob can be dishonest. There are two security requirements that we want to achieve with the protocol π: *correctness* (or *completeness*) and *security* (or *soundness*). Correctness is a property that can be present only when all parties are honest (no adversary present), defined in definition 1. Security is a property involving the presence of an adversary (or adversaries), defined in definition 2 for a cheating Bob.

Definition 1 [14]**.** Let $\pi = (\pi_{\mathcal{A}}, \pi_{\mathcal{O}}, \pi_{\mathcal{B}})$ be a protocol with no adversary and $\mathcal{R}, \mathcal{S}$ be resources. Protocol π is ε-correct if it constructs an ideal resource within ε, viz, $\mathcal{R} \xrightarrow{\pi,\varepsilon} \mathcal{S}$, such that

$$d(\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}\pi_{\mathcal{B}}, \mathcal{S}) \leqslant \varepsilon.$$

Definition 2 [14]**.** Let $\pi = (\pi_{\mathcal{A}}, \pi_{\mathcal{O}})$ be a protocol with an adversary $\mathcal{B}$ and $\mathcal{R}, \mathcal{S}'$ be resources. Protocol π is ε-secure, that is, it constructs an ideal resource within ε, viz, $\mathcal{R} \xrightarrow{\pi,\varepsilon} \mathcal{S}'$, if there exists a converter $\sigma_{\mathcal{B}}$ (called *simulator*), such that

$$d(\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}, \mathcal{S}'\sigma_{\mathcal{B}}) \leqslant \varepsilon.$$

Here, d signifies a pseudo-metric such that: $d(\mathcal{R}, \mathcal{R}) = 0$, $d(\mathcal{R}, \mathcal{S}) = d(\mathcal{S}, \mathcal{R})$, and $d(\mathcal{R}, \mathcal{S}) \leqslant d(\mathcal{R}, \mathcal{T}) + d(\mathcal{T}, \mathcal{S})$. If both definitions above are fulfilled, we say protocol π is ε-secure in producing tasks defined in $\mathcal{S}$, using resources $\mathcal{R}$, and ε is the probability of failing. For $\varepsilon = 0$, we call the protocol π perfectly secure.

In definition 2, with Bob being dishonest, an arbitrary system (simulator $\sigma_{\mathcal{B}}$) is appended to $\mathcal{S}'$ at the $\mathcal{B}$-interface in order to make both systems ($\mathcal{S}'$ and $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}$) comparable. The system $\mathcal{S}'\sigma_{\mathcal{B}}$ is also called a *relaxation* of $\mathcal{S}'$ [34], where the definition of simulator $\sigma_{\mathcal{B}}$ is independent of Bob's cheating strategies. A simulator assures that none of these relaxations can be more useful to Bob than the ideal resource. As all relaxations are defined in the ideal world, a real-world system is secure when it is indistinguishable from at least one relaxation of the ideal system.

It now remains for us to practically specify 'distinguishing two resources'; here, we use the notion of *advantage*.[12] Two resources $\mathcal{R}$ and $\mathcal{S}$ are indistinguishable if the rest of the world cannot tell whether it is interacting with $\mathcal{R}$ or $\mathcal{S}$; a *distinguisher* captures the rest of the world. One can think of a distinguisher as a referee who has some access to and can freely interact with an unknown system ($\mathcal{R}$ or $\mathcal{S}$). A distinguisher can read outputs, give inputs, take the role of an adversary, generate an arbitrary joint system, measure a

[10] Note that interfaces of $\mathcal{R}$ contain i and j.

[11] The writing order is arbitrary, i.e. $\pi_i\pi_j\mathcal{R} = \pi_j\pi_i\mathcal{R} = \mathcal{R}\pi_i\pi_j = \mathcal{R}\pi_j\pi_i = \pi_i\mathcal{R}\pi_j = \pi_j\mathcal{R}\pi_i$.

[12] Advantage here defined as the probability of guessing correctly minus guessing erroneously.

joint system, and measure a purification.[13] The distinguisher is then asked whether it is interacting with $\mathcal{R}$ ($B = 0$) or $\mathcal{S}$ ($B = 1$). In this setting, the distance metric is called *distinguishing advantage*. Given D as a random variable that signifies the distinguisher's guess, the distinguishing advantage is defined as

$$\left|\Pr[D = 0|B = 0] - \Pr[D = 0|B = 1]\right|, \tag{10}$$

which is the difference between guessing correctly and erroneously. Perfect security is accomplished when the distinguishing advantage is zero.

3. The BOQC

3.1. Pre-protocol

This paper aims to provide a mechanism for a client to privately delegate her oracular computation with the cooperation of a separate oracle client. For this we propose our scheme, which we call BOQC. There are several BOQC protocols provided here, which vary based on the used resources, e.g. solid-state qubits, photonic qubits, classical or quantum inputs, and based on whether the outputs are classical or quantum.

Consider the following illustration to give a picture of a situation in which a BOQC will be used. Alice wants to run a private oracular quantum algorithm,[14] but she has neither a powerful quantum computer nor the resources needed to construct her oracle function. On the other hand, Oscar has the information necessary to generate an oracle quantum circuit. Also, Bob has a quantum computer, offered to clients as a service. While building a reliable quantum computer is very hard, in our idealized setting Alice and Oscar will be motivated to use Bob's quantum computer. However, they do not wish to risk revealing information about their computation while running it on Bob's quantum computer. Hence, they run the computation using the BOQC scheme. Consider figure 1 that pictures this situation.

We assume that Alice and Oscar (the clients) possess the same level of quantum power, and are always honest. The client-cooperation scheme is implemented with minimal shared information, where Alice and Oscar are not communicating during the protocol run; this is possible by the nature of an oracle in an algorithm, which is a computation independent of the algorithm itself. Since the algorithm is run within the 1WQC scheme, Alice and Oscar represent their algorithms as graphs and measurement angles. In particular, their preparations are captured in definition 3, the pre-protocol steps, which are performed before the protocol runs.

Definition 3. [5] Pre-protocol steps. Given that Oscar agreed to provide oracle information for a quantum computation that Alice wishes to run, the following steps are done via an authentic channel before starting a BOQC protocol:

(B1) Alice and Oscar determine the size of bit string b; the string must be long enough to indicate all possible measurement angles $\phi_i, \psi_i \in \Omega$. The allowed set of angles is $\Omega = \{\pi k/2^{b-1}\}_{0\leqslant k<2^b}$.

(B2) Alice and Oscar join their graphs in the following way. Given that Alice's oracular quantum algorithm needs k oracle queries, she marks each query as a black box on her graph $\mathcal{A}$. Given that Oscar's graph $\mathcal{O}$ is a graph with k components, he sends Alice $\{\mathcal{O}, f_{\mathcal{O}}\}$. Alice joins their graphs with a connection C by replacing each black box in $\mathcal{A}$ with a component of $\mathcal{O}$ according to C; she obtains $\mathcal{G} = \mathcal{A}\cup_C\mathcal{O}$, and computes the total flow $(f, \succ)$ for the entire graph $\mathcal{G}$. The connection C is valid when the resulting graph $\mathcal{G}$ has flow.[15]

(B3) Alice determines a total ordering $>$ that is consistent with the partial ordering $\succ$. This step is optional for the BOQC protocols (protocols 1 and 3), but it is necessary for optimized protocols such as BOQCo protocols (protocols 2 and 4).

(B4) Alice publicly informs Bob of $\{(\mathcal{G}, \tilde{I}, \tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$, where $\tilde{I} \subseteq I$ is a set of nodes assigned with quantum input ρ^{in}, $\tilde{O} \subseteq O$ is a set unmeasured nodes that will be sent to Alice, $V_{\mathcal{A}} \equiv V(\mathcal{A})$, and $V_{\mathcal{O}} \equiv V(\mathcal{O})$. If $\tilde{I} \neq \varnothing$ and $\tilde{O} \neq \varnothing$, then $\tilde{I} \cap V_{\mathcal{O}} = \varnothing$ and $\tilde{O} \cap V_{\mathcal{O}} = \varnothing$, respectively; simply put, Oscar does not give quantum input and does not receive quantum output. The total order $>$ is given if step **(B3)** is executed.

Notice that we introduce $\tilde{I}$ and $\tilde{O}$ in addition to I and O. While variables I and O are required to define a 1WQC computation (equation (8)) such that one can describe the resulting computation per theorem 1, variables $\tilde{I}$ and $\tilde{O}$ signify nodes whose quantum information is associated with Alice. Set $\tilde{I}$ comprises nodes assigned with Alice's quantum input and $\tilde{O}$ is a set of nodes whose state is received by Alice as the final output. For a classical input $c = c_n \ldots c_2c_1$, where $c_i \in \{0, 1\}$, we can consider it as quantum input with

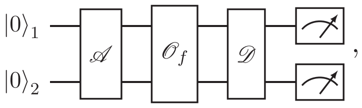
[13] Involvement of purification systems is related to hiding information explained in [35].

[14] Some examples of oracular algorithms can be found in reference [13].

[15] See example 1 that shows the process of this step explicitly.

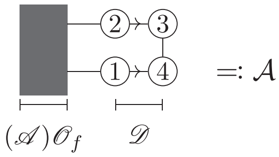
state $\prod_{i=1}^{n} |c_i\rangle\langle c_i|$ in the formalism used in theorem 1. Also, a classical output can be considered as quantum output—in the formalism used in theorem 1—with a density of states whose matrix is diagonal.

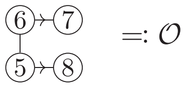
Example 1. Alice wants to run a two-qubit Grover algorithm and Oscar has the four-element database. The algorithm comprises one oracle call and is implementable with circuit



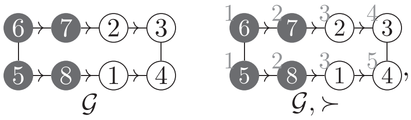
where $\mathscr{A}$, $\mathscr{O}_f(\pi)$ and $\mathscr{D}(\pi)$ indicate preparation, oracle, and diffusion operators respectively. The following steps show the steps in definition 3 to run a two-qubit Grover algorithm; the algorithm (the graph state) is taken from [5].

First Alice and Oscar agree upon the bit string size, e.g. $b = 4$. As Alice does not need any quantum input and output, so she sets $\tilde{I} = \tilde{O} = \varnothing$. Alice's graph state $\mathcal{A}$ contains a black box indicating the oracle. Then, Oscar tells Alice his graph and flow $\{\mathcal{O}, f_{\mathcal{O}}\}$, where $\mathcal{O}$ is a graph with one component. The following shows graphs of Alice and Oscar whose flows are indicated with arrows.





Here, $\mathcal{A} = (\{1, 2, 3, 4\}, \{(2, 3), (1, 4), (3, 4)\})$ and $\mathcal{O} = (\{5, 6, 7, 8\}, \{(5, 6), (6, 7), (5, 8)\})$. Alice obtains the total graph $\mathcal{G} = \mathcal{A} \cup_C \mathcal{O}$ with $C = \{(7, 2), (8, 1)\}$ and the total flow $(f, \succ)$ as shown here:



with $I = \{5, 6\}, \tilde{I} = \varnothing$, $O = \{3, 4\}$, $\tilde{O} = \varnothing$, $V_{\mathcal{A}} = \{1, 2, 3, 4\}$, $V_{\mathcal{O}} = \{5, 6, 7, 8\}$, partial ordering $\{5, 6\} \succ \{7, 8\} \succ \{1, 2\} \succ \{3\} \succ \{4\}$, and a total ordering, e.g. $5 < 6 < 7 < 8 < 1 < 2 < 3 < 4$. Finally, Alice publicly tells Bob $\{(\mathcal{G}, \varnothing, \varnothing), V_{\mathcal{A}}, V_{\mathcal{O}}, >, \succ, 4\}$.

In this example, neither Alice nor Oscar provide classical or quantum input, and $I \subset V_{\mathcal{O}}$; this is consistent with definition 3 since $\tilde{I} = \varnothing$. Instead, here the input is implicit, i.e. two zeros $|00\rangle\langle 00|$ (and).

3.2. The protocol

The BOQC protocol involves three players: Alice, Bob, and Oscar, indicated with $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{O}$, respectively. In the language of AC, we denote the protocol as $\pi_{\text{boqc}} = (\pi_{\mathcal{A}}, \pi_{\mathcal{B}}, \pi_{\mathcal{O}})$—with an honest Bob—which uses a real-world resource $\mathcal{R}$. The resource $\mathcal{R}$ comprises three channels: a secure key channel between Alice and Oscar, a two-way insecure classical channel between each client and Bob, and one-way quantum channels between each client and Bob. If Alice expects quantum outputs, a two-way quantum channel between Alice and Bob is needed.

The BOQC protocol is provided in protocol 1, in a clear-cut style with explicit adaptive measurements. For generality, protocol 1 admits the case in which Alice requires quantum input and quantum output; integrating classical inputs and outputs to the protocol is straightforward, and it is also discussed. Note that symbol $\oplus$ is defined as modulo 2 addition.

Every usage of a resource in $\mathcal{R}$—interactions with communication channels—is depicted in figure 2, where a circled number corresponds to the indicated part in protocol 1. Figure 2 shows that Alice and Oscar alternately take over the computation (transmitting qubits and giving commands with measurement angles) without communicating with each other, apart from sharing a key in the beginning.

In protocol 1 Bob receives input from Alice as $\{(\mathcal{G}, I, O), f, \phi, \rho_{\mathcal{A}}^{\text{in}}\}$ and input from Oscar as $\{\psi\}$, where $\mathcal{G}$ is the total graph with flow f, I is a set of input nodes that will be assigned with input state $\rho_{\mathcal{A}}^{\text{in}}$, O is a set of output nodes, ϕ is a set of measurement angles of Alice's nodes, and ψ is a set of measurement angles of Oscar's nodes. Prior to the protocol, we assume pre-protocol steps in definition 3 have been successfully done.

Protocol 1 comprises five steps: ⓪ pre-preparation, ❶ state preparation, ❷ graph state formation, ❸ classical interaction and measurements, and ❹ output transmission and correction. The interactions of

Protocol 1. BOQC ($\pi_{\text{boqc}} = \{\pi_{\mathcal{A}}, \pi_{\mathcal{B}}, \pi_{\mathcal{O}}\}$).

```
Alice's input: ${(\mathcal{G}, I, O), f, \phi, \rho_{\mathcal{A}}^{\text{in}}}$    ▷ $\tilde{I} = I$ and $\tilde{O} = O$
Oscar's input: ${\psi}$
Alice's output for an honest Bob: $\rho_{\mathcal{A}}^{\text{out}} = \mathcal{E}(\rho_{\mathcal{A}}^{\text{in}})$
  Assumptions and conventions:
    (I) Alice ($\mathcal{A}$) and Oscar ($\mathcal{O}$) have performed pre-protocol steps in definition 3; Bob knows $\{(\mathcal{G}, \tilde{I}, \tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$. Here, we set
        $\tilde{I} = I$ and $\tilde{O} = O$.
        Recall $\tilde{O} \cap V_{\mathcal{O}} = \varnothing$ (quantum outputs are held by Alice) and $\Omega = \{\frac{\pi k}{2^{b-1}}\}_{0 \leqslant k < 2^b}$.
    (II) $s_{\mathrm{inv}f(i)} = 0, \forall i \in I$ and $t_i = 0, \forall i \in I^{\mathrm{c}}$.
    (III) $\mathrm{inv}f(i) \equiv f^{-1}(i)$, $z(i) := \bigoplus_{k \prec i, i \in N_{\mathcal{G}}(f(k))} s_k$, and $t(i) := \bigoplus_{k \in I, i \in N_{\mathcal{G}}(k)} t_k$.
⓿ Pre-preparation
Alice and Oscar receive keys $r, t$ via a secure key channel, where $r_i \in \{0,1\}, i \in O^{\mathrm{c}}$ and $t_j \in \{0,1\}, j \in I$.
❶ State preparation
for $i \in V \backslash O$ following partial ordering $\succ$ do[16]    ▷ It may follow ordering $>$
  if $i \in V_{\mathcal{A}}$ then
      Alice chooses $\alpha_i \in \Omega$ at random.
      if $i \in I$ then
          Alice applies $Z_i(\alpha_i) X_i^{t_i}$ to $\mathrm{tr}_{I \backslash i}[\rho_{\mathcal{A}}^{\text{in}}]$ and sends it to Bob.[17]
          Alice updates angles:
              $\phi_i = (-1)^{t_i} \phi_i$
              $\phi_j = \phi_j + t_i \pi, \ \forall j \in N_{\mathcal{G}}(i) \cap V_{\mathcal{A}}$.
          Oscar updates angles $\psi_j = \psi_j + t_i \pi, \ \forall j \in N_{\mathcal{G}}(i) \cap V_{\mathcal{O}}$.
      else
          Alice prepares $|+_{\alpha_i}\rangle_i$ and sends it to Bob.
      end if
  else if $i \in V_{\mathcal{O}}$ then
      Oscar prepares $|+_{\beta_i}\rangle_i$ and sends it to Bob, where $\beta_i \in \Omega$ is chosen at random.
  end if
end for
For all $i \in O$, Bob prepares $|+\rangle_i$.
❷ Graph state formation
Bob applies entangling operator $E_{\mathcal{G}}$ defined in equation (4).
❸ Classical interaction and measurement
for $i \in V \backslash O$ which follows partial ordering $\succ$ do
  if $i \in V_{\mathcal{A}}$ then
      Alice computes $\phi_i' = (-1)^{s_{invf(i)}} \phi_i + z(i)\pi$.
      Alice computes $\delta_i := \phi_i' + \pi r_i + \alpha_i$, and sends Bob $\delta_i$.
  else if $i \in V_{\mathcal{O}}$ then
      Oscar computes $\psi_i' = (-1)^{s_{invf(i)}} \psi_i + z(i)\pi$.
      Oscar computes $\delta_i := \psi_i' + \pi r_i + \beta_i$, and sends Bob $\delta_i$.
  end if
  Bob measures qubit $i$ in $|\pm_{\delta_i}\rangle$ basis, then sends Alice and Oscar the outcome $\tilde{s}_i$.
  Alice and Oscar set $s_i = \tilde{s}_i \oplus r_i$.
end for
❹ Output transmission and correction
Bob sends Alice output qubits $\rho_{\mathcal{B}}^{\text{out}}$ (all qubits $i \in O$).
Alice corrects the final output $P(\rho_{\mathcal{B}}^{\text{out}}) =: \rho_{\mathcal{A}}^{\text{out}}$, where $P \equiv \otimes_{i \in O} X_i^{s_{invf(i)} + t_i} Z_i^{z(i)+t(i)}$.
```

[16] It means the loop goes through all nodes in $V \backslash O$ following a total ordering $>$ that is consistent with $\succ$; the total ordering $>$ is implicit there.

