Composable Security of Delegated Quantum Computation

Vedran Dunjko^{*,1,2}, Joseph F. Fitzsimons^{3,4}, Christopher Portmann^{5,6}, and Renato Renner^{5}

[1] School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, U.K.
[2] Division of Molecular Biology, Ruđer Bošković Institute, Bijenička cesta 54, P.P. 180, 10002 Zagreb, Croatia.
[3] Singapore University of Technology and Design, 20 Dover Drive, Singapore 138682.
[4] Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, Singapore 117543. joe.fitzsimons@nus.edu.sg.
[5] Institute for Theoretical Physics, ETH Zurich, 8093 Zurich, Switzerland. {chportma,renner}@phys.ethz.ch.
[6] Group of Applied Physics, University of Geneva, 1211 Geneva, Switzerland.

Abstract. Delegating difficult computations to remote large computation facilities, with appropriate security guarantees, is a possible solution for the ever-growing needs of personal computing power. For delegated computation protocols to be usable in a larger context — or simply to securely run two protocols in parallel — the security definitions need to be composable. Here, we define composable security for delegated quantum computation. We distinguish between protocols which provide only *blindness* — the computation is hidden from the server — and those that are also *verifiable* — the client can check that it has received the correct result. We show that the composable security definition capturing both these notions can be reduced to a combination of several distinct "trace-distance-type" criteria — which are, individually, non-composable security definitions.

Additionally, we study the security of some known delegated quantum computation protocols, including Broadbent, Fitzsimons and Kashefi's Universal Blind Quantum Computation protocol. Even though these protocols were originally proposed with insufficient security criteria, they turn out to still be secure given the stronger composable definitions.

1 Introduction

1.1 Background

It is unknown in what form quantum computers will be built. One possibility is that large quantum servers may take a role similar to that occupied by massive superclusters today. They would be available as important components

* Now at: Institute for Theoretical Physics, University of Innsbruck, Technikerstraße 25, A-6020 Innsbruck, Austria. vedran.dunjko@uibk.ac.at.

in large information processing clouds, remotely accessed by clients using their home-based simple devices. The issue of the security and the privacy of the computation is paramount in such a setting.

Childs [16] proposed the first such delegated quantum computation (DQC) protocol, which hides the computation from the server, i.e., the computation is blind. This was followed by Arrighi and Salvail [2], who introduced a notion of verifiability — checking that the server does what is expected — but only for a restricted class of public functions. In recent years, this problem has gained a lot of interest, with many papers proposing new protocols, e.g., [1,11,15,19–21, 28,31–36,41], and even small-scale experimental realizations [7,8].

However, with the exception of recent work by Broadbent, Gutoski and Stebila [12], none of the previous DQC papers consider the *composability* of the protocol. They prove security by showing that the states held by the client and server fulfill some local condition: the server's state must not contain any information about the input and the client's final state must either be the correct outcome or an error flag. Even though this means that the server cannot — from the information leaked during a single execution of the protocol in an isolated environment — learn the computation or produce a wrong output without being detected, it does not guarantee any kind of security in any realistic setting. In particular, if a server treats two requests simultaneously or if the delegated computation is used as part of a larger protocol (such as the quantum coins of Mosca and Stebila [37]), these works on DQC cannot be used to infer security. A *composable security* framework must be used for a protocol to be secure in an arbitrary environment. In the following, we use the expression *local* to denote the non-composable security conditions previously used for DQC. This term is chosen, because these criteria consider the state of a (local) subsystem, instead of the global system as seen by a distinguisher in composable security.[7]

In fact, exactly these local properties have been proven to be insufficient to define secure communication. There exist protocols which are shown to both encrypt and authenticate messages by fulfilling local criteria equivalent to the ones used in DQC — the scheme is secure if the eavesdropper obtains no information about the message from the ciphertext and authentic if the receiver either gets the original message or an error flag. But if the eavesdropper learns whether the message was transmitted faithfully or not, she learns some information about this message [9,27,30]. Since any secure communication protocol can be seen as delegated computation for the identity operation — Eve is required to apply the identity operation to the message, but may cheat and try to learn or modify it — there is a strict gap between security of DQC and previously used local criteria.[8]

[7] Standard terms for various forms of non-composable security, e.g., *stand-alone* or *sequential*, have precise definitions which do not apply to these security criteria.

[8] An alternative example of this gap is as follows. The task is to compute a witness for a positive instance of an NP problem, and we do so with the following protocol: the server simply picks a witness at random and sends it to the client. Although the protocol does not achieve completeness, it appears to be sound: the protocol obviously does not leak any information about the input, since no information is sent

Composable frameworks have the further advantage that they require the interaction between different entities to be modeled explicitly, and often make hidden assumptions apparent. For example, it came as a surprise when Barrett et al. [6] showed that device independent quantum key distribution (DIQKD) is insecure if untrusted devices (with internal memory) are used more than once. It is however immediate when one models the security of DIQKD in a composable framework, that existing security proofs make the assumption that devices are used only once. Another example, the security definitions of zero-knowledge protocols [22] and coin expansion [26] make the assumption that the dishonest party executes his protocol without interaction with the environment.[9] By explicitly modeling this restriction,[10] these proofs can be lifted to a composable framework. This has been used by, e.g., Unruh [44], who explicitly limits the number of parallel executions of a protocol to achieve security in the bounded storage model.

Correctly defining the security of a cryptographic task is fundamental for a protocol and proof to have any usefulness or even meaning. In this paper we solve this problem for DQC, which has been open since the first version of Childs's work [16] was made available in 2001.

1.2 Scope and Security of DQC

A common feature of all DQC protocols is that the client, while not being capable of full-blown quantum computation, has access to limited quantum-enriched technology, which she needs to interact with the server. One of the key points upon which the different DQC protocols vary, is the complexity and the technical feasibility of the aforementioned quantum-enriched technology. In particular, in the proposal of Childs [16], the client has quantum memory, and the capacity to perform local Pauli operations. The protocol of Arrighi and Salvail [2] requires the client to have the ability to generate relatively involved superpositions of multi-qubit states, and perform a family of multi-qubit measurements. Aharonov, Ben-Or and Eban [1], for the purposes of studying quantum prover interactive proof systems, considered a DQC protocol in which the client has a constant-sized quantum computer. The blind DQC protocol proposed by Broadbent, Fitzsimons and Kashefi [11] has arguably the lowest requirements on the

from the client to the server. The client can also verify that the solution received is correct, and never accepts a wrong answer. But if the server ever learns whether the witness was accepted — e.g., it is composed with another protocol which makes this information public — he learns something about the input. If there are only two choices for the input with distinct witnesses, he learns exactly which one was used.

[9] The security definitions for these two problems are instances of what is generally known as *stand-alone security* [23].

[10] This can be done by introducing a resource — e.g., a trusted third party — that runs whatever circuits Alice and Bob give it in an isolated system, then returns the transcript of the protocol to both players.

client. In particular, she does not need any quantum memory,[11] and is only required to prepare single qubits in separable states randomly chosen from a small finite set analogous to the BB84 states.[12] Alternatively, Morimae and Fujii [32,35] propose a DQC protocol in which the client only needs to measure the qubits she receives from the server to perform the computation.

A second important distinction between these protocols is in the types of problems the protocol empowers the client to solve. Most protocols, e.g., [1,11,16,20,32,35], allow a client to perform universal quantum computation, whereas in [2] the client is restricted to the evaluation of random-verifiable[13] functions.

Finally, an important characteristic of these protocols is the flavor of security guaranteed to the client. Here, one is predominantly interested in two distinct features: privacy of computation (generally referred to as blindness) and verifiability of computation. Blindness characterizes the degree to which the computational input and output, and the computation itself, remain hidden from the server. This is the main security concern of, e.g., [11,16,35]. Verifiability ensures that the client has means of confirming that the final output of the computation is correct. In addition to blindness, some form of verifiability is given by, e.g., [1,2,20,32]. These works do however not concern themselves with the cryptographic soundness of their security notions. In particular, none of them consider the issue of composability of DQC. A notable exception is the recent work of Broadbent, Gutoski and Stebila [12], who, independently from our work, prove that a variant of the DQC protocol of Aharonov, Ben-Or and Eban [1] provides composable security.[14]

1.3 Composable Security

The first frameworks for defining composable security were proposed independently by Canetti [13,14] and by Backes, Pfitzmann and Waidner [3,4,39], who dubbed them *Universally Composable (UC) security* and *Reactive Simulatability*, respectively. These security notions have been extended to the quantum setting by Ben-Or and Mayers [10] and Unruh [42,43].

More recently, Maurer and Renner proposed a new composable framework, Abstract Cryptography (AC) [29]. Unlike its predecessors that use a bottom-up approach to defining models of computation, algorithms, complexity, efficiency, and then security of cryptographic schemes, the AC approach is top-down and

[11] This holds in the case of classical input and output. If quantum inputs and/or outputs are considered, then the client has to be able to apply a quantum one-time pad to the input state, and also decrypt a quantum one-time pad of the output state.

[12] The states needed by the protocol of [11] are $\{(|0\rangle + e^{ik\pi/4}|1\rangle)/\sqrt{2}\}_k$ for $k \in \{0,\ldots,7\}$.

[13] Roughly speaking, a function f is random-verifiable if pairs of instances and solutions $(x, f(x))$ can be generated efficiently, where x is sampled according to the uniform distribution from the function's domain.

[14] The work of Broadbent et al. [12] is on one-time programs. Their result on the composability of DQC is obtained by modifying their main one-time program protocol and security proof so that it corresponds to a variant of the DQC protocol from [1].

axiomatic, where lower abstraction levels inherit the definitions and theorems (e.g., a composition theorem) from the higher level, but the definition or concretization of low levels is not required for proving theorems at the higher levels. In particular, it is not hard-coded in the security notions of AC whether the underlying computation model is classical or quantum, and this framework can be used equally for both.

Even though these frameworks differ considerably in their approach, they all share the common notion that composable security is defined by the distance between the real world setting and an ideal setting in which the cryptographic task is accomplished in some perfect way. We use AC in this work, because it simplifies the security definitions by removing many notions which are not necessary at that level of abstraction. But the same results could have been proven using another framework, e.g., a quantum version of UC security [43].

1.4 Results

In this paper, we define a composable framework for analyzing the security of delegated quantum computing, using the aforementioned AC framework [29]. We model DQC in a generic way, which is independent of the computing requirements or universality of the protocol, and encompasses to the best of our knowledge all previous work on DQC. We then define composable blindness and composable verifiability in this framework. The security definitions are thus applicable to any DQC protocol fitting in our model.

We study the relations between local security criteria used in previous works [1,2,11,16,20,32,35] and composable security of DQC. We show that by strengthening the existing notion of local-verifiability, we can close the gap between these local criteria and composable security of DQC. To do this we introduce the notion of *independent* local-verifiability. Intuitively, this captures the idea that the acceptance probability of the client should not depend on the input or computation performed, but rather only on the activities of the (dishonest) server. Our main theorem is as follows.

Theorem 1. *If a DQC protocol implementing a unitary*[15] *transformation provides ε_{bl}-local-blindness and ε_{ind}-independent ε_{ver}-local-verifiability for all inputs $\psi_{A_C A_Q}$, where A_C is classical and A_Q is quantum, then it is δN^2-secure, where $\delta = 4\sqrt{2\varepsilon_{ver}} + 2\varepsilon_{bl} + 2\varepsilon_{ind}$ and $N = \dim \mathcal{H}_{A_Q}$.*

Note that by choosing the parameters such that δ is exponentially small in the size of the quantum input ($\log N$) negates the factor N^2 blow-up in the overall error (see also Remark 13).

Proving that a DQC protocol is secure then reduces to proving that these local criteria are satisfied.[16] For instance, the protocols of Morimae [32] and

[15] Any quantum operation can be written as a unitary on a larger system, effectively allowing this theorem to apply to all quantum operations, see Remark 12.

[16] This is similar in nature to the result on the composable security of quantum key distribution (QKD) [40], which shows that a QKD protocol that satisfies definitions

Fitzsimons and Kashefi [20] are shown to satisfy definitions of local-correctness, local-blindness and local-verifiability, equivalent to the ones considered here. To prove that these protocols are secure, it only remains to show that they also satisfy the stronger notion of *independent* local-verifiability introduced in this work, which we sketch in Sect. 6.1.

Finally, we analyze the security of two protocols — Broadbent, Fitzsimons and Kashefi [11] and Morimae and Fujii [35] — that do not provide any form of verifiability, so the generic reduction cannot be used. Instead we directly prove that both these protocols satisfy the definition of composable blindness, without verifiability (Theorems 14 and 15).

Interestingly — and somewhat unexpectedly — even though the local security definitions used in previous works are insufficient to guarantee composable security, the previously proposed protocols studied in this work are all still secure given the stronger security notions.

1.5 Structure of this Paper

In Sect. 2 we introduce two-party protocols and distance measures that we use in this work.[17] In Sect. 3 we explain delegated quantum computation, and model composable security for such protocols. In Sect. 4 we show that composable verifiability (which encompasses blindness) is equivalent to the distance between the real protocol and some ideal map that simultaneously provides both local-blindness and local-verifiability. This map is however still more elaborate than local criteria used in previous works. In Sect. 5 we break this map down into individual notions of local-blindness and independent local-verifiability, and prove that these are sufficient to achieve security. Finally, in Sect. 6 we look at the security of some existing protocols. We first discuss how our results can be applied to protocols that already provide local-verifiability. Then we prove that the DQC protocols of Broadbent, Kashefi and Fitzsimons [11] and Morimae and Fujii [35] are composably blind.

2 Quantum Systems

2.1 Two-Party Protocols

A two-party protocol can in be modeled by a sequence of CPTP maps $\pi_A = \{\mathcal{E}_i : \mathcal{L}(\mathcal{H}_{AC}) \to \mathcal{L}(\mathcal{H}_{AC})\}_i$ and $\pi_B = \{\mathcal{F}_i : \mathcal{L}(\mathcal{H}_{CB}) \to \mathcal{L}(\mathcal{H}_{CB})\}_i$, where A and

of *robustness*, *correctness* and *secrecy* is secure in a composable sense. These individual notions are all expressed with trace-distance-type criteria, e.g., a QKD protocol is ε-secret if $(1 - p_{\text{abort}})\|\rho_{KE} - \tau_K \otimes \rho_E\|_{\text{tr}} \leq \varepsilon$, where p_{abort} is the probability of aborting, ρ_{KE} the joint state of the final key and the eavesdropper's system and τ_K is the fully mixed state. To prove that a QKD protocol is secure, it is thus sufficient to prove that it satisfies these individual notions.

[17] These are an instantiation of the abstract systems defined in AC. We refer to the full version of this work [18, Sections 2] for an introduction to the AC framework, that is essential to understand the details of the current paper.

B are Alice and Bob's registers, and C represents a communication channel. Initially Alice and Bob place their inputs in their registers, and the channel C is in some fixed state $|0\rangle$. The players then apply successively their maps to their respective registers and the channel. For example, in the first round Alice applies $\mathcal{E}_1$ to the joint system AC, and sends C to Bob, who applies $\mathcal{F}_1$ to CB, and returns C to Alice. Then she applies $\mathcal{E}_2$, etc.