[17] Operator tr_j is a partial trace, where subsystem j is traced-out; thus $\mathrm{tr}_{I \backslash i}[\rho_{\mathcal{A}}^{\text{in}}]$ indicates that subsystem i remains, where $i \in I$.

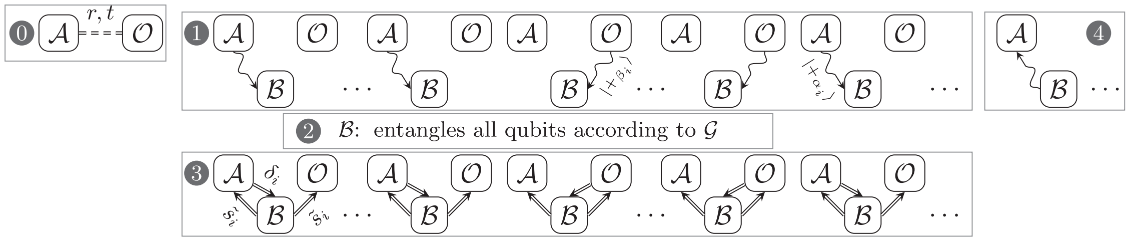
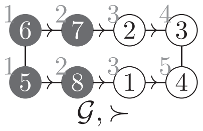


Figure 2. Interactions within BOQC scheme; these also describe the access to the inside interface of resource $\mathcal{R}$. Initials $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{O}$ indicate the interface of Alice, Bob, and Oscar respectively. The double dashed line indicates a secure key channel; the double lines indicate classical channels, with arrow indicating direction of transmission; the wavy lines indicate quantum channels. The circled numbers ⓪–④ correspond to the steps shown in protocol 1.

players via a channel at each step are depicted in figure 2. The protocol is initiated by establishing a symmetric key between Alice and Oscar via a secure key channel—step ⓪. This step allows them to privately delegate their joint computation: in particular, it allows them to know the actual measurement

Table 1. Correction terms in lemma 1 with graph $(\mathcal{G}, I, O)$ from example 1. In pattern $\mathfrak{P}_1$, upon measuring i, X-correction goes to node $f(i)$ and Z-corrections go to nodes $N_{\mathcal{G}}(f(i))\backslash\{i\}$. In pattern $\mathfrak{P}_2$, before measuring i, X-correction is performed based on measurement inv$f(i)$ and Z-correction based on measurements $z(i)$.

Node i	$f(i)$	$N_{\mathcal{G}}(f(i))$	$N_{\mathcal{G}}(f(i)) \setminus \{i\}$	inv$f(i)$	$z(i)$
1	4	$\{1, 3\}$	$\{3\}$	8	s_5
2	3	$\{2, 4\}$	$\{4\}$	7	s_6
3	—	—	—	2	$s_1 \oplus s_7$
4	—	—	—	1	$s_2 \oplus s_8$
5	8	$\{1, 5\}$	$\{1\}$	—	—
6	7	$\{2, 6\}$	$\{2\}$	—	—
7	2	$\{3, 7\}$	$\{3\}$	6	—
8	1	$\{4, 8\}$	$\{4\}$	5	—



outcomes so that their adaptive measurements can be computed independently. Steps ①–④ comprises the delegated computation process: ① the clients send Bob the encrypted qubits, ② Bob entangles the received qubits, ③ the clients communicate to Bob the measurement angles, and ④ Bob sends the output qubits if necessary, i.e. quantum outputs.

The following lemma (lemma 1) shows that, for any computation, the obtained pattern in protocol 1 without randomness is identical to the obtained pattern in the 1WQC scheme. The proof of lemma 1 is available in appendix A.

Lemma 1. *[5] Suppose the open graph state $(\mathcal{G}, I, O)$ has flow $(f, \succ)$, then the following patterns $\mathfrak{P}_1, \mathfrak{P}_2$ are identical $\forall \phi_i$:*

$$\mathfrak{P}_1 := \overset{\succ}{\prod_{i \in O^c}} (X^{s_i}_{f(i)} \prod_{k \in N_{\mathcal{G}}(f(i)) \backslash \{i\}} Z^{s_i}_k M^{\phi_i}_i) E_{\mathcal{G}} N^0_{I^c} \tag{11}$$

$$\mathfrak{P}_2 := \bigotimes_{j \in O} X^{s_{invf(j)}}_j Z^{z(j)}_j \overset{\succ}{\prod_{i \in O^c}} (M^{\phi_i}_i X^{s_{invf(i)}}_i Z^{z(i)}_i) E_{\mathcal{G}} N^0_{I^c}, \tag{12}$$

where $z(i) := \bigoplus_{k \prec i, i \in N_{\mathcal{G}}(f(k))} s_k$, inv$f(i) \equiv f^{-1}(i)$, *and* inv$f(i) = 0$ *for all* $i \in I$.

Patterns $\mathfrak{P}_1$ and $\mathfrak{P}_2$ give different points of view in writing the corrections: pattern $\mathfrak{P}_1$ shows the obtained correction from measuring i, while pattern $\mathfrak{P}_2$ shows the corrections that can be done before measuring i.[18] Those points of view were first introduced in [36]. See table 1 for an illustration.

Using lemma 1, we obtain theorem 1, which states that an algorithm run within the BOQC implements the same map as when the algorithm is run directly within the 1WQC, without requiring Alice and Oscar to share any of their secrets. The proof of theorem 2 is available in appendix A.

Theorem 2. *[5] The BOQC protocol π_{boqc} defined in protocol 1 delegates a computation with the isometry defined in theorem 1 for the same computation, without requiring Alice and Oscar to communicate their computations to each other.*

3.3. The quantum power

Like UBQC, BOQC is a protocol that involves a clear separation of quantum power between client and server, where a client is only capable of producing and transmitting quantum states of the form $|+_\theta\rangle$, while the server is presumed to posses unlimited quantum power. While this formulation is exactly true for classical input and output ($\tilde{I} = \tilde{O} = \varnothing$), a client needs higher quantum power for quantum input and output, i.e. $\tilde{I} \neq \varnothing$ or $\tilde{O} \neq \varnothing$. Table 2 summarizes the minimal quantum power requirement of running BOQC protocols with various input–output types.

Protocol 1 admits the general case, i.e. $\tilde{I} = I$ and $\tilde{O} = O$. As shown in table 2, it has the highest requirement among all the cases. A few adjustments from protocol 1 are needed if $\tilde{I} \subset I$ or $\tilde{O} \subset O$.

Given an entirely classical input case $\tilde{I} = \varnothing$, e.g. Alice's input is a binary string $c = c_1 c_2 \ldots c_n$, where $c_i \in \{0, 1\}$, lines 6–8 in protocol 1 turns into a single line:

"Alice prepares $|+_{\alpha_i} + \pi c_i\rangle$";

[18] The significance of this point of view is in our BOQCo protocol, to be shown later in section 4. In this point of view, the causality of BOQCo becomes apparent.

Table 2. [5] Minimal quantum power requirements of Alice and Bob to run a BOQC protocol. Oscar's requirement remains (C1) for all cases. Initial 'c' indicates 'entirely classical' and initial 'q' indicates 'entirely quantum'. Resources $\mathcal{K}, \mathcal{C}, \mathcal{Q}$, and $\mathcal{Q}2$ are respectively signifying a secure key channel between Alice and Oscar, an insecure classical channel between each client and server, a one-way quantum channel (each client to server), and a two-way quantum channel between Alice and Bob. Other notations follow the notations in protocol 1.

I	O	**Client** (Alice)	**Server** (Bob)	**Resource**
c	c	**(C1)** creates $\lvert +_\theta \rangle$, then transmits it to Bob	**(S1)** receives $\lvert +_\theta \rangle$, performs CPHASE gates, and measures qubits in the *xy*-plane bases	$\mathcal{K}, \mathcal{C}, \mathcal{Q}$
c	q	**(C1)** and **(C2)** receives $\rho_{\mathcal{B}}^{\text{out}}$ from Bob, then performs Pauli correction on it	**(S1)** and **(S2)** create $\lvert + \rangle\ \forall\ i \in O$ and sends Alice the final outputs $\rho_{\mathcal{B}}^{\text{out}}$	$\mathcal{K}, \mathcal{C}, \mathcal{Q}, \mathcal{Q}2$
q	c	**(C1)** and **(C3)** creates quantum input $\rho_{\mathcal{A}}^{\text{in}}$, performs a quantum one-time pad (e.g., line 6 in protocol 1) then transmits it to Bob	**(S1)** and **(S3)** receives input states (arbitrary) from Alice	$\mathcal{K}, \mathcal{C}, \mathcal{Q}$
q	q	**(C1)–(C3)**	**(S1)–(S3)**	$\mathcal{K}, \mathcal{C}, \mathcal{Q}, \mathcal{Q}2$

recall that $Z \lvert + \rangle = \lvert +_\pi \rangle$. Since the quantum one-time pad is unnecessary, the random string t is omitted (or setting $t_i = 0, \forall i$). Thus, requirements **(C3)** and **(S3)** vanish.

Now, when $\tilde{O} = \varnothing$ (entirely classical output), measurements will be performed on all qubits $i \in V$ and Bob does not need to prepare state $\lvert + \rangle$ himself nor to send Alice the final outcome $\rho_{\mathcal{B}}^{\text{out}}$. The first removes requirement **(C2)**, which replaces the loop in line 18 with

$$\text{“}\mathbf{for}\ i \in V \ \text{which follows partial ordering} \succ \text{.”}$$

The latter eliminates requirement **(S2)**, which removes line 16 and removes step ④ entirely.

Therefore, the lowest quantum power demand occurs for the entirely classical input and output case, i.e. $\tilde{I} = \tilde{O} = \varnothing$, requiring only **(C1)** and **(S1)**. We provide an explicit BOQC protocol for entirely classical input and output in protocol 3, appendix B.

4. BOQC optimized: BOQC on solid-state qubits

The 1WQC—as well as BOQC—can efficiently perform computations on memoryless quantum computers, such as photonic qubits, which is shown by successful experimental demonstrations on linear optics quantum computers: one- and two-qubit gates [20], two-qubit Grover's algorithm [22], Deutsch's algorithm [23], blind quantum computing [24], and verification of quantum computations [37]. However, extending the experiments to perform more complex computations is very hard since the individual qubit control is tricky in memoryless qubit systems. Individual-qubit control on solid-state systems are more promising, but scalability remains challenging. This problem motivates us to come up with an 'optimized' version of the BOQC, which we will call *blind oracular quantum computation-optimized* (BOQCo).

BOQCo allows us to perform BOQC algorithms with a minimal number of physical (solid-state) qubits. BOQCo is runnable on an appropriate platform[19] whose qubits possess permanence and can be rapidly re-initialized.

We prepare the graph in parts to minimize the number of physical qubits; the qubits are initialized only when needed. We call such a strategy *lazy 1WQC* in which the server only needs to prepare the closed neighborhood of the qubit about to be measured.[20] Note that such a graph preparation has been introduced in [27] but restricted to graphs with the same number of inputs and outputs ($|I| = |O|$); here, we extend it to arbitrary graphs with flow.

4.1. Lazy 1WQC computation

The lazy 1WQC is a 1WQC-computation scheme that allows one to prepare parts of the graph state as needed such that the number of physical qubits is minimal. The lazy 1WQC scheme is shown in algorithm 1 with input

$$\{(\mathcal{G}^{>}, I, O), f, \phi, \rho^{\text{in}}\}, \tag{13}$$

where $(\mathcal{G}^{>}, I, O)$ is an open graph with total ordering $>$ that follows the flow f, ϕ is the set of measurement angles, and ρ^{in} is the state assigned to I. Note that to describe a lazy 1WQC, one needs an additional parameter, total ordering $>$, compared with the description of a 1WQC computation in equation (8). That

[19] These are a quantum-network platform in which Bob owns a solid-state quantum system, e.g. NV-center and trapped-ions.

[20] The 'lazy' name is inspired from a computational programming paradigm called *lazy evaluation*. Lazy evaluation means that the evaluation of an expression is delayed until the value is needed [38].

Algorithm 1. Lazy 1WQC computation.

```
Input: $\{(\mathcal{G}^{>}, I, O), f, \phi, \rho^{\text{in}}\}$
Output: $\mathcal{E}(\rho^{\text{in}})$                                   ▷ See theorem 3
Conventions:
  (I) Partial order $\succ$ is induced by flow $f$.
  (II) $z(i) := \bigoplus_{k<i, i\in N_{\mathcal{G}}(f(k))} s_k$, $\text{inv}f(i) \equiv f^{-1}(i)$.
  (III) $s_{\text{inv}f(i)} = 0$ for all $i \in I$.
1: Assign $\rho^{\text{in}}$ to the input nodes $I$.
2: for $i \in V$ with ordering $>$ do
3:     for $k \in A(i)$ do                                        ▷ See equation (14)
4:         assign state $|+\rangle$ to node $k$
5:     end for
6:     Apply entangling operations $E^{>}_{iN_{\mathcal{G}}(i)}$.
7:     if $i \in O^{\text{c}}$ then
8:         $\phi_i' := (-1)^{s_{invf(i)}}\phi_i + z(i)\pi$
9:         Measure $i$ in basis $\left|\pm_{\phi_i'}\right\rangle$ and obtain measurement outcome $s_i$.
10:     else
11:         Correct output $i$ applying $X_i^{s_{invf(i)}} Z_i^{z(i)}$.
12:     end if
13: end for
```

is, the user must settle on a total ordering $>$ beforehand.[21] Any two valid total orderings will result in the same computation and have same requirement on the number of physical qubits, but they might require different coherence time for the qubits, as we have illustrated on our work on the Grover algorithm [26].

Algorithm 1 shows the general case, where the input and output are quantum.[22] If the input is classical, one can trivially encode as a quantum state ρ^{in} implemented as the following. Given the input state as a bit string c, one can implement by setting the input nodes as $|+_{c_i\pi}\rangle$ for all $i \in I$. If the output is classical, all nodes in O will be measured, i.e. the loop in line 2 is replaced with

$$\textbf{for } i \in V \text{ withordering} > \textbf{do},$$

and the algorithm is terminated after line 13. Consider example 2 to illustrate running a lazy 1WQC computation.

Example 2. Given a computation $\{(\mathcal{G}^{>}, I, O), f, \phi, \rho^{\text{in}}\}$, where $\mathcal{G} = (V, E)$ for $V = \{1, 2, 3, 4, 5, 6, 7\}$ and $E = \{(1, 3), (2, 3), (2, 4), (4, 6), (4, 5), (3, 5), (3, 7)\}$; state ρ^{in} is assigned to input nodes $I = \{1, 2\}$, output nodes $O = \{5, 6, 7\}$, and quantum inputs–outputs are expected. Running this computation in the lazy 1WQC scheme is illustrated in figure 3.

The allocation of fresh physical qubits, which corresponds to grey nodes in example 2, occurs in algorithm 1 in lines 3–5; these qubits are then initialized with state $|+\rangle$. We denote such a set of nodes as

$$A(i) := N_{\mathcal{G}}[i] \backslash (I \cup_{j<i} N_{\mathcal{G}}[j]), \tag{14}$$

which is a closed neighborhood, excluding the ones that have been assigned before. We assume that the input nodes $i \in I$ are assigned with the desired quantum input ρ^{in} before the scheme starts. As it is obvious that $A(i) \subseteq V$, the lazy 1WQC scheme does not construct the whole graph state at once.[23]

In the following, we establish the correctness of the lazy 1WQC. That is, we show that it results in the same computation as the standard 1WQC scheme. Then we derive bounds on the number of physical qubits needed.

First, note that we can write the resulting pattern of algorithm 1 by consecutively placing the initialization, entanglement, Pauli-correction, and measurement commands:

$$\mathfrak{P}_{lazy} = \bigotimes_{j\in O} (X_j^{invf(j)} Z_j^{z(j)}) E_O \overset{>}{\prod_{i\in O^c}} M_i^{\phi_i} X_i^{invf(i)} Z_i^{z(i)} E^{>}_{iN_{\mathcal{G}}(i)} N^0_{A(i)}, \tag{15}$$

where $z(i) := \oplus_{k<i, i\in N_{\mathcal{G}}(f(k))} s_k$ and $\text{inv}f(i) = 0$ for all $i \in I$. Note that the specification of Pauli operators before measurement follows equation (5).

Theorem 3 formally states the correctness of the lazy 1WQC, with the help of lemmas 1 to 3. The proof of each lemma is available in appendix A.

[21] Compare to BOQC or 1WQC, where one can define the total ordering $>$ during computation.

[22] This is comparable to $\tilde{I} = I$ and $\tilde{O} = O$ in the BOQC.

[23] There are certainly some cases where lazy 1WQC constructs the whole graph, e.g. graphs that are fully connected and star graphs.

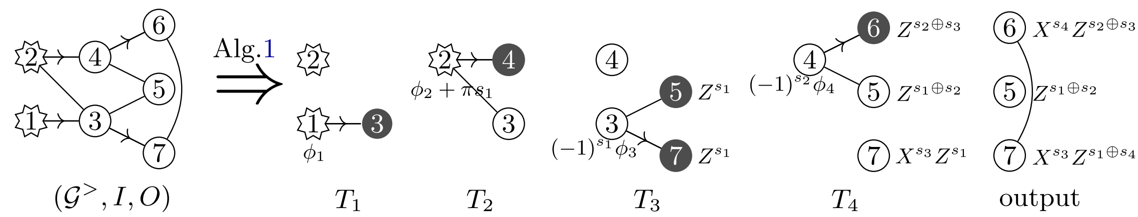


Figure 3. Running algorithm 1 with an open graph $(\mathcal{G}^{>}, I, O)$. The arrows indicate the flow of $\mathcal{G}$, the node number indicates total order $>$, T_i indicates the time step when measuring i, s_i signifies the measurement outcome of measuring i, the grey nodes are fresh qubits initialized with state $|+\rangle$ before measuring i, and the Pauli correction is shown on the corresponding node. The highest number of physical qubits requirements is 4, occurring at time-steps T_3 and T_4.

Lemma 2. *[5] Suppose the open graph state $(\mathcal{G}^{>}, I, O)$ has flow $(f, \succ)$ and a proper total order $>$, then $A(i)$ contains at least $f(i)$ for all $i \in O^c$ and $A(i) \cap A(j) = \varnothing$ for all $i \neq j$.*

Lemma 3. *[5] Suppose the open graph state $(\mathcal{G}^{>}, I, O)$ has flow $(f, \succ)$ and a proper total order $>$, then $\cup_{i\in V} A(i) = I^c$.*

Lemma 4. *[5] Suppose an open graph state $(\mathcal{G}^{>}, I, O)$ has flow $(f, \succ)$ and a proper total order $>$, then $\prod_{i\in V} E^{>}_{iN_{\mathcal{G}}(i)} = E_{\mathcal{G}}$, where $E^{>}_{iN_{\mathcal{G}}(i)} := \prod_{k\in N_{\mathcal{G}}(i), k>i} E_{ik}$.*

Lemma 2 shows that in every time-step, one must assign at least one fresh qubit. Followed by lemma 3 which shows that every non-input qubit is assigned once. Lemma 4 proves that we recover the whole graph. Note that, these lemmas (lemmas 2 and 3) consider the general case—quantum input and output—in which the input nodes are initialized beforehand. Finally, we prove the correctness of lazy 1WQC in the following theorem.

Theorem 3. *[5] The lazy 1WQC scheme and the 1WQC scheme implement the same map, they produce the same output for the same input.*

Proof. The following proof is similar to the one in [5]. Let $\{(\mathcal{G}^{>}, I, O), f, \phi, \rho^{\text{in}}\} := \mathscr{I}$ be the input of the lazy scheme (algorithm 1), where $>$ is consistent with the flow $(f, \succ)$, i.e. the flow of the open graph $(\mathcal{G}, I, O)$. Since $>$ is consistent with $\succ$, $\mathscr{I}$ is a valid input of the 1WQC scheme.

One can prove the map equality by comparing the patterns, e.g. we will show that $\mathfrak{P}_{\text{1wqc}}$ in equation (9) can be reduced to pattern $\mathfrak{P}_{\text{lazy}}$ in equation (15). Using previous results, we have

$$\mathfrak{P}_{1wqc} = {}^{\succ}\overrightarrow{\prod_{i\in O^c}} \left(X^{s_i}_{f(i)} \prod_{k\in N_{\mathcal{G}}(f(i))\backslash\{i\}} Z^{s_i}_i M^{\phi_i}_i\right) E_{\mathcal{G}} N^0_{I^c} \tag{16}$$

$$\overset{\text{Lem. }1}{=} \prod_{j\in O} X_j^{s_{invf(j)}} Z_j^{z(j)} \, {}^{\succ}\overrightarrow{\prod_{i\in O^c}} \left(M^{\phi_i}_i X_i^{s_{invf(i)}} Z_i^{z(i)}\right) E_{\mathcal{G}} N^0_{I^c} \tag{17}$$

$$\overset{\text{Lems. }3,\ 4}{=} \prod_{j\in O} X_j^{s_{invf(j)}} Z_j^{z(j)} \, {}^{>}\overrightarrow{\prod_{i\in O^c}} \left(M^{\phi_i}_i X_i^{s_{invf(i)}} Z_i^{z(i)}\right) E_O \, {}^{>}\overrightarrow{\prod_{k\in O^c}} E^{>}_{kN_{\mathcal{G}}(k)} \, {}^{>}\overrightarrow{\prod_{l\in O^c}} N^0_{A(l)} \tag{18}$$

$$= \bigotimes_{j\in O} X_j^{s_{invf(j)}} Z_j^{z(j)} \, {}^{>}\overrightarrow{\prod_{i\in O^c}} \left(M^{\phi_i}_i X_i^{s_{invf(i)}} Z_i^{z(i)}\right) E_O \, {}^{>}\overrightarrow{\prod_{k\in O^c}} E^{>}_{kN_{\mathcal{G}}(k)} \, {}^{>}\overrightarrow{\prod_{l\in O^c}} N^0_{A(l)}. \tag{19}$$

Note that in the third equality, the partial ordering $\succ$ is replaced with the total ordering $>$; this is valid since $>$ is consistent with $\succ$.

Now we need to commute the entangling and preparation operators such that they are distributed according to the lazy scheme. First, consider any two nodes i and k, where $i, k \in O^c$ and $i < k$. The preparation and entangling operators are commuting, i.e.

$$E^{>}_{kN_{\mathcal{G}}(k)} E^{>}_{iN_{\mathcal{G}}(i)} N^0_{A(k)} N^0_{A(i)} = E^{>}_{kN_{\mathcal{G}}(k)} N^0_{A(k)} E^{>}_{iN_{\mathcal{G}}(i)} N^0_{A(i)}. \tag{20}$$

However, we need to check if the condition of causality holds: there is no entanglement operation involving qubits that are already measured or not yet created. Denote the set of edges $e(i) := \{(i, k) | k \in N_{\mathcal{G}}(i), k > i\}$, i.e. edges that correspond to entangling operations $E^{>}_{iN_{\mathcal{G}}(i)}$. By definition, set

$e(i)$ does not contain any node that has already been measured, namely any $k < i$. The nodes that correspond to edges $e(i)$ are

$$\{k \in N_{\mathcal{G}}[i] | k > i\} =: n(i). \tag{21}$$

By definition, $A(i)$ contains all nodes in $N_{\mathcal{G}}[i]$ minus the ones that have already been created $I \cup_{k<i} N_{\mathcal{G}}[k]$, thus $\forall\, x \in e(i), x \in \{(i,j) | j \in \cup_{j \leqslant i} A(j)\}$, which means every qubit connected by an edge in $e(i)$ is already initialized. Thus, there is no entanglement involving a qubit that has not yet been created. Therefore, equation (20) is causal.

Considering the measurement operator and the Pauli corrections, we need to commute the entangling operation through them, namely

$$M_i^{\phi} X_i Z_i E^{>}_{kN_{\mathcal{G}}(k)} N^0_{A(k)} E^{>}_{iN_{\mathcal{G}}(i)} N^0_{A(i)} = E^{>}_{kN_{\mathcal{G}}(k)} N^0_{A(k)} M_i^{\phi} X_i Z_i E^{>}_{iN_{\mathcal{G}}(i)} N^0_{A(i)}, \tag{22}$$

which is true if and only if $i \notin n(k)$. By definition of $n(k)$ (see equation (21)), $i < k$, thus, $i \notin n(k)$. Thus, we can distribute the entangling and preparation operators in equation (16) with respect to the ordering $>$ and obtain

$$\bigotimes_{j \in O} (X_j^{s_{invf(j)}} Z_j^{z(j)}) E_O \overset{>}{\prod_{i \in O^c}}\!\!\rightarrow M_i^{\phi_i} X_i^{s_{invf(i)}} Z_i^{z(i)} E^{>}_{iN_{\mathcal{G}}(i)} N^0_{A(i)} = \mathfrak{P}_{lazy}. \tag{23}$$

□

Since the minimal number of physical qubits is the pivot in the lazy 1WQC, it is natural to ask for a bound on the number of physical qubits for an arbitrary 1WQC computation. We provide conjecture 1 to immediately suggest an answer. The intuition behind conjecture 1 stems from a property of a graph with flow; namely, the number of nodes per layer cannot shrink. It is due to non-colliding correction nodes: two distinct nodes i, j cannot have the same X-correction node, $f(i) \neq f(j)$; otherwise, it violates the flow criteria (equation (7)).

Conjecture 1. *The number of physical qubits required to run lazy 1WQC in algorithm 1, regardless the input and output type—whether classical or quantum—is bounded by* $|O| + 1$.

4.2. The BOQCo protocol

Executing a 1WQC computation within the lazy scheme reduces the number of physical qubits vastly, bounded to $|O| + 1$ per conjecture 1. Here we integrate the lazy 1WQC paradigm into BOQC, producing a protocol that we call BOQCo (BOQC-optimized). BOQCo allows the server to prepare the graph state as needed, employing the minimal number of physical qubits while maintaining the blindness of the multi-party scheme.[24]

The BOQCo protocol is shown in protocol 2; in AC language, we address it as $\pi_{\text{boqco}} = \{\pi_{\mathcal{A}}, \pi_{\mathcal{B}}, \pi_{\mathcal{O}}\}$. The scheme employs a strategy identical to BOQC to provide blindness; it is apparent from the introduced randomness: r, t, α, β. The distinguishing feature of BOQCo lies in the distribution of the computation, which follows the lazy 1WQC.

In the BOQCo, we divide the process into three main steps: ⓪ pre-preparation that is identical to BOQC, ❶ computation part by part, which is BOQC (excluding pre-preparation) done one part at a time, and ❷ output transmission and correction that is also identical to BOQC. The input and output of the BOQCo protocol is identical to the BOQC, i.e. it receives input from Alice as $\{(\mathcal{G}, I, O), f, \phi, \rho^{\text{in}}_{\mathcal{A}}\}$ and input from Oscar as ψ.

We define the correctness of a BOQCo computation as if it is run in the 1WQC scheme. Formally, we state the correctness in theorem 4, where the proof is provided in appendix A.

Theorem 4. *[5] The BOQCo protocol* π_{boqco} *defined in protocol 2 delegates a computation with the isometry defined in theorem 1 for the same computation, without requiring Alice and Oscar to communicate their computation to each other.*

In terms of quantum power between clients and servers, BOQCo has requirements identical to those of BOQC as discussed in section 3.3. This is because BOQC and BOQCo differ only in the ordering among qubit transmissions, entanglements, and measurements.

[24] At this point, we claim that BOQC and BOQCo are blind; this statement is proven and exclusively discussed in section 5.

Protocol 2. BOQCo ($\pi_{\text{boqco}} = \{\pi_{\mathcal{A}}, \pi_{\mathcal{B}}, \pi_{\mathcal{O}}\}$).

```
Alice's input: $\{(\mathcal{G}, I, O), f, \phi, \rho_{\mathcal{A}}^{\text{in}}\}$    ▷ $\tilde{I} = I$ and $\tilde{O} = O$
Oscar's input: $\{\psi\}$
Alice's output for an honest Bob: $\rho_{\mathcal{A}}^{\text{out}} = \mathcal{E}(\rho_{\mathcal{A}}^{\text{in}})$
  Assumptions and conventions:
    (I) Alice ($\mathcal{A}$) and Oscar ($\mathcal{O}$) have performed pre-protocol steps in definition 3; Bob knows $\{(\mathcal{G}, \tilde{I}, \tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$. Here, we set
      $\tilde{I} = I$ and $\tilde{O} = O$. Recall $\tilde{O} \cap V_{\mathcal{O}} = \varnothing$
      (quantum outputs are held by Alice) and $\Omega = \{\frac{\pi k}{2^{b-1}}\}_{0 \leqslant k < 2^b}$.
    (II) $s_{\text{inv}f(i)} = 0, \forall i \in I$ and $t_i = 0, \forall i \in I^{\text{c}}$.
    (III) $\text{inv}f(i) \equiv f^{-1}(i)$, $z(i) := \bigoplus_{k \prec i, i \in N_{\mathcal{G}}(f(k))} s_k$, and $t(i) := \bigoplus_{k \in I, i \in N_{\mathcal{G}}(k)} t_k$.
⓿ Pre-preparation
Alice and Oscar receive keys $r, t$ via a secure key channel, where $r_i \in \{0,1\}, i \in O^{\text{c}}$ and    $t_j \in \{0,1\}, j \in I$.
❶ BOQC by parts
for $i \in V$ with ordering $>$ do
    for $k \in A(i) \cup I$ do    ▷ Equation (14), section 2.1
        if $k \in I$ then    ▷ Input qubit
            Alice applies $Z_k(\alpha_k) X_k^{t_k}$ to $\text{tr}_{I \setminus k}[\rho_{\mathcal{A}}^{\text{in}}]$ and sends it to Bob,
            $\alpha_k \in \Omega$ is chosen at random.
            Alice updates angles:
                $$\phi_k = (-1)^{t_k} \phi_k$$
                $$\phi_j = \phi_j + t_k \pi, \ \forall\, j \in N_{\mathcal{G}}(k) \cap V_{\mathcal{A}}.$$
            Oscar updates angles: $\psi_j = \psi_j + t_k \pi, \ \forall\, j \in N_{\mathcal{G}}(k) \cap V_{\mathcal{O}}$.
        else if $k \in O$ then    ▷ Output qubit
            Bob prepares $|+\rangle_k$.
        else    ▷ Auxiliary qubit
            if $k \in V_{\mathcal{A}}$ then
                Alice prepares $\left|+_{\alpha_k}\right\rangle_k$, sends it to Bob, $\alpha_k \in \Omega$ is chosen at random.
            else if $k \in V_{\mathcal{O}}$ then
                Oscar prepares $\left|+_{\beta_k}\right\rangle_k$ and send it to Bob, $\beta_k \in \Omega$ is chosen at random.
            end if
        end if
    end for
    Bob applies entangling operations $E_{iN_{\mathcal{G}}(i)}^{>}$.    ▷ Equation (4), section 2.1
    if $i \in O^{\text{c}}$ then
        if $i \in V_{\mathcal{A}}$ then
            Alice computes $\phi_i' = (-1)^{s_{invf(i)}} \phi_i + z(i)\pi$.
            Alice computes $\delta_i := \phi_i' + \pi r_i + \alpha_i$ and sends Bob $\delta_i$.
            Bob measures $i$ in $\left|\pm_{\delta_i}\right\rangle$ basis, sends Alice and Oscar the outcome $\tilde{s}_i$.
            Alice and Oscar set $s_i = \tilde{s}_i \oplus r_i$.
        else if $i \in V_{\mathcal{O}}$ then
            Oscar computes $\psi_i' = (-1)^{s_{invf(i)}} \psi_i + z(i)\pi$.
            Oscar computes $\delta_i := \psi_i' + \pi r_i + \beta_i$ and sends Bob $\delta_i$.
        end if
    else    ▷ Output qubit transmissions and corrections
        Bob sends Alice qubit $i$.
        Alice corrects qubit $i$ by applying $X_i^{s_{invf(i)} + t_i} Z_i^{z(i) + t(i)}$.
    end if
end for
```

5. Security analysis

This section elaborates on the security of BOQC and BOQCo using the AC framework [14]. We promise composable blindness for our protocols, which is the security level achieved in the UBQC. We investigate the consequences of our security definition; the issue of leaked information in the ideal resource puts limits on the permitted graph states of oracles.