Such a sequence of maps, $\{\mathcal{E}_i : \mathcal{L}(\mathcal{H}_{AC}) \to \mathcal{L}(\mathcal{H}_{AC})\}_i$, has been called a *quantum strategy* by Gutoski and Watrous [24, 25] and a *quantum N-comb* by Chiribella, D'Ariano and Perinotti [17]. In particular, these authors derived independently a concise representation of combs/strategies in terms of the Choi-Jamiołkowski isomorphism. They also define the appropriate distance measure between two combs/strategies, corresponding to the optimal distinguishing advantage, which we sketch in the next section.

2.2 Distance Measures

The trace distance between two states ρ and σ is given by $D(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_{\mathrm{tr}}$, where $\|\cdot\|_{\mathrm{tr}}$ denotes the trace norm and is defined as $\|A\|_{\mathrm{tr}} := \operatorname{tr}\sqrt{A^\dagger A}$. If $D(\rho, \sigma) \leq \varepsilon$, we say that the two states are ε-close and often write $\rho \approx_\varepsilon \sigma$. This corresponds to the distinguishing advantage between two resources $\mathcal{R}$ and $\mathcal{S}$, which take no input and produce ρ and σ, respectively, as output: the probability of a distinguisher guessing correctly whether he holds $\mathcal{R}$ or $\mathcal{S}$ is exactly $\frac{1}{2} + \frac{1}{2}D(\rho, \sigma)$.

Another common metric which corresponds to the distinguishing advantage between resources of a certain type is the diamond norm. If the resources $\mathcal{R}$ and $\mathcal{S}$ take an input $\rho \in \mathcal{S}(\mathcal{H}_A)$ and produce an output $\sigma \in \mathcal{S}(\mathcal{H}_B)$, the distinguishing advantage between these resources is the diamond distance between the correspond maps $\mathcal{E}, \mathcal{F} : \mathcal{L}(\mathcal{H}_A) \to \mathcal{L}(\mathcal{H}_B)$. A distinguisher can generate a state ρ_{AR}, input the A part to the resource, and try to distinguish between the resulting states $\mathcal{E}(\rho_{AR})$ and $\mathcal{F}(\rho_{AR})$. We have $d(\mathcal{R}, \mathcal{S}) = \diamond(\mathcal{E}, \mathcal{F}) = \frac{1}{2}\|\mathcal{E} - \mathcal{F}\|_\diamond$, where

$$\|\Phi\|_\diamond := \max\{\|(\Phi \otimes \mathrm{id}_R)(\rho)\|_{\mathrm{tr}} : \rho \in \mathcal{S}(\mathcal{H}_{AR})\}$$

is the diamond norm. Note that the maximum of the diamond norm can always be achieved for a system R with $\dim \mathcal{H}_R = \dim \mathcal{H}_A$. Here too, we sometimes write $\mathcal{E} \approx_\varepsilon \mathcal{F}$ if two maps are ε-close.

If the resources considered are halves of two player protocols, say π_i or π_j, the above reasoning can be generalized for obtaining the distinguishing advantage. The distinguisher can first generate an initial state $\rho \in \mathcal{S}(\mathcal{H}_{AR})$ — which for convenience we define as a map on no input $\rho := \mathcal{D}_0()$ — and input the A part of the state into the resource. It receives some output ρ_{CR} from the resource, can apply some arbitrary map $\mathcal{D}_1 : \mathcal{L}(\mathcal{H}_{CR}) \to \mathcal{L}(\mathcal{H}_{CR})$ to the state, and input the C part of the new state in the resource. Let it repeat this procedure with different maps $\mathcal{D}_i$ until the end of the protocol, after which it holds one of two states: φ_{AR} if it had access to π_i and ψ_{AR} if it had access to π_j. The trace distance $D(\varphi_{AR}, \psi_{AR})$ defines the advantage the distinguisher has of correctly guessing whether it was interacting with π_i or π_j, and by maximizing this over all possible initial inputs $\rho_{AR} = \mathcal{D}_0()$, and all subsequent maps $\{\mathcal{D}_i : \mathcal{L}(\mathcal{H}_{CR}) \rightarrow \mathcal{L}(\mathcal{H}_{CR})\}_i$, the distinguishing advantage between these resources becomes

$$d(\pi_i, \pi_j) = \max_{\{\mathcal{D}_i\}_i} D(\varphi_{AR}, \psi_{AR}). \tag{1}$$

This has been studied by both Gutoski [24] and Chiribella et al. [17], and we refer to their work for more details.

3 Delegated Quantum Computation

In the (two-party) delegated quantum computation (DQC) model, Alice asks a server, Bob, to execute some quantum computation for her. Intuitively, Alice plays the role of a client, and Bob the part of a computationally more powerful server. Alice has several security concerns. She wants the protocol to be blind, that is, she wants the server to execute the quantum computation without learning anything about the input other than what is unavoidable, e.g., an upper bound on its size, and possibly whether the output is classical or quantum. She may also want to know if the result sent to her by Bob is correct, which we refer to as verifiability.

In Sect. 3.1 we model the ideal resource that a DQC protocol constructs and the structure of a generic DQC protocol. And in Sect. 3.2 we give the corresponding security definitions. This section uses the AC cryptography nomenclature, which is explained in detail in the full version [18].

3.1 DQC Model

Ideal Resource. To model the security (and correctness) of a delegated quantum computation protocol, we need to model the ideal delegated computation resource $\mathcal{S}$ that we wish to build. We start with an ideal resource that provides blindness, and denote it $\mathcal{S}^{\text{blind}}$.

The task Alice wants to be executed is provided as an input to the resource $\mathcal{S}^{\text{blind}}$ at the A-interface. It could be modeled as having two parts, some quantum state ψ_{A_1} and a classical description Φ_{A_2} of some quantum operation that she wants to apply to ψ, i.e., she wishes to compute $\Phi(\psi)$. This can alternatively be seen as applying a universal computation $\mathcal{U}$ to the input $\psi_{A_1} \otimes |\Phi\rangle\langle\Phi|_{A_2}$. We adopt this view in the remainder of this paper, and model the resource as performing some fixed computation $\mathcal{U}$ on an input ψ_A that may be part quantum and part classical.[18]

[18] Alternatively, the input can be modeled as entirely quantum, and both Alice and the ideal resource first measure the part of the input that should be classical, before executing π_A and the universal computation $\mathcal{U}$, respectively. This corresponds to plugging an extra measurement converter into the A-interfaces of both the real and ideal systems (that converts the quantum input into a classical-quantum input), which can only decrease the distance between the real and ideal systems, i.e., increase the security.

Any DQC protocol must reveal to the server an upper bound on the size of the computation it is required to execute. Other information might also be made intentionally available, such as whether the output of the computation is classical or quantum. Although one could imagine a generic DQC model in which these "permitted leaks" are entangled with the rest of the input, we restrict our considerations to classical information, i.e., a subsystem of the input ψ_A is classical[18] and contains a string $\ell^{\psi_A} \in \{0,1\}^*$ that is copied and provided to the server Bob at the start of the protocol, so that he may set up the required resources and programs for the computation. Alternatively, this string can be taken to be some fixed publicly available information, not modeled explicitly. We do so in the following sections to simplify the notation, but prefer make it explicit in this section so as not to hide the fact that some information about the input is always given to the server.