Note that our proof construction here follows the notation of Portmann and Renner in reference [16]. In particular, the notion of *filter* introduced in [14] is removed in [16]. Filters cover some functionalities to the honest parties, while dishonest parties remove them accordingly. For instance, a filter is used to define UBQC blindness by Dunjko *et al* in [15]. One ideal resource with a filter captures functionalities in the ideal world for both honest and malicious Bob. Here, we offer an alternative definition of blindness without the notion of filter as introduced in [16]. Since the notion of filter is removed, we need to define two resources in the ideal world for honest and malicious Bob. Then, we simply compare our resources in the real world to both resources in the ideal world: we achieve *correctness* when Bob is honest and *security (blindness)* when

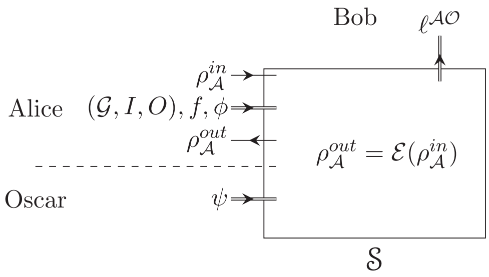

Bob
$\ell^{\mathcal{AO}}$
$\rho_{\mathcal{A}}^{in}$
Alice
$(\mathcal{G}, I, O), f, \phi$
$\rho_{\mathcal{A}}^{out}$
$\rho_{\mathcal{A}}^{out} = \mathcal{E}(\rho_{\mathcal{A}}^{in})$
Oscar
ψ
$\mathcal{S}$

Figure 4. [5] The ideal BOQC resource $\mathcal{S}$ in the absence of an adversary. The left side is the interface of clients, and the top side is the server interface. Single- and double- line arrows indicate quantum and classical information, respectively. The resource receives inputs from Alice: an open graph $(\mathcal{G}, I, O)$ with flow f, a quantum input $\rho_{\mathcal{A}}^{\text{in}}$, and a set of measurement angles $\phi_i \in \Omega$, where $\Omega = \{\frac{k\pi}{2^{b-1}}\}_{0\leqslant k<2^b}$ for an integer b. Also, it receives input $\psi_i \in \Omega$ from Oscar, and it does not take any inputs from Bob. Alice receives the final output $\rho_{\mathcal{A}}^{\text{out}} = \mathcal{E}(\rho_{\mathcal{A}}^{\text{in}})$, where $\mathcal{E}$ is the resulting superoperator of the algorithm, i.e. $\mathcal{E}$ is the isometry described in theorem 1. Some classical information $\ell^{\mathcal{AO}} = \{(\mathcal{G}, \tilde{I}, \tilde{O}), f, V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$ leaks on Bob's interface, which corresponds to public information in definition 3, and is necessary to set up the protocol.

Bob is dishonest.[25] In this fashion, the paradigm 'ideal-world real-world' becomes apparent. We hope that our explicit pedagogical-style proof is easy to follow and potentially attracts a broader audience to use AC.

Note that if we use a filter to define the security of BOQC/BOQCo, we can use some results of Dunjko *et al* [15]. We provide such a security proof in appendix C.

5.1. Correctness

We use definition 1 to state the correctness—also known as completeness—of BOQC and BOQCo protocols. Here, we prove that both protocols are perfectly correct ($\varepsilon = 0$) and realize an ideal resource $\mathcal{S}$, denoted as $\mathcal{R} \xrightarrow{\pi_{\text{boqc}},0} \mathcal{S}$ and $\mathcal{R} \xrightarrow{\pi_{\text{boqco}},0} \mathcal{S}$, where $\mathcal{R}$ is the real-world resource connected to the protocols and $\mathcal{S}$ is defined in figure 4. In both protocols, resource $\mathcal{R}$ comprises a secure key channel, quantum channels, and insecure classical channels.

The ideal resource $\mathcal{S}$ describes the BOQC system in the ideal world when Bob is honest. Resource $\mathcal{S}$ has an identical description of inputs and outputs with the BOQC protocol in protocol 1. Resource $\mathcal{S}$ also describes the BOQCo system in the ideal world. It also has an identical configuration of inputs and outputs with protocol 2. The leaked information $\ell^{\mathcal{AO}}$ is not apparent in the protocols (protocols 1 and 2); however this leakage is revealed in the proofs of the correctness theorems: theorem 5 for the BOQC and theorem 6 for the BOQCo.

Theorem 5. *[5] The BOQC protocol $\pi_{boqc} = (\pi_{\mathcal{A}}, \pi_{\mathcal{O}}, \pi_{\mathcal{B}})$ defined in protocol 1 is perfectly correct and emulates the ideal resource $\mathcal{S}$ defined in figure 4.*

Proof. The following proof is similar to the one in [5]. Protocol π_{boqc} is correct if it satisfies definition 1: $d(\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}\pi_{\mathcal{B}}, \mathcal{S}) = 0$, i.e. resources $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}\pi_{\mathcal{B}}$ and $\mathcal{S}$ must be perfectly indistinguishable, where d is a pseudo-metric with properties discussed in section 2.2. Which means, we show that the distinguishing advantage (defined in equation (10)) is zero. For that, we show that both resources have the same—or statistically the same—inputs and outputs.

First, we show that π_{boqc} and $\mathcal{S}$ have an identical description of inputs. As shown in protocol 1, the protocol receives inputs $\{(\mathcal{G}, I, O), f, \phi, \rho_{\mathcal{A}}^{\text{in}}\}$ from Alice, ψ from Oscar, and no honest input from Bob. These inputs are identical to the inputs of $\mathcal{S}$.

Second, $\mathcal{S}$ sends an output $\rho_{\mathcal{A}}^{\text{out}} = \mathcal{E}(\rho_{\mathcal{A}}^{\text{in}})$, where $\mathcal{E}$ is the superoperator with an isometry given in theorem 1; using theorem 2, BOQC also implements that isometry.

Finally we show that BOQC leaks the same information as $\mathcal{S}$, which is $\ell^{\mathcal{AO}}$. Bob receives information $\{(\mathcal{G}, \tilde{I}, \tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$ in the protocol, which is public information obtained from the pre-protocol steps defined in definition 3 before the protocol starts. The public information is identical to leak $\ell^{\mathcal{AO}}$. Bob is not curious in this setting, thus there is no additional information obtained beyond $\ell^{\mathcal{AO}}$. □

We note here that Bob does not erase the received information, as it is required to run the protocol and to provide the bill for his clients.[26]

[25] If we have more than one dishonest party who may arbitrarily cooperate, the security must capture every possible cooperation among these dishonest parties. Thus, the number of ideal resources will be the size of the power set of dishonest parties.

[26] Bob is not curious, but he needs to record some information for his clients to pay. For example, the dense Ω may cost more than the sparse Ω.

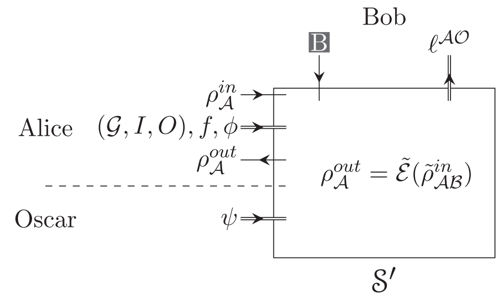

Bob
$\boxed{\mathbb{B}}$
$\ell^{\mathcal{AO}}$
$\rho^{in}_{\mathcal{A}}$
Alice
$(\mathcal{G}, I, O), f, \phi$
$\rho^{out}_{\mathcal{A}}$
$\rho^{out}_{\mathcal{A}} = \tilde{\mathcal{E}}(\tilde{\rho}^{in}_{\mathcal{AB}})$
Oscar
ψ
$\mathcal{S}'$

Figure 5. [5] Ideal resource $\mathcal{S}'$, the ideal-world resource when Bob is malicious. The input and output setting on the clients' interfaces is identical to $\mathcal{S}$—figure 4. Resource $\mathcal{S}'$ does not take any honest input from Bob, but he may entangle Alice's input with his state ($\tilde{\rho}^{\text{in}}_{\mathcal{AB}}$) and apply a superoperator of his choice ($\mathcal{E}$), obtaining $\tilde{\mathcal{E}}(\tilde{\rho}^{\text{in}}_{\mathcal{AB}})$. Variable $\boxed{\mathbb{B}}$ captures all Bob's dishonest inputs, which determines the final output $\rho^{\text{out}}_{\mathcal{A}}$. Input $\boxed{\mathbb{B}}$ is used to send the extension $\rho^{\text{in}}_{\mathcal{B}}$ to $\mathcal{S}'$, and to determine map $\tilde{\mathcal{E}}$.

The resource $\mathcal{S}$ models a general case in which Alice expects quantum input and quantum output, i.e. $\tilde{I} = I$ and $\tilde{O} = O$. If Alice needs only classical outputs, Bob will measure all output nodes O and her output density matrix $\rho^{\text{out}}_{\mathcal{A}}$ has diagonal form in the security model $\mathcal{S}$. The same applies to the classical input, e.g. for a bit string $c = c_n, \dots, c_2c_1$, we set $\rho^{\text{in}}_{\mathcal{A}} = \bigotimes_{i=1}^{n} |c_i\rangle\langle c_i|$ in the security model. Note that, per definition 3, Bob knows beforehand the input–output type, which is captured in the leaked information $\tilde{I}$ and $\tilde{O}$. For instance, Bob knows the input is entirely classical if $\tilde{I} = \varnothing$. Such information is necessary for Bob to prepare his channel.

As the counterpart of theorem 5, we prove the correctness of BOQCo protocol in theorem 6:

Theorem 6. *[5] The BOQCo protocol $\pi_{boqco} = (\pi_{\mathcal{A}}, \pi_{\mathcal{O}}, \pi_{\mathcal{B}})$, defined in protocol 2 is perfectly correct, and emulates ideal resource $\mathcal{S}$ defined in figure 4.*

Proof. The following proof is similar to the one in [5]. Using definition 1, correctness is achieved when $d(\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}\pi_{\mathcal{B}}, \mathcal{S}) = 0$. We prove this condition by reducing BOQCo to BOQC.

First, protocol 2 (BOQCo) has the same inputs as protocol 1 (BOQC). Applying theorem 4, BOQCo also results in the same computation as BOQC, thus, the same output. Secondly, BOQCo and BOQC differ only in the ordering among transmissions, entanglements, and measurements; thus, there is no additional leak introduced beyond $\ell^{\mathcal{AO}}$ (figure 4). In terms of correctness, BOQCo is reducible to BOQC. Finally, since BOQC is perfectly correct within the composable definitions, BOQCo is also perfectly correct within the composable definitions. □

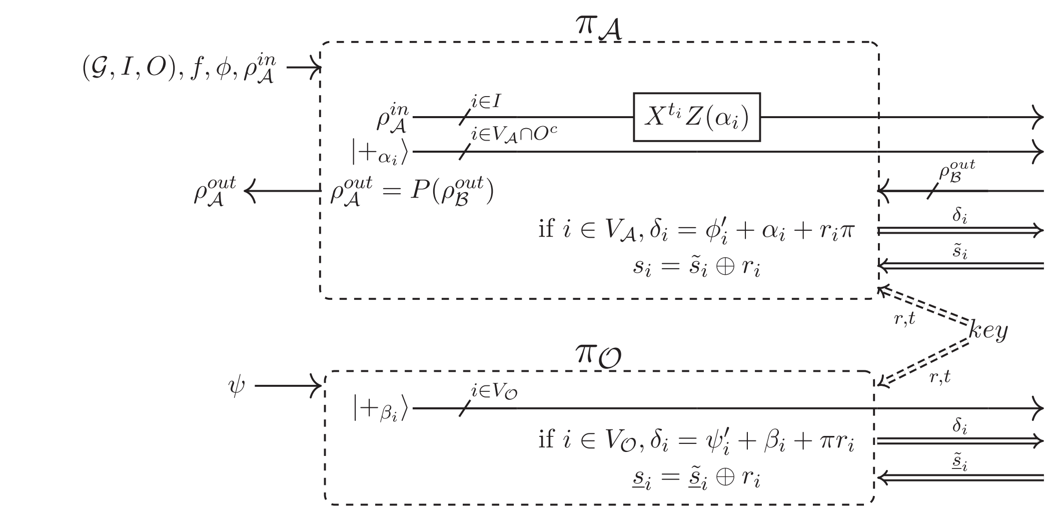
5.2. Blindness: when Bob is malicious

It remains to provide the security—also known as soundness—statement for BOQC and BOQCo protocols to achieve a composable secure definition. The security that is aimed for is *perfect blindness*, meaning the adversary (Bob) can learn nothing about the computation or the measurement outcomes. The same principles used for proving correctness apply also to prove blindness. We set the security model in the ideal world that captures the desired blindness, and then prove that our protocols that live in the real world are indistinguishable to the ideal-world model. However, while the correctness captures the system when everyone is honest, the blindness captures the situation when in the presence of an adversary, i.e. when Bob cheats.

In the presence of an adversary, our protocols realize the ideal resource $\mathcal{S}'$ that is defined in figure 5. Resource $\mathcal{S}'$ models the ideal system in the ideal world when Bob is malicious. On the clients' side, $\mathcal{S}'$ has the same input and output configuration as $\mathcal{S}$; however, in resource $\mathcal{S}'$, Bob provides dishonest inputs as he wishes, which is captured in $\boxed{\mathbb{B}}$. Nevertheless, both resources $\mathcal{S}$ and $\mathcal{S}'$ leak the same information $\ell^{\mathcal{AO}}$.

Given that $\mathcal{R}$ is the real-world resource used in our protocols, we need to achieve statements $\mathcal{R} \xrightarrow{\pi_{\text{boqc}},0} \mathcal{S}'$ and $\mathcal{R} \xrightarrow{\pi_{\text{boqco}},0} \mathcal{S}'$ where $\varepsilon = 0$ signifies perfect blindness. To prove that, we must satisfy definition 2, i.e. there exists a simulator $\sigma_{\mathcal{B}}$, such that $d(\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}, \mathcal{S}'\sigma_{\mathcal{B}}) = 0$. Recall that a simulator $\sigma_{\mathcal{B}}$ is needed to make $\mathcal{S}'$ and $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}$ become comparable, i.e. $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}$ has more inputs and outputs than $\mathcal{S}'$. See the proof of theorem 7 for explicit details. Theorem 7 provides the security statement of the BOQC protocol, whose relaxation $\mathcal{S}'\sigma_{\mathcal{B}}$ is defined in protocol 5 and in appendix B.

Theorem 7. *[5] The BOQC protocol with dishonest Bob $\pi_{boqc} = \{\pi_{\mathcal{A}}, \pi_{\mathcal{O}}\}$, defined in protocol 1, is perfectly blind and realizes the ideal resource $\mathcal{S}'$ defined in figure 5.*



$\pi_{\mathcal{A}}$
$(\mathcal{G}, I, O), f, \phi, \rho_{\mathcal{A}}^{in}$
$\rho_{\mathcal{A}}^{in}$
$i{\in}I$
$X^{t_i}Z(\alpha_i)$
$|+_{\alpha_i}\rangle$
$i{\in}V_{\mathcal{A}}{\cap}O^c$
$\rho_{\mathcal{A}}^{out}$
$\rho_{\mathcal{A}}^{out} = P(\rho_{\mathcal{B}}^{out})$
$\rho_{\mathcal{B}}^{out}$
if $i \in V_{\mathcal{A}}, \delta_i = \phi_i' + \alpha_i + r_i\pi$
δ_i
$s_i = \tilde{s}_i \oplus r_i$
$\tilde{s}_i$
$r{,}t$
key
$r{,}t$
$\pi_{\mathcal{O}}$
ψ
$|+_{\beta_i}\rangle$
$i{\in}V_{\mathcal{O}}$
if $i \in V_{\mathcal{O}}, \delta_i = \psi_i' + \beta_i + \pi r_i$
δ_i
$\underset{\sim}{s}_i = \underset{\sim}{\tilde{s}}_i \oplus r_i$
$\underset{\sim}{\tilde{s}}_i$

Figure 6. Pictorial representation of $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}$. Each variable corresponds to the one in protocol 1. We distinguish measurement outcomes on Alice's interface ($\tilde{s}_i$) and on Oscar's interface ($\underset{\sim}{\tilde{s}}_i$), because Bob is allowed to be dishonest, thus $\tilde{s}_i$ and $\underset{\sim}{\tilde{s}}_i$ may be different. The left-hand side denotes the outside interface of the converters. The right-hand side indicates the inside interface connected to resources: secure key channel (dashed double arrow), classical channel (single arrow), and quantum channel (double arrow). The strike-through line indicates multiple qubits.

Proof. The following proof is similar to the one in [5]. Applying definition 2, $\mathcal{R} \xrightarrow{\pi_{\mathrm{boqc}},0} \mathcal{S}'$ if there exists a simulator $\sigma_{\mathcal{B}}$ such that $d(\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}, \mathcal{S}'\sigma_{\mathcal{B}}) = 0$; thus, we must find a relaxation $\mathcal{S}'\sigma_{\mathcal{B}}$ that is perfectly indistinguishable from $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}$. Suppose the relaxation $\mathcal{S}'\sigma_{\mathcal{B}}$ is defined in protocol 5, then we proceed to prove that it is indistinguishable from $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}$.

To simplify the problem, we first reduce the protocol to a one-client protocol as follows. Consider a pictorial representation of π_{boqc} (in the absence of Bob) in figure 6; it clearly shows that the common information between Alice and Oscar is the keys s, t, shared via a secure key channel. Since it is known that a secure key channel guarantees secrecy and authenticity, Alice and Oscar obtain their keys without leaking any information to Bob. Therefore, we may think that Alice and Oscar have already shared the keys before the protocol starts. Thus, protocol $\pi_{\mathcal{A}}\pi_{\mathcal{O}}$ is reducible to a one-client protocol $\pi_{\mathcal{AO}}$, shown in figure 7, where the alternating part between Alice and Oscar is captured within functions $\theta(i)$, $s(i)$, and $\delta(i)$.

Relaxation $\mathcal{S}'\sigma_{\mathcal{B}}$ defined in protocol 5 is pictured in figure 8. Thus, we now can prove the statement $d(\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}, \mathcal{S}'\sigma_{\mathcal{B}}) = 0$ by showing that figures 7 and 8 are indistinguishable.

Notice that figures 6 and 7 clearly show identical inputs and outputs, indicated by the same configuration of arrows. Thus, it now remains for us to prove that the arrows with a circled letter are (statistically) the same.

Remark that we focus only on blindness without verifiability. Since verification is not involved, Alice does not care whether her computation is correct or not. That is, some information related to the computation can be arbitrary, such as $\rho_{\mathcal{B}}^{\mathrm{out}}$, $\tilde{s}_i$, $\underset{\sim}{\tilde{s}}_i$, and δ_i.

Consider the information sent by Bob: ⓒ $\rho_{\mathcal{B}}^{out}$, ⓔ $\tilde{s}_i$, and ⓕ $\underset{\sim}{\tilde{s}}_i$. First, in both figures, $\rho_{\mathcal{B}}^{\mathrm{out}}$ is an arbitrary state chosen by Bob (simulator). Second, in figure 7, $\tilde{s}_i$ and $\underset{\sim}{\tilde{s}}_i$ signify measurement outcomes seen by Bob, which is random information independent of the actual measurement outcomes: $s_i = \tilde{s}_i \oplus r_i$ and $\underset{\sim}{s}_i = \underset{\sim}{\tilde{s}}_i \oplus r_i$. The same is true in figure 8; $\tilde{s}_i, \underset{\sim}{\tilde{s}}_i$ are arbitrary information inputted to Bob's interface in $\mathcal{S}'$.

We now analyze the information received by Bob: ⓐ, ⓑ, and ⓓ δ_i. First, in both figures, δ_i are uniformly distributed random angles, independent of the actual measurement angles, ϕ_i', ψ_i'. Second, consider the information at ⓐ, namely at input node $i \in I$. In figure 7, ⓐ is an encrypted input state $X^{t_i}Z_i(\alpha_i)(\rho_i)$, where $\rho_i := \mathrm{tr}_{I\setminus i}[\rho_{\mathcal{A}}^{\mathrm{in}}]$. In figure 8, ⓐ is an uncorrected teleported state $X_i^{t_i}Z_i^{r_i}Z_i(\delta_i - \phi_i')(\rho_i) = X_i^{t_i}Z_i(\delta_i - \phi_i' - \pi r_i)(\rho_i) = X_i^{t_i}Z_i(\alpha_i)(\rho_i)$, which is identical to ⓐ in figure 7. Third, for ⓑ, consider $i \in V_{\mathcal{A}}$. In figure 7, ⓑ is $|+_{\alpha_i}\rangle$. In figure 8, ⓑ is a remote state preparation [39] of $Z^{r_i}\left|+_{\delta_i - \phi_i'}\right\rangle = \left|+_{\delta_i - \phi_i' - \pi r_i}\right\rangle = |+_{\alpha_i}\rangle$. This is also clearly true for $i \in V_{\mathcal{O}}$.

Finally, since both figures have the same inputs and outputs, it remains to show that the leak is given by $\ell^{\mathcal{AO}}$. The leak of information $\ell^{\mathcal{AO}}$ is inputted to the simulator from Bob's interface. The simulator does not use the leak to create any useful information. Now we need to prove that the simulator does not learn anything beyond $\ell^{\mathcal{AO}}$. For that, the information received by the simulator must be independent of the computation. First, δ_i is an arbitrary angle, thus independent of the computation. Second, ⓐ and ⓑ are completely mixed states because of the randomness α_i, β_i; thus, the simulator cannot guess α_i or β_i with

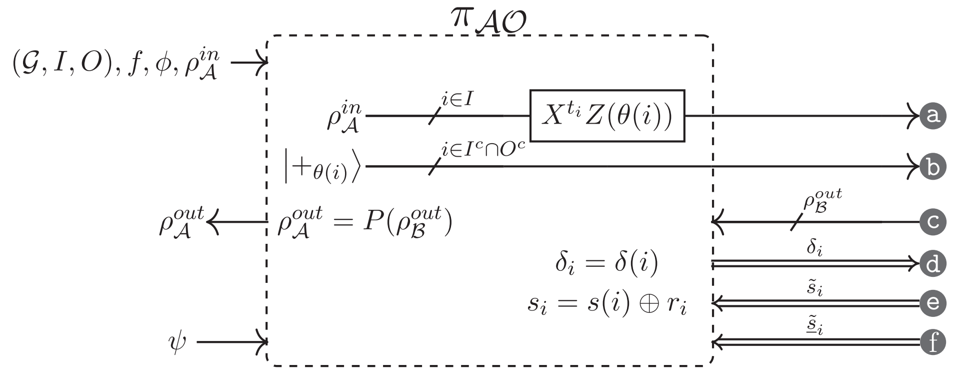

$\pi_{\mathcal{AO}}$
$(\mathcal{G}, I, O), f, \phi, \rho^{in}_{\mathcal{A}}$
$\rho^{in}_{\mathcal{A}}$
$i \in I$
$X^{t_i} Z(\theta(i))$
$|+_{\theta(i)}\rangle$
$i \in I^c \cap O^c$
$\rho^{out}_{\mathcal{A}}$
$\rho^{out}_{\mathcal{A}} = P(\rho^{out}_{\mathcal{B}})$
$\rho^{out}_{\mathcal{B}}$
$\delta_i = \delta(i)$
$s_i = s(i) \oplus r_i$
δ_i
$\tilde{s}_i$
$\underline{\tilde{s}}_i$
ψ
a
b
c
d
e
f

Figure 7. Pictorial representation of $\pi_{\mathcal{AO}}\mathcal{R}$. See figure 6 for notation; additionally, $\theta(i) = \begin{cases} \alpha_i, & \text{if } i \in V_{\mathcal{A}} \\ \beta_i, & \text{if } i \in V_{\mathcal{O}}, \end{cases}$ $s(i) = \begin{cases} \tilde{s}_i, & \text{if } i \in V_{\mathcal{A}} \\ \underline{\tilde{s}}_i, & \text{if } i \in V_{\mathcal{O}}, \end{cases}$ and $\delta(i) = \begin{cases} \phi'_i + \alpha_i + \pi r_i, & \text{if } i \in V_{\mathcal{A}} \\ \psi'_i + \beta_i + \pi r_i, & \text{if } i \in V_{\mathcal{O}}. \end{cases}$

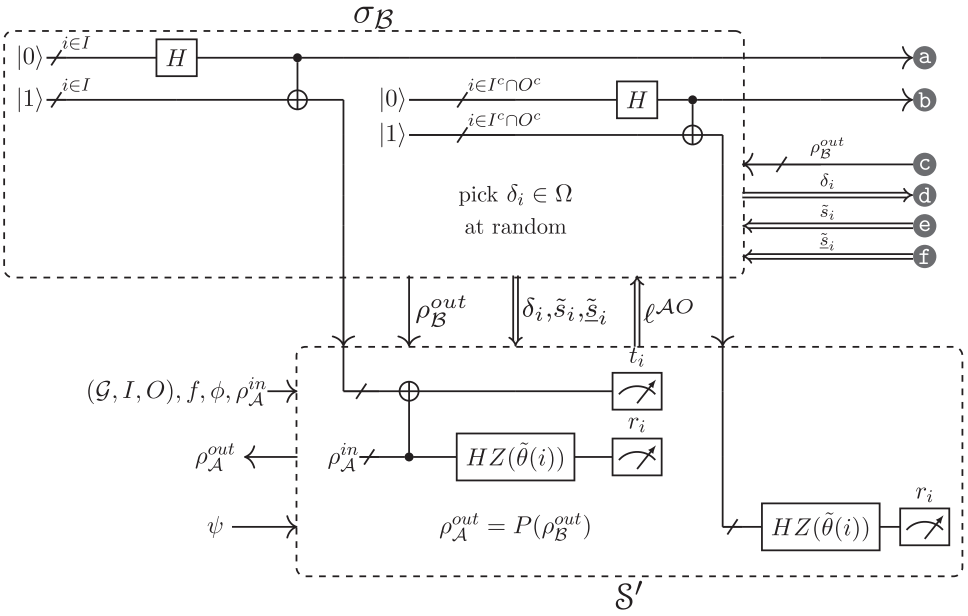

$\sigma_{\mathcal{B}}$
$|0\rangle$
$|1\rangle$
$i \in I$
H
$i \in I^c \cap O^c$
pick $\delta_i \in \Omega$
at random
$\rho^{out}_{\mathcal{B}}$
δ_i
$\tilde{s}_i$
$\underline{\tilde{s}}_i$
$\delta_i, \tilde{s}_i, \underline{\tilde{s}}_i$
$\ell^{\mathcal{AO}}$
t_i
r_i
$(\mathcal{G}, I, O), f, \phi, \rho^{in}_{\mathcal{A}}$
$\rho^{out}_{\mathcal{A}}$
$\rho^{in}_{\mathcal{A}}$
$HZ(\tilde{\theta}(i))$
ψ
$\rho^{out}_{\mathcal{A}} = P(\rho^{out}_{\mathcal{B}})$
$\mathcal{S}'$
a
b
c
d
e
f

Figure 8. Pictorial representation of $\mathcal{S}'\sigma_{\mathcal{B}}$ (protocol 5). The variables correspond to those defined in protocol 5. See figure 6 for the notation; additionally, $\tilde{\theta}(i) = \begin{cases} \delta_i - \phi'_i, & \text{if } i \in V_{\mathcal{A}} \\ \delta_i - \psi'_i, & \text{if } i \in V_{\mathcal{O}}. \end{cases}$

complete certainty. A curious Bob might correctly guess other information such as the flow f, however it gives him no advantage. Therefore, there is no more leak than $\ell^{\mathcal{AO}}$ throughout the protocol. This concludes the proof. □

Note that our strategy for constructing the simulator $\sigma_{\mathcal{B}}$ in protocol 5 comes to us from the work on composable security of delegated quantum computation by Dunjko *et al* in reference [15]. However, its usefulness in our case is only guaranteed by theorem 2.

Here we consider the general case when Alice's input and output are quantum, i.e. $\tilde{I} = I$ and $\tilde{O} = O$. The security is maintained as long as the input–output configuration admits the security model $\mathcal{S}'$. Thus, the security is maintained for $\tilde{I} \subset I$ and $\tilde{O} \subset O$ for the following reasons:

First, letting $\tilde{I} \subset I$, we denote Alice's quantum input $\rho^{\text{in}} \in \mathcal{H}_{\tilde{I}}$ and quantum input of the security model $\rho^{\text{in}}_{\mathcal{A}} \in \mathcal{H}_I$. Since $\mathcal{H}_I \subset \mathcal{H}_{\tilde{I}}$, she has another classical input in the form of a bit string c with length $|I \backslash \tilde{I}|$. As one may always encode classical information into qubits as we choose, we can set $\rho^{\text{in}}_{\mathcal{A}} = \rho^{\text{in}} \bigotimes_{i \in I \backslash I} |c_i\rangle\langle c_i|$, where $c_i \in \{0, 1\}$. Therefore, $\rho^{\text{in}}_{\mathcal{A}}$ is now entirely quantum as modeled by $\mathcal{S}'$. Finally, the same applies for $\tilde{O} \subset O$ in which classical output can be represented as a diagonal density matrix.

Now it remains to prove the blindness of BOQCo. The proof is rather straightforward because BOQC and BOQCo differ only in the ordering among qubit transmissions, entanglements, and measurements.

Theorem 8 provides the security statement of BOQCo, whose relaxation $\mathcal{S}'\sigma_{\mathcal{B}}$ is provided in protocol 6, in appendix B.

Theorem 8. *The BOQCo protocol with dishonest Bob $\pi'_{boqco} = (\pi_{\mathcal{A}}, \pi_{\mathcal{O}})$, defined in protocol 2 is perfectly blind, and realizes ideal resource $\mathcal{S}'$ defined in figure 5.*

Proof. By definition 2, statement $\mathcal{R} \xrightarrow{\pi_{\text{boqco}},0} \mathcal{S}'$ is achieved if there exists a simulator $\sigma_{\mathcal{B}}$ such that $d(\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}, \mathcal{S}'\sigma_{\mathcal{B}}) = 0$. Such a relaxation can be straightforwardly derived from protocol 5 by rearranging the process order, viz, qubit transmissions, entanglements, and measurements according to the lazy computation in algorithm 1 (the relaxation is shown explicitly in protocol 6, appendix B).

Since BOQCo and BOQC differ only by the process order, they have the same configuration of inputs and outputs. Therefore, $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}$ and $\mathcal{S}'\sigma_{\mathcal{B}}$ have pictorial representations as figures 7 and 8, respectively, where the figures are proven to be indistinguishable in theorem 7. The difference in the process order will not reveal any information about the computation because the ordering is determined before the protocol starts, arranged in pre-protocol steps (definition 3).

Since we obtain an indistinguishable relaxation to $\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}$ that has the same leak as BOQC, this concludes the proof. □

5.3. The BOQC-compatible graph states

The BOQC and the BOQCo protocols provide simple cooperation between Alice and Oscar to delegate their blind computations, allowing the malicious Bob to learn no more that public information $\ell^{\mathcal{AO}} := \{(\mathcal{G}, \tilde{I}, \tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$. The leaking information $\ell^{\mathcal{AO}}$ is insufficient to infer the computation, but there is a catch:

Consider an example of a real-life situation in which Oscar's company is well-known for storing massive confidential databases. Thus, Bob might reasonably make an priori assumption that they are running a quantum search algorithm. If the oracle's graph varies according to the request being made, Bob might infer some information about Alice's request, such as, a simple graph marks a zeros state in the Grover algorithm. Thus, Oscar's oracle graphs must remain the same for all requests; he is only allowed to vary only the measurement angles for different queries. We call such graphs, i.e. a class of graphs that runs a set of different requests, 'BOQC-compatible' graphs.

A BOQC-compatible graph is a standard oracle graph, determined by Oscar, for a class of requests. One can use a universal graph, e.g. a fixed-size brickwork state as introduced in reference [10]; however, the number of qubits increases rapidly with the circuit depth of the gate model representation. Another strategy is to optimize the graphs, constrained to the set of requests as done in reference [26], for two- and three-qubits exact Grover algorithms. The resulting graph is significantly more compact than the brickwork graph.