The ideal resource $\mathcal{S}^{\text{blind}}$ thus takes this input ψ_A at its A-interface, and, if Bob does not activate his filtered functionalities — which can be modeled by a bit b, set to 0 by default, and which a simulator σ_B can flip to 1 to signify that it is activating the cheating interface — $\mathcal{S}^{\text{blind}}$ outputs $\mathcal{U}(\psi_A)$. This ensures both correctness and universality (in the case where $\mathcal{U}$ is a universal computation). Alternatively, $\mathcal{S}^{\text{blind}}$ can be restricted to work for inputs corresponding to a certain class of computational problems, if we desire a construction only designed for such a class.

If the cheating B-interface is activated, the ideal resource outputs a copy of the string ℓ^{ψ_A} at this interface. Bob also has another filtered functionality, one which allows him to tamper with the final output. The most general operation he could perform is to give $\mathcal{S}^{\text{blind}}$ a quantum state ψ_B — which could be entangled with Alice's input ψ_A — along with the description of some map $\mathcal{E} : \mathcal{L}(\mathcal{H}_{AB}) \to \mathcal{L}(\mathcal{H}_A)$, and ask it to output $\mathcal{E}(\psi_{AB})$ at Alice's interface. Since $\mathcal{S}^{\text{blind}}$ only captures blindness, but says nothing about Bob's ability to manipulate the final output, we define it to perform this operation and output any $\mathcal{E}(\psi_{AB})$ at Bob's request. This is depicted in Fig. 1 with the filtered functionalities in gray.

Definition 2. *The ideal DQC resource $\mathcal{S}^{blind}$ which provides both correctness and blindness takes an input ψ_A at Alice's interface, but no honest input at Bob's interface. Bob's filtered interface has a control bit b, set by default to 0, which he can flip to activate the other filtered functionalities. The resource $\mathcal{S}^{blind}$ then outputs the permitted leak ℓ^{ψ_A} at Bob's interface, and accepts two further inputs, a state ψ_B and map description $|\mathcal{E}\rangle\langle\mathcal{E}|$. If $b = 0$, it outputs the correct result $\mathcal{U}(\psi_A)$ at Alice's interface; otherwise it outputs Bob's choice, $\mathcal{E}(\psi_{AB})$.*

A DQC protocol is verifiable if it provides Alice with a mechanism to detect a cheating Bob and output an error flag `err` instead of some incorrect computation. This is modeled by weakening Bob's filtered functionality: an ideal DQC resource with verifiability, $\mathcal{S}^{\text{blind}}_{\text{verif}}$, only allows Bob to input one classical bit c, which specifies whether the output should be $\mathcal{U}(\psi_A)$ or some error state $|\text{err}\rangle$, which by construction is orthogonal to the space of valid outputs. The ideal resource thus never outputs a wrong computation. This is illustrated in Fig. 2.

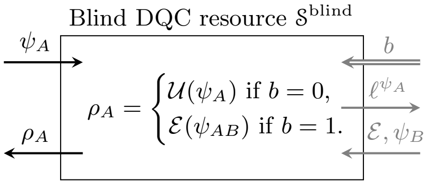
Blind DQC resource $\mathcal{S}^{\text{blind}}$
ψ_A
ρ_A
$\rho_A = \begin{cases} \mathcal{U}(\psi_A) \text{ if } b = 0, \\ \mathcal{E}(\psi_{AB}) \text{ if } b = 1. \end{cases}$
b
ℓ^{ψ_A}
$\mathcal{E}, \psi_B$

Fig. 1. An ideal DQC resources. The client Alice has access to the left interface, and the server Bob to the right interface. The double-lined input flips a bit set by default to 0. The functionalities provided at Bob's interface are grayed to signify that they are accessible only to a cheating server. If Bob is honest, this interface is obstructed by a filter, which we denote by $\perp_B$ in the following. $\mathcal{S}^{\text{blind}}$ provides blindness — it only leaks the permitted information at Bob's interface — but allows Bob to choose Alice's output.

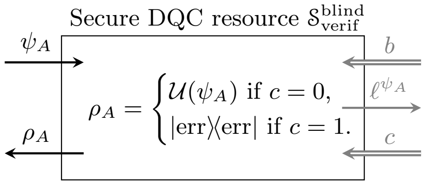
Secure DQC resource $\mathcal{S}^{\text{blind}}_{\text{verif}}$
ψ_A
ρ_A
$\rho_A = \begin{cases} \mathcal{U}(\psi_A) \text{ if } c = 0, \\ |\text{err}\rangle\langle\text{err}| \text{ if } c = 1. \end{cases}$
b
ℓ^{ψ_A}
c

Fig. 2. Another ideal DQC resources. $\mathcal{S}^{\text{blind}}_{\text{verif}}$ provides both blindness and verifiability — in addition to leaking only the permitted information, it never outputs an erroneous computation result.

Definition 3. *The ideal DQC resource $\mathcal{S}^{blind}_{verif}$ which provides correctness, blindness and verifiability takes an input ψ_A at Alice's interface, and two filtered control bits b and c (set by default to 0). If $b = 0$, it simply outputs $\mathcal{U}(\psi_A)$ at Alice's interface. If $b = 1$, it outputs the permitted leak ℓ^{ψ_A} at Bob's interface, then reads the bit c, and conditioned on its value, it either outputs $\mathcal{U}(\psi_A)$ or $|err\rangle$ at Alice's interface.*

Concrete Setting. In the concrete (or real) setting, the only resource that Alice and Bob need is a (two-way) communication channel $\mathcal{R}$. Alice's protocol π_A receives ψ_A as an input on its outside interface. It then communicates through $\mathcal{R}$ with Bob's protocol π_B, and produces some final output ρ_A. For the sake of generality we assume that the operations performed by π_A and π_B, and the communication between them, are all quantum. Of course, a protocol is only useful if Alice has very few quantum operations to perform, and most of the communication is classical. However, to model security, it is more convenient to consider the most general case possible, so that it applies to all possible protocols.

As described in Sect. 2.1, their protocols can be modeled by a sequence of CPTP maps $\{\mathcal{E}_i : \mathcal{L}(\mathcal{H}_{AC}) \to \mathcal{L}(\mathcal{H}_{AC})\}_{i=1}^N$ and $\{\mathcal{F}_i : \mathcal{L}(\mathcal{H}_{CB}) \to \mathcal{L}(\mathcal{H}_{CB})\}_{i=1}^{N-1}$. We illustrate a run of such a protocol in Fig. 3. The entire system consisting of the protocol (π_A, π_B) and the channel $\mathcal{R}$ is a map which transforms ψ_A into ρ_A.

If both players played honestly and the protocol is correct, this should result in $\rho_A = \mathcal{U}(\psi_A)$.

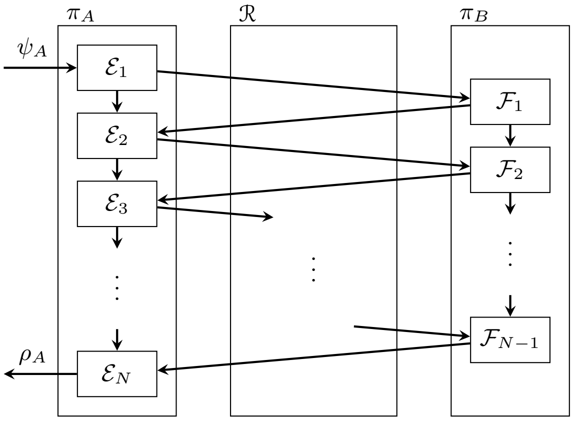

Fig. 3. A generic run of a DQC protocol. Alice has access to the left interface and Bob to the right interface. The entire system builds one CPTP operation which maps ψ_A to ρ_A.

In the following, when we refer to a DQC protocol, we simply mean any protocol satisfying the model of Fig. 3. Whether the protocol actually performs delegated quantum computation depends on whether it satisfies the correctness condition, which we define in Sect. 3.2.

3.2 Security of DQC

Applying the AC security definition (see the full version, [18, Definition 2.1]) to the DQC model from the previous section, we get that a protocol π constructs a blind quantum computation resource $\mathcal{S}^{\text{blind}}$ from a communication channel $\mathcal{R}$ within ε if there exists a simulator σ_B such that

$$\pi_A \mathcal{R} \pi_B \approx_\varepsilon \mathcal{S}^{\text{blind}} \bot_B \qquad \text{and} \qquad \pi_A \mathcal{R} \approx_\varepsilon \mathcal{S}^{\text{blind}} \sigma_B, \tag{2}$$

where $\bot_B$ is a filter which obstructs Bob's cheating interface.[19] The fist condition in (2) captures the correctness of the protocol, and we say that a protocol provides *ε-correctness* if this condition is fulfilled. The second condition, which we illustrate in Fig. 4, measures the security. If it is fulfilled, we have *ε-blindness.* If $\varepsilon = 0$ we say that we have *perfect blindness.*

[19] These equations are to be interpreted graphically: π_A is plugged into the left interface of $\mathcal{R}$, and π_B is plugged into the right interface, see the illustrations in Figs. 3 and 4 or the full version [18] for further explanations.

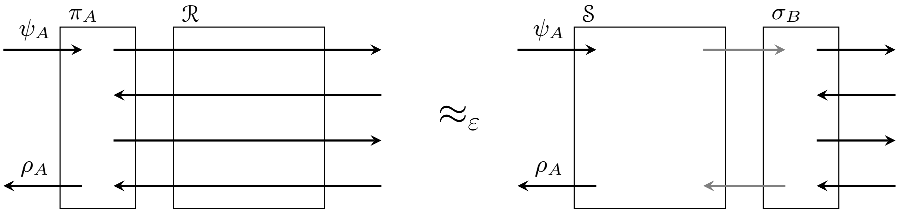

Fig. 4. An illustration of the second terms of (2) and (3). If a distinguisher cannot guess with advantage greater than ε whether it is interacting with the real construct on the left or the ideal construct on the right, the two are ε-close and the protocol ε-secure against a cheating Bob.

Likewise in the case of verifiability, the ideal resource $\mathcal{S}^{\text{blind}}_{\text{verif}}$ is constructed by π from $\mathcal{R}$ if there exists a simulator σ_B such that,

$$\pi_A \mathcal{R} \pi_B \approx_\varepsilon \mathcal{S}^{\text{blind}}_{\text{verif}} \perp_B \qquad \text{and} \qquad \pi_A \mathcal{R} \approx_\varepsilon \mathcal{S}^{\text{blind}}_{\text{verif}} \sigma_B . \tag{3}$$

The first condition from (3) is identical to the first condition of (2), and captures ε-correctness. The second condition in (3) (also illustrated by Fig. 4) guarantees both blindness and verifiability, and if it is satisfied we say that the we have *ε-blind-verifiability.*

Note that the exact metrics used to distinguish between the resources from (2) and (3) are defined in Sect. 2.2. $\pi_A \mathcal{R} \pi_B$ and $\mathcal{S} \perp_B$ — as can be seen from their depictions in Figs. 3 and 2 (with a filter blocking the cheating interface of the latter) — are resources which implement a single map, so the diamond distance corresponds to the distinguishing advantage. $\pi_A \mathcal{R}$ and $\mathcal{S} \sigma_B$ are half of two-party protocols, so the distinguishing metric corresponds to the distance between quantum strategies/combs introduced by Gutoski and Watrous [24, 25] and Chiribella et al. [17], and described in Sect. 2.2.

4 Blind and Verifiable DQC

Finding a simulator to prove the security of a protocol can be challenging. In this section we reduce the task of proving that a DQC protocol constructs the ideal resource $\mathcal{S}^{\text{blind}}_{\text{verif}}$ to proving that the map implemented by the protocol is close to some ideal map that intuitively provides some form of local-blindness-and-verifiability. The converse also holds: any protocol which constructs $\mathcal{S}^{\text{blind}}_{\text{verif}}$ must be close to this ideal map.

A malicious server Bob will not apply the CPTP maps assigned to him by the protocol, but his own set of cheating maps $\{\mathcal{F}_i : \mathcal{L}(\mathcal{H}_{CB}) \to \mathcal{L}(\mathcal{H}_{CB})\}_{i=1}^{N-1}$. Furthermore, he might hold (the B part of) a purification of Alice's input, ψ_{ABR}. Intuitively, a protocol provides local-blindness[20] if the final state held by Bob

[20] We provide formal definitions of local-blindness and local-verifiability in Sect. 5.

could have been generated by a local map on his system — say, $\mathcal{F}$ — independently from Alice's input, but which naturally depends on his behavior given by the maps $\{\mathcal{F}_i\}_i$. It provides local-verifiability[20] if the final state held by Alice is either the correct outcome or some error flag. Combining the two gives an ideal map of the from $\mathcal{U} \otimes \mathcal{F}^{\mathrm{ok}} + \mathcal{E}^{\mathrm{err}} \otimes \mathcal{F}^{\mathrm{err}}$, where $\mathcal{F}^{\mathrm{ok}}$ and $\mathcal{F}^{\mathrm{err}}$ break $\mathcal{F}$ down in two maps which result in the correct outcome and an error flag, respectively.

Definition 4 (local-blind-verifiability). *We say that a DQC protocol provides ε-local-blind-verifiability, if, for all adversarial behaviors $\{\mathcal{F}_i\}_i$, there exist two completely positive, trace non-increasing maps $\mathcal{F}_B^{ok}$ and $\mathcal{F}_B^{err}$, such that*

$$\mathcal{P}_{AB} \approx_\varepsilon \mathcal{U}_A \otimes \mathcal{F}_B^{ok} + \mathcal{E}_A^{err} \otimes \mathcal{F}_B^{err}, \tag{4}$$

where $\mathcal{P}_{AB} : \mathcal{L}(\mathcal{H}_{AB}) \to \mathcal{L}(\mathcal{H}_{AB})$ is the map corresponding to a protocol run with Alice behaving honestly and Bob using his cheating operations $\{\mathcal{F}_i\}_i$, and $\mathcal{E}_A^{err}$ discards the A system and produces an error flag $|err\rangle\langle err|$ orthogonal to all possible valid outputs. We say that the protocol provides ε-local-blind-verifiability for a set of initial states $\mathcal{B}$, if (4) *holds when applied to these states, i.e., for all $\psi_{ABR} \in \mathcal{B}$,*

$$\mathcal{P}_{AB}(\psi_{ABR}) \approx_\varepsilon \big(\mathcal{U}_A \otimes \mathcal{F}_B^{ok} + \mathcal{E}_A^{err} \otimes \mathcal{F}_B^{err}\big)(\psi_{ABR}).$$

Remark 5. For simplicity, this definition assumes the allowed leaks (e.g., input size, computation size) to be fixed, and applies to all protocols $\mathcal{P}_{AB}$ tailored for inputs with an identical leak (e.g., identical size). These leaks could be explicitly modeled by allowing the maps $\mathcal{F}_B^{\mathrm{ok}}$ and $\mathcal{F}_B^{\mathrm{err}}$ to depend on them.

We now state the main theorem of this section, namely that it is both necessary and sufficient for a DQC protocol to satisfy Definition 4 to be blind-verifiable, i.e., to satisfy the second condition of Equation (3). A proof is is given in the full version [18]. In order to construct $\mathcal{S}_{\mathrm{verif}}^{\mathrm{blind}}$, a DQC protocol also needs to be ε-correct, that is, satisfy the first condition from Equation (3). We show in Appendix A that this is fulfilled, if, when Bob behaves honestly, Equation (4) is satisfied for $\mathcal{F}_B^{\mathrm{ok}} = \mathrm{id}_B$ and $\mathcal{F}_B^{\mathrm{err}} = 0$.