As the final remark of this paper, it is worth noting how BOQC and UBQC are related. As an extension of the UBQC, BOQC implements the same hiding protocol and achieves the same security level as the UBQC. This similarity shows clearly in section 5, in figure 6, where Alice and Oscar can be reduced to a client who has full knowledge of the computation. The difference between BOQC and UBQC can be summarized as follows. First, BOQC allows two clients with different computation knowledge, where UBQC has one client who has the full knowledge of the computation. Thus, BOQC works only for oracular quantum algorithms, whereas UBQC works for general computations. Second, UBQC restricts the graph state to the brickwork state, but BOQC allows arbitrary graph states from Alice and Oscar; moreover, Oscar needs BOQC-compatible graphs. Lifting the graph restriction will leak more classical information; however, it allows the clients to have compact graphs according to their wish or graphs optimized to the hardware architecture. Finally, we also extend the BOQC optimized to a particular solid-state hardware, introducing the BOQCo, which also inherits the same level of security as the UBQC.

6. Conclusion

Here, we have introduced BOQC, a secure client–server oracle computation scheme with three parties, that is an extension of the UBQC. Our scheme provides a means for a client, without a quantum computer and without the capability of performing the oracle evaluation, to securely cooperate with an oracle client to delegate the oracular quantum algorithm to a malicious quantum server. We formally prove that BOQC is blind within composable definitions using the AC framework. A drawback of our security definition is that it requires BOQC-compatible graphs as graph states. We extend BOQC to BOQCo (BOQC-optimized), which adapts the protocol for efficient use on a solid-state quantum network, which has the attributes that the server's qubits possess permanence and can be rapidly re-initialized. We provide explicit BOQC and BOQCo protocols that deal with both quantum and classical input and output. The BOQCo scheme allows the server to employ a minimal number of physical qubits; in separate work we show how the scheme enables the implementation of Grover and Simon algorithms in a realistic NV-center network. While our protocols promise blindness, we cannot tell whether the server is being malicious; thus, integrating verifiability into our protocol would be a desirable next step in this work.

Acknowledgment

We gratefully acknowledge discussions on abstract cryptography with C Portmann; we also thank him for proofreading the security proofs in this manuscript. CG thanks E Sovetkin for proofreading some lemmas and for his idea to employ measurement calculus in reasoning correctness. CG acknowledges support from a research collaboration with Huawei Technologies. DDV acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1 – 390534769.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Appendix A. Proof of lemmas and theorems

All proofs in this section are similar to the ones in reference [5].

The following proposition is required to proof lemma 1.

Proposition 1. *[5] Suppose i and j are two distinct vertices in an open graph $(\mathcal{G}, I, O)$ with flow $(f, \succ)$, then $f(i) \neq f(j)$.*

Proof. Let us assume the contrary, there exists three distinct nodes $i, j, k \in \mathcal{G}$ such that $f(i) = f(j) = k$. Given that flow f induces a partial ordering $\succ$; by the definition of flow, every criterion in it must be satisfied, i.e.

$$\begin{aligned}&(\text{F0})\ (k,i) \in E(\mathcal{G}) \text{ and } (k,j) \in E(\mathcal{G})\\&(\text{F1})\ k \succ i \text{ and } k \succ j\\&(\text{F2})\ \forall\ v \in N_{\mathcal{G}}(k)\backslash\{i\}, v \succ i \text{ and } \forall\ w \in N_{\mathcal{G}}(k)\backslash\{j\}, w \succ k.\end{aligned}$$

The first criterion of **(F2)** entails $j \in N_{\mathcal{G}}(k)\backslash\{i\}$ since $j \in N_{\mathcal{G}}(k)$ by criterion **(F0)**, which implies $j \succ i$; hence, **(F1)** imposes an ordering $k \succ j \succ i$. However, the second criterion of **(F2)** also implies that $i \in N_{\mathcal{G}}(k)\backslash\{j\}$, because $i \in N_{\mathcal{G}}(k)$—**(F0)**; thus, $i \succ j$, which leads to a contradiction. □

Lemma 1. *[5] Suppose the open graph state $(\mathcal{G}, I, O)$ has flow $(f, \succ)$, then the following patterns $\mathfrak{P}_1, \mathfrak{P}_2$ are identical $\forall\ \phi_i$:*

$$\mathfrak{P}_1 := \overset{\succ}{\prod_{i \in O^c}} (X_{f(i)}^{s_i} \prod_{k \in N_{\mathcal{G}}(f(i))\backslash\{i\}} Z_k^{s_i} M_i^{\phi_i}) E_{\mathcal{G}} N_{I^c}^0 \tag{11}$$

$$\mathfrak{P}_2 := \bigotimes_{j \in O} X_j^{s_{invf(j)}} Z_j^{z(j)} \overset{\succ}{\prod_{i \in O^c}} (M_i^{\phi_i} X_i^{s_{invf(i)}} Z_i^{z(i)}) E_{\mathcal{G}} N_{I^c}^0, \tag{12}$$

where $z(i) := \bigotimes_{k \prec i, i \in N_{\mathcal{G}}(f(k))} s_k$, $\text{inv}f(i) \equiv f^{-1}(i)$, *and* $\text{inv}f(i) = 0$ *for all* $i \in I$.

Proof of Lemma 1. Observe that both patterns are ordered in the following manner: (1) graph state preparation (qubit initialization and entangling operations), (2) adaptive measurements that follow the partial order $\succ$, and (3) applied Pauli corrections interspersed among the measurements. It is evident from inspection that the measurepatternnt operators $M_i^{\phi_i}$ occur in the same order in patterns $\mathfrak{P}_1$ and $\mathfrak{P}_2$. Thus, it only remains to prove that both patterns implement identical Pauli corrections.

Consider the X-corrections. After measuring i with output s_i, the X-correction propagates to qubit $f(i)$, that is, $X_{f(i)}^{s_i}$ will appear before measuring $f(i)$, as seen in $\mathfrak{P}_1$. Equivalently,[27] qubit $f(i)$ receives X-correction from measurement outcome of qubit i, where i is the preimage of $f(i)$. Thus, before measuring $f(i)$, the correction from measuring i must be present, namely $X_{f(i)}^{s_i}$. Let $f(i) =: j$, thus $X_{f(i)}^{s_i} = X_j^{s_{invf(j)}}$, which is the corresponding operator in $\mathfrak{P}_2$.

The same treatment applies to the Z-corrections; however, the corrections propagate to a subset of nodes instead of a single node. From pattern $\mathfrak{P}_1$, measuring i produces Z-corrections on nodes $\{N_{\mathcal{G}}(f(i))\backslash\{i\}\} =: K$, which are the neighbors of $f(i)$. From the perspective of $k \in K$, qubit k receives corrections from all measurement outcomes of i, where the set of neighbors of $f(i)$ contains k. From the flow condition, we know that $i \prec k$, meaning that i has already been measured. Thus, the Z-correction of qubit k must include all measurement outcomes from qubits $\{i \prec k | k \in N_{\mathcal{G}}(f(i))\}$. That is what we see in the superscript of Z in pattern $\mathfrak{P}_2$.

Since every node is corrected by one Pauli operation, the correction of the output can be represented as an operator acting on the output qubits. Note that the output qubits lie in the last layer of the graph: for all $k \notin O$, for all $i \in O$, we have $k \prec i$; thus, the corrections can appear at the very end of the pattern. □

Theorem 2. [5] *The BOQC protocol* π_{boqc} *defined in protocol* 1 *delegates a computation with the isometry defined in theorem* 1 *for the same computation, without requiring Alice and Oscar to communicate their computations to each other.*

Proof of Theorem 2. Given are Alice's input $\{(\mathcal{G}, I, O), \phi\}$ together with a flow f, and Oscar's input ψ. We can write the total computation as $\{(\mathcal{G}, I, O), \theta\}$, where $\theta = \phi \cup \psi$ contains measurement angles for all nodes $i \in O^{\mathrm{c}}$. We need to show that protocol 1 results in pattern $\mathfrak{P}_{boqc} := \succ\!\!\prod\limits_{i\in O^c}\!\!\succ (X_{f(i)}^{s_i} \prod\limits_{k\in N_{\mathcal{G}}(f(i))\backslash\{i\}} Z_k^{s_i+r_i} M_i^{\theta_i+\gamma_i+\pi r_i}) E_{\mathcal{G}} N_{I^c}^{\gamma}$.

First, we consider Alice and Oscar as one client called super-Alice. Omitting the randomness $r_i = \alpha_i = \beta_i = 0$,[28] running a computation with protocol 1 results in the pattern

$$\mathfrak{P}_{boqc} = \bigotimes_{j\in O} X_j^{s_{invf(j)}} Z_j^{z(j)} \succ\!\!\prod_{i\in O^c}\!\!\succ M_i^{\theta_i'} X_i^{s_{invf(i)}} Z_i^{z(i)} E_{\mathcal{G}} N_{I^c}^0 \overset{\text{Lem. 1}}{=} \mathfrak{P}_{1wqc},$$

where $z(i) := \sum_{k\prec i, i\in N_{\mathcal{G}}(f(k))} s_k$. Thus, with the absence of randomness, $\mathfrak{P}_{boqc} = \mathfrak{P}_{1wqc}$.

Second, consider γ, a random variable that contains α and β, thus $\gamma = \alpha \cup \beta$. Including γ and r in the computation,[29] we obtain pattern

$$\succ\!\!\prod\!\!\succ_{i\in O^c} (X_{f(i)}^{s_i} \prod\nolimits_{k\in N_{\mathcal{G}}(f(i))\backslash\{i\}} Z_k^{s_i} M_i^{\theta_i}) E_{\mathcal{G}} N_{I^c}^0 =: \mathfrak{P}_{1wqc}. \tag{24}$$

Thus, using equation (5) and commutation $E_{\mathcal{G}} N_i^{\gamma_i} = E_{\mathcal{G}} Z_i(\gamma_i) N_i^0 = Z_i(\gamma_i) E_{\mathcal{G}} N_i^0$, random variable γ_i is cancelled out.[30] The random angle πr_i is canceled out by flipping the measurement outcome, $s_i = s_i \oplus r_i$, which is expressed in the superscript of Z-corrections. Thus, equality $\mathfrak{P}_{boqc} = \mathfrak{P}_{1wqc}$ is maintained.

Third, we show that encryption of the quantum inputs $X_i^{t_i} Z_i(\alpha_i)(\rho^{\text{in}})$ is perfectly decrypted during the process. Random variable α_i is cancelled out in the same manner as removing γ above. The random variable t_i is canceled out by flipping the sign of the measurement angle[31] $M_i^{\theta} X_i \rho^{\text{in}} = M_i^{-\theta} \rho^{\text{in}}$. Thus, we conclude that ρ^{in} is intact.

Finally, we divide super-Alice into Alice and Oscar. Calculating corrected angles is separately done, since they know the real measurement outcomes $s_i = \tilde{s}_i \oplus r_i$. Canceling out the z-rotations $R_i^z(\alpha_i)$ or $R_i^z(\beta_i)$ is also done separately, see lines 21 and 24 of the Protocol. Thus, Alice and Oscar keep their measurement angles and the random variables to themselves. □

Lemma 2. *[5] Suppose the open graph state* $(\mathcal{G}^{>}, I, O)$ *has flow* $(f, \succ)$ *and a proper total order* $>$, *then* $A(i)$ *contains at least* $f(i)$ *for all* $i \in O^{\mathrm{c}}$ *and* $A(i) \cap A(j) = \varnothing$ *for all* $i \neq j$.

Proof of Lemma 2. We need to prove that $f(i) \in A(i)$ and $A(i) \cap A(k) = \varnothing, k \neq i$, for all $i, k \in O^{\mathrm{c}}$.

Proof of the first part: using the existence of flow $f\colon O^{\mathrm{c}} \to I^{\mathrm{c}}$, thus $f(i) =: j$ exists (where $j \in I^{\mathrm{c}}$), and equation (7) applies. It follows that $j \in N_{\mathcal{G}}(i)$, and also $j \in N_{\mathcal{G}}[i]$, where $j \succ i$. But $j \notin \cup_{k<i} N_{\mathcal{G}}[k]$, because

[27] Note that f is injective, which is shown in proposition 1.

[28] In this condition the computation is within the 1WQC scheme.

[29] This can be seen as UBQC with graph $\mathcal{G}$ instead of the brickwork graph [10].

[30] Within the gate representation, measuring i in γ means $M_i^{\gamma}\rho = \text{tr}[P_i H_i Z_i(-\gamma)\rho]$, where P is measurement projector in the computational basis; thus, $M_i^{\gamma} N_i^{\gamma} = M_i^0 N_i^0$.

[31] This can be considered as a measurement that is X-dependent.

the flow condition $k \in N_{\mathcal{G}}(j)\backslash\{i\}, k \succ i$ (it is also true that $k > i$);[32] then clearly $j \notin I$. Thus, $A(i)$ contains at least $f(i)$.

Proof of the second part: denote $a_i := N_{\mathcal{G}}[i]$, $b_i := I \cup_{j<i} N_{\mathcal{G}}[j]$, thus $A(i) \equiv a_i \backslash b_i$. Given that every node has an ordering, consider the case $i < k$:

$$A(i) \cap A(k) \equiv (a_i \backslash b_i) \cap (a_k \backslash b_k) = [(a_i \backslash b_i)\backslash b_k] \cap a_k \overset{b_i \subset b_k}{=} (a_i \backslash b_k) \cap a_k \overset{a_i \subset b_k}{=} \varnothing.$$

It is clear that $b_i \subset b_k$ and $a_i \subset b_k$, since $i < k$. □

Lemma 3. *[5] Suppose the open graph state* $(\mathcal{G}^{>}, I, O)$ *has flow* $(f, \succ)$ *and a proper total order* $>$*, then* $\cup_{i \in V} A(i) = I^c$.

Proof of Lemma 3. Let $\mathcal{G} = (V, E)$. Recall that $\cup_{i\in V} A(i) = I^c \Leftrightarrow \cup_{i\in V} A(i) \subseteq I^c$ and $\cup_{i \in V} A(i) \supseteq I^c$; we assume the contrary, that is, let $\cup_{i\in V} A(i) = S$, thus $S \neq I^c$, i.e. either $S \subsetneq I^c$ or $S \supsetneq I^c$ is true.

Consider the first case, where $S \subsetneq I^c$. Here, there exists $k \in I^c$, such that $k \notin A(i)$ for all $i \in I^c$. We split this first case proof into two parts: for $i \in O^c$ and for $i \in O$.

For $i \in O^c$, by using lemma 2, $A(i)$ contains at least $f(i)$. Note that $f\colon O^c \to I^c$; thus, $\mathrm{inv}f(k) =: j$ exists, where $j \in O^c$. Therefore, $k \in A(j)$. Moreover, from lemma 2 we know that $S \neq \varnothing$.

For $i \in O$, we have $A(i) \subset O$ because if $A(i)$ contains an element in O^c, it is covered in the case $i \in O^c$ above. As output nodes also have a total ordering, $\cup_{i\in O} A(i)$ covers all output nodes that are disjoint to O^c nodes.

We arrive at a contradiction, so the assumption is incorrect, and we conclude that $I_c \backslash S = \varnothing$ or $I_c \subseteq S$.

Consider the second case, where $S \supsetneq I^c$. Here, there exists a $k \in I^c$, such that $A(k)$ contains m where $m \notin I^c$. By definition, $A(k)$ contains at most its closed neighborhood excluding the inputs, viz $N_{\mathcal{G}}(k)\backslash I$. Thus, $A(k)$ must be in the graph, since $\mathcal{G}[N_{\mathcal{G}}(k)]$ is a subgraph of $\mathcal{G}$; also, any element of I cannot be in $A(k)$. We again arrive at a contradiction, and must conclude that $S\backslash I_c = \varnothing$ or $S \subseteq I_c$.

Since $S \subseteq I^c$ and $I^c \subseteq S$, it follows that $S = I^c$. □

Lemma 4. *[5] Suppose an open graph state* $(\mathcal{G}^{>}, I, O)$ *has flow* $(f, \succ)$ *and a proper total order* $>$*, then* $\prod_{i\in V} E^{>}_{iN_{\mathcal{G}}(i)} = E_{\mathcal{G}}$*, where* $E^{>}_{iN_{\mathcal{G}}(i)} := \prod_{k \in N_{\mathcal{G}}(i), k>i} E_{ik}$.

Proof of Lemma 4. Note that we are employing the notation of equation (4). Denote (i, j) as an edge in $E(\mathcal{G})$ with end nodes i and j. If we represent $\prod_{i\in V} E^{>}_{iN_{\mathcal{G}}(i)}$ as a collection of edges, we obtain $\bigcup_{i\in V}\{(i,k)|k \in N_{\mathcal{G}}(i), k > i\} =: S$. On the other hand, the Handshaking lemma [40] implies that, $\biguplus_{i\in V}\{(i,k)|k\in N_{\mathcal{G}}(i)\}$ will result in a multiset that contains double copies of edges—$\forall\ a \in E(\mathcal{G})$ the multiplicity of a is 2, where $\uplus$ signifies the union of multisets [41]. Thus restricting to edges (i, k) where $i < k$ eliminates the double counting; therefore $S = E(\mathcal{G})$. □

Theorem 4. *[5] The BOQCo protocol* π_{boqco} *defined in protocol 2 delegates a computation with the isometry defined in theorem 1 for the same computation, without requiring Alice and Oscar to communicate their computation to each other.*

Proof of Theorem 4. We use the same reasoning as in the proof of theorem 2 by showing that the resulting pattern from protocol 2 can be reduced to the pattern in theorem 1. Note that, in this context, converters $\pi_{\mathcal{A}}, \pi_{\mathcal{O}}$, and $\pi_{\mathcal{B}}$ correspond to procedures of the BOQCo protocol defined in protocol 2; real resource $\mathcal{R}$ comprises the same elements as the BOQC viz a secure key, two-way classical channels, and two-way quantum channels.

Consider Alice and Oscar as one party, which we call super-Alice, who has all information about the angles ϕ, ψ and the random variables α, β, r, t. We denote $\gamma := \alpha \cup \beta$ and $\theta := \phi \cup \psi$.

First, we omit randomness, so that $r_i = t_i = \alpha_i = \beta_i = 0$ for all i; the resulting BOQCo pattern ($\mathfrak{P}_{\mathrm{boqco}}$) can be written as

$$\mathfrak{P}_{boqco} := \bigotimes_{j\in O} X_j^{s_{invf(j)}} Z_j^{z(j)} E_O \prod^{\succ}_{i\in O^c} M_i^{\theta_i} X_i^{s_{invf(i)}} Z_i^{z(i)} E^{>}_{iN_{\mathcal{G}}(i)} N^0_{A(i)} = \mathfrak{P}_{lazy} \overset{\text{Theo.3}}{=} \mathfrak{P}_{1wqc},$$

where $z(i) := \bigoplus_{k<i, i\in N_{\mathcal{G}}(f(k))} s_k$, $\mathrm{inv}f(i) \equiv f^{-1}(i)$, $\mathfrak{P}_{\mathrm{lazy}}$ is the resulting pattern of lazy 1WQC (equation (15)), and $\mathfrak{P}_{\mathrm{1wqc}}$ is the resulting pattern of 1WQC (equation (9)), where

$$\mathfrak{P}_{1wqc} := \prod^{\succ}_{i\in O^c} (X^{s_i}_{f(i)} \prod_{k\in N_{\mathcal{G}}(f(i))\backslash\{i\}} Z_i^{s_i} M_i^{\theta_i}) E_{\mathcal{G}} N^0_{I^c}.$$

[32] The flow must not have a neighbour in the past, i.e. a neighbour that has already measured.

Thus, $\mathfrak{P}_{\text{boqco}} = \mathfrak{P}_{\text{1wqc}}$ holds in the absence of randomness. Note that total ordering $>$ is consistent with partial ordering $\succ$ (partial ordering induced by the flow); thus we can interchangeably use both. Recall that operator $\bigotimes$ allows for concurrency, whereas $\prod$, $\overset{\succ}{\prod}$, and $\overset{>}{\prod}$ indicate serial operations.

Second, we introduce random variables γ and r into the protocol; now, pattern $\mathfrak{P}_{\text{boqco}}$ becomes

$$\mathfrak{P}_{boqc} := \overset{\succ}{\prod_{i \in O^c}} (X_{f(i)}^{s_i} \prod_{k \in N_{\mathcal{G}}(f(i)) \setminus \{i\}} Z_k^{s_i + r_i} M_i^{\theta_i + \gamma_i + \pi r_i}) E_{\mathcal{G}} N_{I^c}^{\gamma}. \tag{25}$$

which is identical to equation (24). Thus, from this point, the proof proceeds identically to that of theorem 2 from equation (24). □

Appendix B. Protocols and relaxations

Protocol 3. BOQC with classical input–output.

```
Alice's input: $\{(\mathcal{G}, I, O), f, \phi, c = c_1c_2, \ldots, c_n\}$    ▷ $\tilde{I} = \tilde{O} = \varnothing, \rho_{\mathcal{A}}^{\text{in}} = \prod_{i=1}^{n} |c_i\rangle\langle c_i|$
Oscar's input: $\{\psi\}$
Alice's output for an honest Bob: $\rho_{\mathcal{A}}^{\text{out}} = \mathcal{E}(\rho_{\mathcal{A}}^{\text{in}})$    ▷ $\rho_{\mathcal{A}}^{\text{out}}$ is a diagonal matrix
  Assumptions and conventions:
    (I) Alice ($\mathcal{A}$) and Oscar ($\mathcal{O}$) have performed pre-protocol steps in definition 3; Bob knows $\{(\mathcal{G}, \tilde{I}, \tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$. Here, we set
      $\tilde{I} = \tilde{O} = \varnothing$, and recall $\Omega = \{\frac{\pi k}{2^{b-1}}\}_{0 \leqslant k < 2^b}$.
    (II) $s_{\text{invf}(i)} = 0, \forall i \in I$.
    (III) $\text{invf}(i) \equiv f^{-1}(i)$ and $z(i) := \oplus_{k \prec i, i \in N_{\mathcal{G}}(f(k))} s_k$.
⓿ Pre-preparation
Alice and Oscar receive a key $r$ via a secure key channel, where $r_i \in \{0, 1\}$, for $i \in O^c$.
❶ State preparation
for $i \in V$ which follows partial ordering $\succ$ do
    if $i \in V_{\mathcal{A}}$ then
        if $i \in I$ and explicit input then
            Alice prepares $|+_{\alpha_i + c_i\pi}\rangle_i$ and sends it to Bob; $\alpha_i \in \Omega$ is chosen at random.
        else
            Alice prepares $|+_{\alpha_i}\rangle_i$ and sends it to Bob, where $\alpha_i \in \Omega$ is chosen at random
        end if
    else if $i \in V_{\mathcal{O}}$ then
        Oscar prepares $|+_{\beta_i}\rangle_i$ and sends it to Bob; $\beta_i \in \Omega$ is chosen at random.
    end if
end for
❷ Graph state formation
Bob applies entangling operator $E_{\mathcal{G}}$ defined in equation (4).
❸ Classical interaction and measurement
for $i \in V$ which follows partial ordering $\succ$ do    ▷ e.g., it follows $>$
    if $i \in V_{\mathcal{A}}$ then
        Alice computes $\phi_i' = (-1)^{s_{invf(i)}} \phi_i + z(i)\pi$.
        Alice computes $\delta_i := \phi_i' + \pi r_i + \alpha_i$, and sends Bob $\delta_i$.
    else if $i \in V_{\mathcal{O}}$ then
        Oscar computes $\psi_i' = (-1)^{s_{invf(i)}} \psi_i + z(i)\pi$.
        Oscar computes $\delta_i := \psi_i' + \pi r_i + \beta_i$, and sends Bob $\delta_i$.
    end if
    Bob measures qubit $i$ in basis $|\pm_{\delta_i}\rangle$, then sends Alice and Oscar the outcome $\tilde{s}_i$.
    Alice and Oscar set $s_i = \tilde{s}_i \oplus r_i$.
end for
```

Protocol 4. BOQCo with classical input–output.

```
Alice's input: $\{(\mathcal{G}, I, O), f, \phi, c = c_1c_2, \ldots, c_n\}$        ▷ $\tilde{I} = \tilde{O} = \varnothing, \rho_{\mathcal{A}}^{\text{in}} = \prod_{i=1}^{n} |c_i\rangle\langle c_i|$
Oscar's input: $\{\psi\}$
Alice's output for an honest Bob: $\rho_{\mathcal{A}}^{\text{out}} = \mathcal{E}(\rho_{\mathcal{A}}^{\text{in}})$        ▷ $\rho_{\mathcal{A}}^{\text{out}}$ is a diagonal matrix
  Assumptions and conventions:
    (I) Alice ($\mathcal{A}$) and Oscar ($\mathcal{O}$) have performed pre-protocol steps in definition 3; Bob knows $\{(\mathcal{G}, \tilde{I}, \tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$. Here, we set
      $\tilde{I} = \tilde{O} = \varnothing$, and recall $\Omega = \{\frac{\pi k}{2^{b-1}}\}_{0 \leqslant k < 2^b}$.
    (II) $s_{\text{inv}f(i)} = 0, \forall i \in I$.
    (III) $\text{inv}f(i) \equiv f^{-1}(i)$ and $z(i) := \oplus_{k \prec i, i \in N_{\mathcal{G}}(f(k))} s_k$.
⓪ Pre-preparation
Alice and Oscar receive a key $r$ via a secure key channel, where $r_i \in \{0, 1\}$, for $i \in O^c$.
① BOQC by parts
for $i \in V$ with ordering $>$ do
    for $k \in A(i) \cup I$ do        ▷ See equation (14) in section 2.1
        if $k \in I$ and classical input then        ▷ Input qubit
            Alice prepares $|+_{\alpha_k + c_k\pi}\rangle_k$ and sends it to Bob; $\alpha_k \in \Omega$ is chosen at random.
        else        ▷ Auxiliary qubit
            if $k \in V_{\mathcal{A}}$ then
                Alice prepares $|+_{\alpha_k}\rangle_k$ and sends it to Bob, $\alpha_k \in \Omega$ is chosen at random.
            else if $k \in V_{\mathcal{O}}$ then
                Oscar prepares $|+_{\beta_k}\rangle_k$ and send it to Bob, $\beta_k \in \Omega$ is chosen at random.
            end if
        end if
    end for
    Bob applies entangling operation $E^{>}_{iN_{\mathcal{G}}(i)}$.        ▷ See equation (4) in section 2.1
    if $i \in V_{\mathcal{A}}$ then
        Alice computes $\phi'_i = (-1)^{s_{invf(i)}} \phi_i + z(i)\pi$.
        Alice computes $\delta_i := \phi'_i + \pi r_i + \alpha_i$ and sends Bob $\delta_i$.
    else if $i \in V_{\mathcal{O}}$ then
        Oscar computes $\psi'_i = (-1)^{s_{invf(i)}} \psi_i + z(i)\pi$.
        Oscar computes $\delta_i := \psi'_i + \pi r_i + \beta_i$ and sends Bob $\delta_i$.
    end if
    Bob measures $i$ in $|\pm_{\delta_i}\rangle$ basis, then sends Alice and Oscar the outcome $\tilde{s}_i$.
    Alice and Oscar set $s_i = \tilde{s}_i \oplus r_i$.
end for
```

Protocol 5. [5] A relaxation $\mathcal{S}'\sigma_{\mathcal{B}}$.

```
Conventions:
    (I) given public information $\{(\mathcal{G},\tilde{I},\tilde{O}),V_{\mathcal{A}},V_{\mathcal{O}},\succ,>,b\}$, where $\tilde{I}=I$, $\tilde{O}=O$, $\mathcal{G}=(V,E)$, $V=V_{\mathcal{A}}\cup V_{\mathcal{O}}$, and $\Omega=\{\frac{\pi k}{2^{b-1}}\}_{0\leqslant k<2^{b}}$.
    (II) $s_{invf(i)}=0, \underline{s}_{invf(i)}=0$ for all $i\in I$ and $t_i=0$ for all $i\in I^{c}$.
    (III) $\mathrm{invf}(i)\equiv f^{-1}(i)$, $z(i):=\bigoplus_{k<i,i\in N_{\mathcal{G}}(f(k))}s_k$, $\underline{z}(i):=\bigoplus_{k<i,i\in N_{\mathcal{G}}(f(k))}\underline{s}_k$, and $t(i):=\sum_{k\in I,i\in N_{\mathcal{G}}(k)}t_k$.
    (IV) Measurements are performed in the computational basis.
The simulator $\sigma_{\mathcal{B}}$
1: Prepares an EPR pair $(|00\rangle+|11\rangle)/\sqrt{2}$ for every node $i\in O^{c}$ and outputs its half.
2: Picks random angles $\{\delta_i\in\Omega|i\in O^{c}\}=:\delta$ and outputs it.
3: Receives responses $\{\tilde{s}_i\in\{0,1\}|i\in O^{c}\cap V_{\mathcal{A}}\}=:\tilde{s}$ and $\{\tilde{\underline{s}}_i\in\{0,1\}|i\in O^{c}\cap V_{\mathcal{O}}\}=:\tilde{\underline{s}}$.
4: Receives the corresponding output qubits $\rho_{\mathcal{B}}^{\mathrm{out}}$ (all qubits $i\in O$).
5: Sends $\mathcal{S}'$ the other halves of EPR pairs ($\{\mathrm{half}-\mathrm{EPR}-i|i\in O^{c}\}=:$ EPRs), $\delta,\tilde{s},\tilde{\underline{s}}$, and $\rho_{\mathcal{B}}^{\mathrm{out}}$.
The ideal BOQC resource $\mathcal{S}'$
6: Receives $\{(\mathcal{G},I,O),f,\rho_{\mathcal{A}}^{\mathrm{in}},\phi\}$ at Alice's interface and $\psi$ at Oscar's interface, and information from step 5—EPRs, $\delta,\tilde{s},\tilde{\underline{s}}$, and $\rho_{\mathcal{B}}^{\mathrm{out}}$.
7: Applies CNOT gates between $\mathrm{tr}_{I\setminus i}[\rho_{\mathcal{A}}^{\mathrm{in}}]$ (as control) and half-EPR-$i$, for all $i\in I$, then measures the half-EPR, stores the outcome as $t_i$, and updates measurement angles:
       $\forall\ i\in I,\phi_i=(-1)^{t_i}\phi_i,$
       $\forall\ i\in I,\ \forall\ j\in N_{\mathcal{G}}(i)\cap V_{\mathcal{A}},\phi_j=\phi_j+t_i\pi$
       $\forall\ i\in I,\forall\ j\in N_{\mathcal{G}}(i)\cap V_{\mathcal{O}},\psi_j=\psi_j+t_i\pi.$
8: for $i\in O^{c}$ which follows ordering $>$ do
9:    if $i\in V_{\mathcal{A}}$ then
10:       Computes $\phi'_i=(-1)^{s_{invf(i)}}\phi_i+z(i)\pi$.
11:       Computes $\theta'_i=\delta_i-\phi'_i$.
12:   else if $i\in V_{\mathcal{O}}$ then
13:       Computes $\psi'_i=(-1)^{\underline{s}_{invf(i)}}\psi_i+\underline{z}(i)\pi$.
14:       Computes $\theta'_i=\delta_i-\psi'_i$.
15:   end if
16:   if $i\in I$ then
17:       Applies $HZ_i(\theta_i)$ to an input qubit $\mathrm{tr}_{I\setminus i}[\rho_{\mathcal{A}}^{\mathrm{in}}]$ followed by measurement.
18:   else
19:       Applies $HZ_i(\theta_i)$ to the half-EPR-$i$ followed by measurement
20:   end if
21:   Stores the measurement outcome as $r_i$ and sets $s_i=\tilde{s}_i\oplus r_i$ and $\underline{s}_i=\tilde{\underline{s}}_i\oplus r_i$.
22: end for
23: Corrects the output $P(\rho_{\mathcal{B}}^{\mathrm{out}})=:\rho_{\mathcal{A}}^{\mathrm{out}}$, where $P\equiv\bigotimes_{i\in O}X_i^{s_{invf(i)}+t_i}Z_i^{z(i)+t(i)}$, then outputs it on Alice's interface.
```