Theorem 6. *Any DQC protocol which provides ε-local-blind-verifiability is 2ε-blind-verifiable. And any DQC protocol which is ε-blind-verifiable provides ε-local-blind-verifiability.*

5 Reduction to Local Criteria

Although the notion of local-blind-verifiability defined in the previous section captures the security of DQC in a single equation, it is still more elaborate than existing definitions found in the literature, that treat blindness and verifiability separately.

In this section we provide separate definitions for these local notions, and strengthen local-verifiability by requiring that the server Bob be able to infer on his own whether the client Alice will reject his response — learning whether Alice did reject will then not provide him with any information that he could not obtain on his own. We then show that in the case where Bob does not hold a state entangled with the input (e.g., when the input is entirely classical), these notions are sufficient to obtain local-blind-verifiability with a similar error parameter. In the case where Bob's system is entangled to Alice's input, we show that the same holds, albeit with an error increased by a factor $\left(\dim \mathcal{H}_{A_Q}\right)^2$, where A_Q is the subsystem of Alice's input which is quantum.

This can be used to show that the protocol of Fitzsimons and Kashefi [20] and Morimae [32], which have already been analyzed using (insufficient) local criteria, are secure. We provide a proof sketch of the missing steps for both these protocols in the full version of this paper [18].

Local-blindness can be seen as a simplification of local-blind-verifiability, in which we ignore Alice's outcome and only check that Bob's system could have been generated locally, i.e., is independent from Alice's input (and output).

Definition 7 (Local-blindness). *A DQC protocol provides ε-local-blindness, if, for all adversarial behaviors $\{\mathcal{F}_i\}_i$, there exists a CPTP map $\mathcal{F} : \mathcal{L}(\mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_B)$ such that*

$$\mathrm{tr}_A \circ \mathcal{P}_{AB} \approx_\varepsilon \mathcal{F} \circ \mathrm{tr}_A, \tag{5}$$

where $\circ$ is the composition of maps, tr_A the operator that trace out the A-system, and $\mathcal{P}_{AB} : \mathcal{L}(\mathcal{H}_{AB}) \to \mathcal{L}(\mathcal{H}_{AB})$ is the map corresponding to a protocol run with Alice behaving honestly and Bob using his cheating operations $\{\mathcal{F}_i\}_i$. We say that the protocol provides ε-local-blindness for a set of initial states $\mathcal{B}$, if (5) holds when applied to these states, i.e., for all $\psi_{ABR} \in \mathcal{B}$,

$$\mathrm{tr}_A \circ \mathcal{P}_{AB}(\psi_{ABR}) \approx_\varepsilon \mathcal{F} \circ \mathrm{tr}_A(\psi_{ABR}).$$

Likewise, local-verifiability can also be seen as a simplification of local-blind-verifiability, in which we ignore Bob's system and only check that Alice holds either the correct outcome or an error flag $|\mathrm{err}\rangle$, which by construction is orthogonal to any possible valid output. In the following we define local-verifiability only for the case where Bob's system is not entangled to Alice's input, since otherwise the correct outcome depends on Bob's actions, and cannot be modeled by describing Alice's system alone.[21]

Definition 8 (Local-verifiability). *A DQC protocol provides ε-local-verifiability, if, for all adversarial behaviors $\{\mathcal{F}_i\}_i$ and all initial states $\psi_{AR_1} \otimes \psi_{R_2B}$, there exists a $0 \leq p^\psi \leq 1$ such that*

$$\rho^\psi_{AR_1} \approx_\varepsilon p^\psi(\mathcal{U} \otimes \mathrm{id}_{R_1})(\psi_{AR_1}) + (1-p^\psi)|err\rangle\langle err| \otimes \psi_{R_1}, \tag{6}$$

where $\rho^\psi_{AR_1}$ is the final state of Alice and the first part of the reference system. We say that the protocol provides ε-local-verifiability for a set $\mathcal{B}$ of initial states in product form, if (6) holds for all $\psi_{AR_1} \otimes \psi_{R_2B} \in \mathcal{B}$.

[21] The resulting definition is equivalent to that of [20] and non-composable authentication definitions [5], which bound the probability of projecting the outcome on the space of invalid results.

As mentioned in Sect. 1, local-blindness and local-verifiability together do not provide the security guarantees one expects from DQC. This seems to be because the verification procedure can depend on the input (as in the example from Footnote 8), and thus if Bob learns the result of this measurement, he learns something about the input. This motivates us to define a stronger notion, in which Bob can reconstruct on his own whether the output will be accepted—the outcome of Alice's verification procedure must thus be independent of her input. To do this, we introduce a new qubit in a system $\bar{B}$, which contains a copy of the information whether Alice accepts or rejects, i.e., for a final state

$$\rho^{\psi}_{ARB} = \phi^{\text{ok}}_{ARB} + |\text{err}\rangle\langle\text{err}| \otimes \phi^{\text{err}}_{RB}, \tag{7}$$

we define

$$\rho^{\psi}_{ARB\bar{B}} \coloneqq \phi^{\text{ok}}_{ARB} \otimes |\text{ok}\rangle\langle\text{ok}| + |\text{err}\rangle\langle\text{err}| \otimes \phi^{\text{err}}_{RB} \otimes |\text{err}\rangle\langle\text{err}|. \tag{8}$$

Note that (8) can be generated from (7) by introducing a system $\bar{B}$ in the state $|\text{ok}\rangle$ and changing its value to $|\text{err}\rangle$ conditioned on A being in the state $|\text{err}\rangle$. Let $\mathcal{Q}_{A\bar{B}} : \mathcal{L}(\mathcal{H}_A) \to \mathcal{L}(\mathcal{H}_{A\bar{B}})$ be such an operation, i.e., $\rho^{\psi}_{ARB\bar{B}} = \mathcal{Q}_{A\bar{B}}(\rho^{\psi}_{ARB})$. Equation (7) can then be recovered from (8) by tracing out the system $\bar{B}$.

The notion of verifiability is strengthened by additionally requiring that leaking this system $\bar{B}$ to the adversary does not provide him with more information about the input, i.e., Bob could (using alternative maps) generate the system $\bar{B}$ on his own.

Definition 9. *A DQC protocol provides $\bar{\varepsilon}$-independent ε-local-verifiability, if, in addition to providing ε-local-verifiability, for all adversarial behaviors $\{\mathcal{F}_i : \mathcal{L}(\mathcal{H}_{CB}) \to \mathcal{L}(\mathcal{H}_{CB})\}_i$ there exist alternative maps $\{\mathcal{F}'_i : \mathcal{L}(\mathcal{H}_{CB\bar{B}}) \to \mathcal{L}(\mathcal{H}_{CB\bar{B}})\}_i$ (for an initially empty system $\bar{B}$), such that*

$$\operatorname{tr}_A \circ \mathcal{Q}_{A\bar{B}} \circ \mathcal{P}_{AB} \approx_{\bar{\varepsilon}} \operatorname{tr}_A \circ \mathcal{P}'_{AB\bar{B}}, \tag{9}$$

where $\circ$ is the composition of maps, $\mathcal{P}_{AB} : \mathcal{L}(\mathcal{H}_{AB}) \to \mathcal{L}(\mathcal{H}_{AB})$ and $\mathcal{P}'_{AB\bar{B}} : \mathcal{L}(\mathcal{H}_{AB}) \to \mathcal{L}(\mathcal{H}_{AB\bar{B}})$ are the maps corresponding to runs of the protocol with Alice being honest and Bob using maps $\{\mathcal{F}_i\}_i$ and $\{\mathcal{F}'_i\}_i$ respectively, and $\mathcal{Q}_{A\bar{B}} : \mathcal{L}(\mathcal{H}_A) \to \mathcal{L}(\mathcal{H}_{A\bar{B}})$ is a map which generates from A a system $\bar{B}$ holding a copy of the information whether Alice accepts or rejects. We say that a protocol provides $\bar{\varepsilon}$-independent ε-local-verifiability for a set of initial states $\mathcal{B}$, if the same conditions hold for all states in $\mathcal{B}$, i.e., if we have ε-local-verifiability for $\mathcal{B}$, and if for all $\psi_{ABR} \in \mathcal{B}$,

$$\operatorname{tr}_A \circ \mathcal{Q}_{A\bar{B}} \circ \mathcal{P}_{AB}(\psi_{ABR}) \approx_{\bar{\varepsilon}} \operatorname{tr}_A \circ \mathcal{P}'_{AB\bar{B}}(\psi_{ABR}).$$

Remark 10. By the triangle inequality, if a protocol provides both ε-local-blindness and $\bar{\varepsilon}$-independent ε'-local-verifiability, then there exists a map $\mathcal{F}' : \mathcal{L}(\mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_{B\bar{B}})$ such that

$$\operatorname{tr}_A \circ \mathcal{Q}_{A\bar{B}} \circ \mathcal{P}_{AB} \approx_{\varepsilon+\bar{\varepsilon}} \mathcal{F}' \circ \operatorname{tr}_A . \tag{10}$$

We are now ready to state the main theorem, namely that the above local definitions are sufficient to achieve composable security.

Theorem 11 (Theorem 1 restated). *If a DQC protocol implementing a unitary transformation provides ε_{bl}-local-blindness and ε_{ind}-independent ε_{ver}-local-verifiability for all inputs $\psi_{A_C A_Q}$, where A_C is classical and A_Q is quantum, then it is δN^2-blind-verifiable, where $\delta = 4\sqrt{2\varepsilon_{ver}} + 2\varepsilon_{bl} + 2\varepsilon_{ind}$ and $N = \dim \mathcal{H}_{A_Q}$. If additionally it provides ε_{cor}-local-correctness,*[22] *it constructs $\mathcal{S}^{blind}_{verif}$ from a communication channel within $\varepsilon = \max\{\delta N^2, \varepsilon_{cor}\}$.*

Independent local-verifiability makes a statement about Alice's system at the end of the protocol — it is either in the correct state or contains an error flag. Local-blindness makes a statement about Bob's system at the end of the protocol — it contains no information about the input. To prove Theorem 11, we need to combine these two definitions to make a statement about the joint system of Alice and Bob at the end of the protocol, equivalent to local-blind-verifiability (Definition 4). The result then follows from Theorem 6.

The main idea of the proof is to show that, in the case of an input in product form between Alice and Bob, Uhlmann's theorem can be used to extend the statement about Alice's system being close to ideal to a joint AB system. We then show that if the input is entangled between Alice and Bob, the error can increase at most by a multiplicative factor of N^2. A complete proof is given in the full version [18].

Remark 12. This theorem only hold for protocols that construct a DQC resource for which the implemented operation $\mathcal{U}$ is unitary. Since any quantum operation can be written as a unitary on a larger system [38], this effectively allows the theorems to apply to any CPTP operation $\mathcal{E}$ as long as the necessary qubits for the unitary implementation are appended to the in- and outputs. For example, instead of defining universal computation as a unitary, most papers — e.g., [11, 20, 32, 35] — describe how to perform any (arbitrary) unitary operation U_x on any arbitrary input ρ_{in}. By appending the description x of the unitary U_x to the input and output, this is equivalent to applying the unitary transformation $\mathcal{U} := \sum_x U_x \otimes |x\rangle\langle x|$ to the input $\rho_{\text{in}} \otimes |x\rangle\langle x|$.

Remark 13. If the input is entirely classical (e.g., the client wants to factor a number), the failure ε is polynomial in the error parameters of the different local criteria, and the reduction is tight. If the input is quantum, the failure is multiplied by the dimension squared of the quantum (sub)system, and the errors of the local criteria need to be exponentially small in the size of the quantum input to compensate.

[22] See Definition 16 in Appendix A.

6 Existing Protocols

6.1 Applying the Security Reduction

The definitions of local-blindness and local-verifiability used in this work are equivalent to those used to prove local-security for most protocols in the literature, e.g., by Fitzsimons and Kashefi [20] and Morimae [32]. To prove that such protocols are secure, it remains to show that they satisfy the stronger definition of *independent* local-verifiability introduced in this work. We sketch in this section that this is the case for [20] and [32], and refer to the full version [18] for a longer discussion.

Both these works achieve local-verifiability by introducing randomly positioned *trap qubits* in the protocol: these are states which are independent of Alice's input, and for which she knows the outcome of the operation that the server, Bob, should perform. If the server does not trigger any of the traps, then with high probability he is running the correct program [20, 32].

This technique used to achieve local-verifiability also provides independent local-verifiability, because the position of the traps and whether they get trigged are independent of the input. Thus, Bob could run the protocol on his own — without knowing Alice's input and choosing himself the position of the trap qubits — and would end up holding exactly the same bit as Alice that decides if the output is accepted or rejected.

6.2 Blindness

We present in this section the security results for two different DQC protocols proposed in the literature: we show that they construct the ideal blind quantum computation resource $\mathcal{S}^{\text{blind}}$ defined in Definition 2. The protocols and proofs appear in the full version [18], we only give a brief overview here. Note that since these protocols do not provide verifiability, we cannot use the generic results from Sect. 5 to prove that they are blind.

In the DQC protocol of Broadbent, Fitzsimons and Kashefi [11], Alice hides the computation by encrypting all the communication with a one-time pad. The main idea of the security proof is for the simulator to replace the encrypted states sent to the distinguisher by halves of EPR pairs. It then forwards the other halves to the ideal DQC resource, which gate teleports the real inputs. The distinguisher is then oblivious to whether it is interacting with the real protocol or the ideal resource and simulator.

Theorem 14. *The DQC protocol of Broadbent, Fitzsimons and Kashefi [11] provides perfect blindness.*

Morimae and Fujii [34] proposed a DQC protocol with one-way communication from Bob to Alice, in which Alice simply measures each qubit she receives, one at a time. We show that the general class of protocols with one-way communication is perfectly blind.

Theorem 15. *Any DQC protocol π with one-way communication from Bob to Alice provides perfect blindness.*

A Correctness

Intuitively, a protocol is correct if, when Bob behaves honestly, Alice ends up with the correct output. This must also hold with respect to a purification of the input.

Definition 16. *A DQC protocol provides ε-local-correctness, if, when both parties behave honestly, for all initial states ψ_{AR}, the map implemented by the protocol on Alice's input, $\mathcal{P}_A : \mathcal{L}(\mathcal{H}_A) \to \mathcal{L}(\mathcal{H}_A)$ is*

$$\mathcal{P}_A \approx_\varepsilon \mathcal{U}. \tag{11}$$

It is straightforward, that this is equivalent to the composable notion defined in Equations (2) and (3) in Sect. 3.2.

Lemma 17. *A DQC protocol which provides ε-local-correctness is also ε-correct.*

A proof is given in the full version [18].