Protocol 6. [5] A relaxation ($\mathcal{S}'\sigma_{\mathcal{B}}$) for BOQCo.

```
Conventions:
  (I) given public information $\{(\mathcal{G},\tilde{I},\tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$,where $\tilde{I} = I$, $\tilde{O} = O$, $\mathcal{G} = (V, E)$,
    $V = V_{\mathcal{A}} \cup V_{\mathcal{O}}$, and $\Omega = \{\frac{\pi k}{2^{b-1}}\}_{0 \leqslant k < 2^b}$.
  (II) $s_{invf(i)} = 0$, $\underline{s}_{invf(i)} = 0$ for all $i \in I$ and $t_i = 0$ for all $i \in I^c$.
  (III) invf$(i) \equiv f^{-1}(i)$, $z(i) := \bigoplus_{k<i, i \in N_{\mathcal{G}}(f(k))} s_k$, $\underline{z}(i) := \bigoplus_{k<i, i \in N_{\mathcal{G}}(f(k))} \underline{s}_k$, and $t(i) := \sum_{k \in I, i \in N_{\mathcal{G}}(k)} t_k$.
  (IV) Measurements are performed in the computational basis.
Part 1
$\mathcal{S}'$ receives $\{(\mathcal{G}, I, O), f, \rho^{\text{in}}_{\mathcal{A}}, \phi\}$ at Alice's interface and $\psi$ at Oscar's interface.
Part 2
for $i \in V \setminus O$ with ordering $>$ do
    for $k \in A(i) \cup I$ do                                        ▷ See equation (14) in section 4.1
  The simulator $\sigma_{\mathcal{B}}$
        if $k \notin O$ then
            Prepares an EPR pair $(|00\rangle + |11\rangle)/\sqrt{2}$ and outputs its half.
            Picks $\delta_k \in \Omega$ at random, outputs $\delta_k$.
            Receives a responses $\tilde{s}_k$ and $\underline{\tilde{s}}_k$, where $\tilde{s}_k, \underline{\tilde{s}}_k \in \{0, 1\}$.
            Sends resource $\mathcal{S}'$ the other half of EPR pair (half-EPR-$k$), $\delta_k$,$\tilde{s}_k$, and $\underline{\tilde{s}}_k$.
        else if $k \in O$ then
            Receives output qubit $\text{tr}_{O\setminus k}[\rho^{\text{out}}_{\mathcal{B}}]$, and sends it to the ideal resource.
        end if
  The ideal BOQCo resource $\mathcal{S}'$
        Receives {half-EPR-$k$, $s_k$, $\tilde{s}_k$, $\underline{\tilde{s}}_k$, $\delta_k$} if $k \in O^c$, otherwise receives $\text{tr}_{O\setminus k}[\rho^{\text{out}}_{\mathcal{B}}]$.
        if $k \in I$ then
            Applies CNOT gate between $\text{tr}_{I\setminus k}[\rho^{\text{in}}_{\mathcal{A}}]$ and half-EPR-$k$.
            Measures half-EPR-$k$, stores the outcome as $r_k$, then updates:
                $$\phi_k = (-1)^{t_k}\phi_k,$$
                $$\forall\, j \in N_{\mathcal{G}}(k) \cap V_{\mathcal{A}}, \phi_j = \phi_j + t_k\pi$$
                $$\forall\, j \in N_{\mathcal{G}}(k) \cap V_{\mathcal{O}}, \psi_j = \psi_j + t_k\pi.$$
        end if
        if $k \in O^c$ then
            if $k \in V_{\mathcal{A}}$ then
                Computes $\phi'_k = (-1)^{s_{invf(k)}}\phi_k + z(k)\pi$.
                Computes $\theta'_k = \delta_k - \phi'_k$.
            else if $k \in V_{\mathcal{O}}$ then
                Computes $\psi'_k = (-1)^{\underline{s}_{invf(k)}}\psi_k + \underline{z}(k)\pi$.
                Computes $\theta'_k = \delta_k - \psi'_k$
            end if
            Applies $HZ_k(\theta'_k)$ to half-EPR-$k$ followed by measurement.
            Stores the measurement outcomes as $r_k$.
            Sets $s_k = \tilde{s}_k \oplus r_k$ and $\underline{s}_k = \underline{\tilde{s}}_k \oplus r_k$ if $k \in V_{\mathcal{O}}$.
        else                                                          ▷ $k \in O$
            Corrects $\text{tr}_{O\setminus k}[\rho^{\text{out}}_{\mathcal{B}}]$ with $X_k^{s_{invf(k)}+t_k} Z_k^{z(k)+t(k)}$ and outputs it on Alice's interface.
        end if
    end for
end for
```

Appendix C. Using filter on the security proof of BOQC/BOQCo

In the following, we prove the composable blindness of the BOQC; the blindness of BOQCo is inherited from the BOQC by constructing a proof as done in theorems 6 and 8.

Using the filter notion, we can combine ideal resources $\mathcal{S}$ (figure 4) that captures correctness and $\mathcal{S}'$ (figure 5) that captures blindness into a single resource $\mathcal{S}_{\text{boqc}}$ (figure 9) that captures both correctness and blindness. Note that ▣ captures all classical and quantum input from Bob; for clarity, we explicitly add another input b to indicate whether the filter is on or off.

Figure 10 shows the ideal blind UBQC—or blind delegated computing resource in general—by Dunjko *et al* [15] adapted to our notation. The ideal resource $\mathcal{S}^{\text{blind}}$ leaks a permitted classical string that contains information needed to set up the protocol. Since we allow the graph to be inputted (not necessarily to be brickwork state), the leak in $\mathcal{S}^{\text{blind}}$ is the same as the leak of BOQC, i.e. $\{(\mathcal{G},\tilde{I},\tilde{O}), V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >\}$, given that Alice is running an oracular quantum algorithm and has full knowledge of her computation. Thus, given that Alice knows Oscar's input, $\mathcal{S}_{\text{boqc}}$ is directly reduced to $\mathcal{S}^{\text{blind}}$. Therefore, BOQC is useful when Alice does not have knowledge of the oracle.

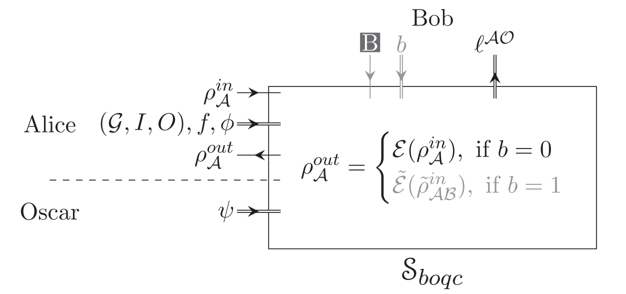

Bob
B
b
$\ell^{\mathcal{AO}}$
Alice
$\rho_{\mathcal{A}}^{in}$
$(\mathcal{G}, I, O), f, \phi$
$\rho_{\mathcal{A}}^{out}$
Oscar
ψ
$\rho_{\mathcal{A}}^{out} = \begin{cases} \mathcal{E}(\rho_{\mathcal{A}}^{in}), & \text{if } b = 0 \\ \tilde{\mathcal{E}}(\tilde{\rho}_{\mathcal{AB}}^{in}), & \text{if } b = 1 \end{cases}$
$\mathcal{S}_{boqc}$

Figure 9. The ideal BOQC resource $\mathcal{S}_{\text{boqc}}$ captures both correctness and blindness. See figure 5 for the notation. Additionally, gray color indicates functionalities accessible to Bob when the filter is removed. We indicate $b = 0$ when the filter is present and $b = 1$ when the filter is removed.

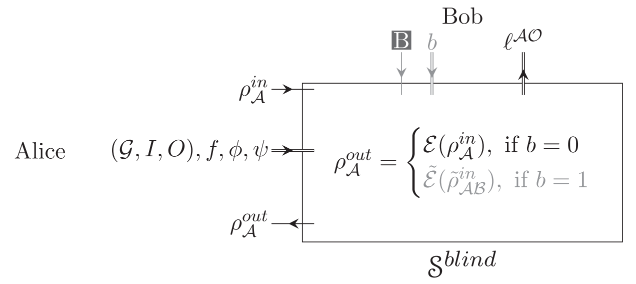

Bob
B
b
$\ell^{\mathcal{AO}}$
$\rho_{\mathcal{A}}^{in}$
Alice
$(\mathcal{G}, I, O), f, \phi, \psi$
$\rho_{\mathcal{A}}^{out}$
$\rho_{\mathcal{A}}^{out} = \begin{cases} \mathcal{E}(\rho_{\mathcal{A}}^{in}), & \text{if } b = 0 \\ \tilde{\mathcal{E}}(\tilde{\rho}_{\mathcal{AB}}^{in}), & \text{if } b = 1 \end{cases}$
$\mathcal{S}^{blind}$

Figure 10. [15] The ideal UBQC resource $\mathcal{S}^{\text{blind}}$ that captures both correctness and blindness adapted to our notation.

From the result of [15], we know that the UBQC protocol is correct and blind, fulfilling

$$\tilde{\pi}_{\mathcal{A}}\tilde{\mathcal{R}}\tilde{\pi}_{\mathcal{B}} = \mathcal{S}^{\text{blind}}\perp_{\mathcal{B}} \quad \text{and} \quad \tilde{\pi}_{\mathcal{A}}\tilde{\mathcal{R}} = \mathcal{S}^{\text{blind}}\tilde{\sigma}_{\mathcal{B}} \tag{26}$$

respectively. Here, $\tilde{\pi}_{\mathcal{A}}$ and $\tilde{\pi}_{\mathcal{B}}$ are Alice's and Bob's protocol in the UBQC, $\tilde{\mathcal{R}}$ indicates the real resource in the UBQC, $\perp_{\mathcal{B}}$ indicates the filter ($b = 0$), and $\tilde{\sigma}_{\mathcal{B}}$ denotes the simulator defined in protocol 4 of reference [15].

Notice that $\tilde{\pi}_{\mathcal{A}}\tilde{\mathcal{R}}$ has the same input–output as protocol $\pi_{\mathcal{AO}}\mathcal{R}$ of the BOQC in figure 6. The input–output of $\tilde{\pi}_{\mathcal{A}}\tilde{\mathcal{R}}$ can be observed from the UBQC protocol itself, i.e. protocol 1 of reference [15]. Using the proof of theorem 7, we know that

$$\pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R} = \pi_{\mathcal{AO}}\mathcal{R} = \tilde{\pi}_{\mathcal{A}}\tilde{\mathcal{R}} = \mathcal{S}^{\text{blind}}\tilde{\sigma}_{\mathcal{B}}, \tag{27}$$

which proves the blindness of BOQC. Therefore, we see that our simulator $\sigma_{\mathcal{B}}$ in protocol 5 can be similar to the simulator of Dunjko *et al* in protocol 4 of reference [15]. It is also consistent with the fact that $\mathcal{S}_{\text{boqc}}$ can be reduced to $\mathcal{S}^{\text{blind}}$ when Alice has knowledge of the oracles.

The correctness of the BOQC comes from the fact that $\mathcal{S}\perp_{\mathcal{B}} = \mathcal{S} = \pi_{\mathcal{A}}\pi_{\mathcal{O}}\mathcal{R}\pi_{\mathcal{B}}$, using the proof of theorem 5. Hence, we conclude that BOQC is composable blind—proven in the fashion of Dunjko *et al*.

ORCID iDs

Cica Gustiani https://orcid.org/0000-0003-0558-4685
David P DiVincenzo https://orcid.org/0000-0003-4332-645X

References

[1] Regev O and Schiff L 2008 Impossibility of a quantum speed-up with a faulty oracle *Int. Coll. Automata, Languages, and Programming* (Berlin: Springer) pp 773–81

[2] Harrow A W and Rosenbaum D J 2011 Uselessness for an oracle model with internal randomness (arXiv:1111.1462)

[3] Zhandry M 2018 How to record quantum queries, and applications to quantum indifferentiability *Report 2018/276* Cryptology ePrint Archive

[4] Meyer D A and Pommersheim J 2009 Single query learning from abelian and nonabelian Hamming distance oracles (arXiv:0912.0583)

[5] Gustiani C 2020–2021 Blind oracular quantum computation: from concept to physical implementation *PhD Thesis* RWTH Aachen University

[6] Aaronson S *et al* 2017 Complexity-theoretic limitations on blind delegated quantum computation (arXiv:1704.08482)

[7] Reichardt B W, Unger F and Vazirani U 2013 Classical command of quantum systems *Nature* **496** 456–60

[8] Huang H-L *et al* 2017 Experimental blind quantum computing for a classical client *Phys. Rev. Lett.* **119** 050503

[9] Childs A M 2001 Secure assisted quantum computation (arXiv:quantph/0111046)

[10] Broadbent A, Fitzsimons J and Kashefi E 2009 Universal blind quantum computation *50th Annual IEEE Symp. Foundations of Computer Science 2009. FOCS'09* (Piscataway, NJ: IEEE) pp 517–26

[11] Fitzsimons J F 2017 Private quantum computation: an introduction to blind quantum computing and related protocols *npj Quantum Inf.* **3** 23

[12] Morimae T and Fujii K 2012 Blind topological measurement-based quantum computation *Nat. Commun.* **3** 1–6

[13] Jordan S 2018 Quantum algorithm zoo https://math.nist.gov/quantum/zoo/#oracular (accessed 25 May 2011)

[14] Maurer U and Renner R 2011 Abstract cryptography *Innovations in Computer Science - ICS 2011: Proceedings* (Beijing, China Tsinghua University Press) pp 1–21

[15] Dunjko V, Fitzsimons J F, Portmann C and Renner R 2014 Composable security of delegated quantum computation *Int. Conf. Theory and Application of Cryptology and Information Security* (Berlin: Springer) pp 406–25

[16] Portmann C and Renner R 2021 Security in quantum cryptography (arXiv:2102.00021)

[17] Raussendorf R and Briegel H J 2001 A one-way quantum computer *Phys. Rev. Lett.* **86** 5188–91

[18] Raussendorf R, Browne D E and Briegel H J 2003 Measurement-based quantum computation on cluster states *Phys. Rev.* A **68** 022312

[19] Jaksch D, Briegel H-J, Cirac J I, Gardiner C W and Zoller P 1999 Entanglement of atoms via cold controlled collisions *Phys. Rev. Lett.* **82** 1975–8

[20] Walther P, Resch K J, Rudolph T, Schenck E, Weinfurter H, Vedral V, Aspelmeyer M and Zeilinger A 2005 Experimental one-way quantum computing *Nature* **434** 169–76

[21] Prevedel R, Walther P, Tiefenbacher F, Böhi P, Kaltenbaek R, Jennewein T and Zeilinger A 2007 High-speed linear optics quantum computing using active feed-forward *Nature* **445** 65–9

[22] Chen K, Li C-M, Zhang Q, Chen Y-A, Goebel A, Chen S, Mair A and Pan J-W 2007 Experimental realization of one-way quantum computing with two-photon four-qubit cluster states *Phys. Rev. Lett.* **99** 120503

[23] Tame M S, Prevedel R, Paternostro M, Böhi P, Kim M S and Zeilinger A 2007 Experimental realization of Deutsch's algorithm in a one-way quantum computer *Phys. Rev. Lett.* **98** 140501

[24] Barz S, Kashefi E, Broadbent A, Fitzsimons J F, Zeilinger A and Walther P 2012 Demonstration of blind quantum computing *Science* **335** 303–8

[25] Greganti C, Roehsner M-C, Barz S, Morimae T and Walther P 2016 Demonstration of measurement-only blind quantum computing *New J. Phys.* **18** 013020

[26] Gustiani C and DiVincenzo D P 2019 Three-qubit exact Grover within the blind oracular quantum computation scheme (arXiv:1902.05534)

[27] Houshmand M, Houshmand M and Fitzsimons J F 2018 Minimal qubit resources for the realization of measurement-based quantum computation *Phys. Rev.* A **98** 012318

[28] Danos V, Kashefi E and Panangaden P 2007 The measurement calculus *JACM* **54** 8

[29] Danos V and Kashefi E 2006 Determinism in the one-way model *Phys. Rev.* A **74** 052310

[30] Briegel H J, Browne D E, Dür W, Raussendorf R and Van den Nest M 2009 Measurement-based quantum computation *Nat. Phys.* **5** 19

[31] Browne D E, Kashefi E, Mhalla M and Perdrix S 2007 Generalized flow and determinism in measurement-based quantum computation *New J. Phys.* **9** 250

[32] Maurer U 2012 Constructive cryptography—a new paradigm for security definitions and proofs *Theory of Security and Applications* ed S Mödersheim and C Palamidessi (Berlin: Springer) pp 33–56

[33] Chiribella G, D'Ariano G M and Perinotti P 2008 Quantum circuit architecture *Phys. Rev. Lett.* **101** 060401

[34] Goldreich O 2006 On promise problems: a survey *Theoretical Computer Science: Essays in Memory of Shimon Even* ed O Goldreich, A L Rosenberg and A L Selman (Berlin: Springer) pp 254–90

[35] Ambainis A, Bouda J and Winter A 2009 Nonmalleable encryption of quantum information *J. Math. Phys.* **50** 042106

[36] Fitzsimons J F and Kashefi E 2017 Unconditionally verifiable blind quantum computation *Phys. Rev.* A **96** 012303

[37] Barz S, Fitzsimons J F, Kashefi E and Walther P 2013 Experimental verification of quantum computation *Nat. Phys.* **9** 727–31

[38] Watt D 2004 *Programming Language Design Concepts* (New York: Wiley)

[39] Bennett C H *et al* 2001 Remote state preparation *Phys. Rev. Lett.* **87** 077902

[40] Biggs N, Lloyd E K and Wilson R J 1986 *Graph Theory, 1736-1936* (Oxford: Oxford University Press) p 10

[41] Syropoulos A 2000 Mathematics of multisets *Workshop on Membrane Computing* (Berlin: Springer) pp 347–58