Acknowledgments

This material is based on research supported in part by the Singapore National Research Foundation under NRF Award No. NRF-NRFF2013-01. VD acknowledges the support of the EPSRC Doctoral Prize Fellowship. Initial part of this work was performed while VD was at Heriot-Watt University, Edinburgh, supported by EPSRC (grant EP/E059600/1). CP and RR are supported by the Swiss National Science Foundation (via grant No. 200020-135048 and the National Centre of Competence in Research 'Quantum Science and Technology') and the European Research Council – ERC (grant No. 258932).

References

1. Aharonov, D., Ben-Or, M., Eban, E.: Interactive proofs for quantum computations. In: Proceedings of Innovations in Computer Science, ICS 2010. pp. 453–469 (2010)
2. Arrighi, P., Salvail, L.: Blind quantum computation. International Journal of Quantum Information 4(05), 883–898 (2006)
3. Backes, M., Pfitzmann, B., Waidner, M.: A general composition theorem for secure reactive systems. In: Theory of Cryptography, Proceedings of TCC 2004. Lecture Notes in Computer Science, vol. 2951, pp. 336–354. Springer (2004)
4. Backes, M., Pfitzmann, B., Waidner, M.: The reactive simulatability (RSIM) framework for asynchronous systems. Information and Computation 205(12), 1685–1720 (2007), extended version of [39]
5. Barnum, H., Crépeau, C., Gottesman, D., Smith, A., Tapp, A.: Authentication of quantum messages. In: Proceedings of the 43rd Symposium on Foundations of Computer Science, FOCS '02. pp. 449–458. IEEE (2002)
6. Barrett, J., Colbeck, R., Kent, A.: Memory attacks on device-independent quantum cryptography. Physical Review Letters 110, 010503 (Jan 2013)

7. Barz, S., Fitzsimons, J.F., Kashefi, E., Walther, P.: Experimental verification of quantum computation. Nature Physics (2013)
8. Barz, S., Kashefi, E., Broadbent, A., Fitzsimons, J.F., Zeilinger, A., Walther, P.: Demonstration of blind quantum computing. Science 335(6066), 303–308 (Jan 2012)
9. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Advances in Cryptology – ASIACRYPT 2000. Lecture Notes in Computer Science, vol. 1976, pp. 531–545. Springer (2000)
10. Ben-Or, M., Mayers, D.: General security definition and composability for quantum & classical protocols (2004), http://www.arxiv.org/abs/quant-ph/0409062, eprint
11. Broadbent, A., Fitzsimons, J., Kashefi, E.: Universal blind quantum computation. In: Proceedings of the 50th Symposium on Foundations of Computer Science, FOCS '09. pp. 517–526. IEEE Computer Society (2009)
12. Broadbent, A., Gutoski, G., Stebila, D.: Quantum one-time programs. In: Advances in Cryptology – CRYPTO 2013. Lecture Notes in Computer Science, vol. 8043, pp. 344–360. Springer (2013)
13. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Proceedings of the 42nd Symposium on Foundations of Computer Science, FOCS '01. pp. 136–145. IEEE (2001)
14. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067 (2013), http://eprint.iacr.org/2000/067, updated version of [13]
15. Chien, C.H., Meter, R.V., Kuo, S.Y.: Fault-tolerant operations for universal blind quantum computation (2013), http://www.arxiv.org/abs/1306.3664, eprint
16. Childs, A.M.: Secure assisted quantum computation. Quantum Information & Computation 5(6), 456–466 (2005)
17. Chiribella, G., D'Ariano, G.M., Perinotti, P.: Theoretical framework for quantum networks. Physical Review A 80, 022339 (Aug 2009)
18. Dunjko, V., Fitzsimons, J., Portmann, C., Renner, R.: Composable security of delegated quantum computation. eprint (2014), http://www.arxiv.org/abs/1301.3662
19. Dunjko, V., Kashefi, E., Leverrier, A.: Universal blind quantum computing with weak coherent pulses. Physical Review Letters 108, 200502 (May 2012)
20. Fitzsimons, J., Kashefi, E.: Unconditionally verifiable blind computation (2012), http://www.arxiv.org/abs/1203.5217, eprint
21. Giovannetti, V., Maccone, L., Morimae, T., Rudolph, T.G.: Efficient universal blind computation. Physical Review Letters 111, 230501 (Dec 2013)
22. Goldreich, O.: Foundations of Cryptography: Volume 1, Basic Tools. Cambridge University Press, New York, NY, USA (2001)
23. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, New York, NY, USA (2004)
24. Gutoski, G.: On a measure of distance for quantum strategies. Journal of Mathematical Physics 53(3), 032202 (2012)
25. Gutoski, G., Watrous, J.: Toward a general theory of quantum games. In: Proceedings of the 39th Symposium on Theory of Computing, STOC '07. pp. 565–574. ACM (2007)
26. Hofheinz, D., Müller-Quade, J., Unruh, D.: On the (im)possibility of extending coin toss. In: Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 504–521. Springer (2006)

27. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is ssl?). In: Advances in Cryptology – CRYPTO 2001. Lecture Notes in Computer Science, vol. 2139, pp. 310–331. Springer (2001)
28. Mantri, A., Pérez-Delgado, C.A., Fitzsimons, J.F.: Optimal blind quantum computation. Physical Review Letters 111, 230502 (Dec 2013)
29. Maurer, U., Renner, R.: Abstract cryptography. In: Proceedings of Innovations in Computer Science, ICS 2010. pp. 1–21. Tsinghua University Press (2011)
30. Maurer, U., Tackmann, B.: On the soundness of authenticate-then-encrypt: Formalizing the malleability of symmetric encryption. In: Proceedings of the 17th ACM Conference on Computer and Communication Security. pp. 505–515. ACM (2010)
31. Morimae, T.: Continuous-variable blind quantum computation. Physical Review Letters 109, 230502 (Dec 2012)
32. Morimae, T.: Verification for measurement-only blind quantum computing. Physical Review A 89, 060302 (Jun 2014)
33. Morimae, T., Dunjko, V., Kashefi, E.: Ground state blind quantum computation on AKLT state (2010), http://www.arxiv.org/abs/1009.3486, eprint
34. Morimae, T., Fujii, K.: Blind topological measurement-based quantum computation. Nature Communications 3, 1036 (2012)
35. Morimae, T., Fujii, K.: Blind quantum computation protocol in which alice only makes measurements. Physical Review A 87, 050301 (May 2013)
36. Morimae, T., Koshiba, T.: Composable security of measuring-Alice blind quantum computation (2013), http://www.arxiv.org/abs/1306.2113, eprint
37. Mosca, M., Stebila, D.: Quantum coins. In: Error-Correcting Codes, Finite Geometries and Cryptography. Contemporary Mathematics, vol. 523, pp. 35–47. American Mathematical Society (2010)
38. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
39. Pfitzmann, B., Waidner, M.: A model for asynchronous reactive systems and its application to secure message transmission. In: IEEE Symposium on Security and Privacy. pp. 184–200. IEEE (2001)
40. Portmann, C., Renner, R.: Cryptographic security of quantum key distribution (2014), http://www.arxiv.org/abs/1409.3525, eprint
41. Sueki, T., Koshiba, T., Morimae, T.: Ancilla-driven universal blind quantum computation. Physical Review A 87, 060301 (Jun 2013)
42. Unruh, D.: Simulatable security for quantum protocols (2004), http://www.arxiv.org/abs/quant-ph/0409125, eprint
43. Unruh, D.: Universally composable quantum multi-party computation. In: Advances in Cryptology – EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 486–505. Springer (2010)
44. Unruh, D.: Concurrent composition in the bounded quantum storage model. In: Advances in Cryptology – EUROCRYPT 2011. Lecture Notes in Computer Science, vol. 6632, pp. 467–486. Springer (2011)